



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2015/03.17.13.37-TDI

UMA ABORDAGEM PARA O PROCESSO DE DESENVOLVIMENTO DE SISTEMA EMBARCADO QUE ATENDE AO NÍVEL 2 DE MATURIDADE DO CMMI-DEV

Magda Aparecida Silverio Miyashiro

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelos Drs. Mauricio Gonçalves Vieira Ferreira, e Nilson Sant'Anna, aprovada em 10 de fevereiro de 2015.

URL do documento original:

<http://urlib.net/8JMKD3MGP3W34P/3J6DTUH>

INPE
São José dos Campos
2015

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO
DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Amauri Silva Montes - Coordenação Engenharia e Tecnologia Espaciais (ETE)

Dr. André de Castro Milone - Coordenação Ciências Espaciais e Atmosféricas
(CEA)

Dr. Joaquim José Barroso de Castro - Centro de Tecnologias Espaciais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos
(CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Duca Barbedo - Serviço de Informação e Documentação
(SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2015/03.17.13.37-TDI

UMA ABORDAGEM PARA O PROCESSO DE DESENVOLVIMENTO DE SISTEMA EMBARCADO QUE ATENDE AO NÍVEL 2 DE MATURIDADE DO CMMI-DEV

Magda Aparecida Silverio Miyashiro

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelos Drs. Mauricio Gonçalves Vieira Ferreira, e Nilson Sant'Anna, aprovada em 10 de fevereiro de 2015.

URL do documento original:

<http://urlib.net/8JMKD3MGP3W34P/3J6DTUH>

INPE
São José dos Campos
2015

Dados Internacionais de Catalogação na Publicação (CIP)

Miyashiro, Magda Aparecida Silverio.

M699a Uma abordagem para o processo de desenvolvimento de sistema embarcado que atende ao nível 2 de maturidade do cmmi-dev / Magda Aparecida Silverio Miyashiro. – São José dos Campos : INPE, 2015.

xxvi + 211 p. ; (sid.inpe.br/mtc-m21b/2015/03.17.13.37-TDI)

Tese (Doutorado em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2015.

Orientadores : Drs. Mauricio Gonçalves Vieira Ferreira, e Nilson Sant'Anna.

1. Qualidade de software. 2. Software embarcado. 3. Sistemas embarcados. 4. Sistemas críticos. 5. CMMI. I.Título.

CDU 629.7:004.05



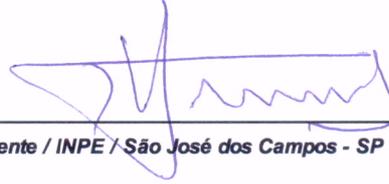
Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Doutor(a)** em

**Engenharia e Tecnologia
Espaciais/Gerenciamento de Sistemas
Espaciais**

Dr. Ronaldo Arias



Presidente / INPE / São José dos Campos - SP

Dr. Mauricio Gonçalves Vieira Ferreira



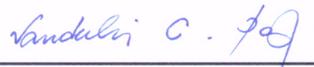
Orientador(a) / INPE / SJCampos - SP

Dr. Nilson Sant'Anna



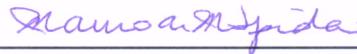
Orientador(a) / INPE / SJCampos - SP

Dr. Vanderlei Cunha Parro



Convidado(a) / Inst. Mauá Tecnologi / Santo André - SP

Dr. Mauro de Mesquita Spinola



Convidado(a) / USP / São Paulo - SP

Este trabalho foi aprovado por:

() **maioria simples**

unanimidade

Aluno (a): **Magda Aparecida Silvério Miyashiro**

São José dos Campos, 10 de Fevereiro de 2015

“[...] Pode pedir sempre. Lembre-se, está na Bíblia: DEUS é onipotente e onipresente. Isso significa que cada um de nós tem um DEUS só seu. Quando você pede algo a ELE, não está tirando a vez de ninguém, fique tranquila [...]”

(Professor Doutor José Demísio Simões da Silva, INPE, 2005.)

“Permaneçei firmes na fé ante a tempestade.”

(Emmanuel)

“Vereis que as tempestades passarão e um lindo dia de
bonança raiará, como prenúncio do amor de Deus por vós.”

(Dr. Bezerra de Menezes)

“[...] Quando não conseguir correr através dos anos, trote.
Quando não conseguir trotar, caminhe.
Quando não conseguir caminhar, use uma bengala.
Mas nunca se detenha.”

(Madre Teresa de Calcutá)

DEDICATÓRIA

Dedico este trabalho a meu pai José Luiz Silvério, que sempre dizia que tinha orgulho de mim, de minha determinação e do meu empenho, e esperava ver sua filha se tornar “Doutora”.

Meu pai estudou somente até o Ensino Fundamental, que na época se chamava Ensino Primário. Para se manter, vendeu pedaços de coco na porta da escola e foi engraxate de sapatos em pontos de ônibus. Mais tarde, foi trabalhar em alfaiataria, onde passava roupas, e como operário em diversas empresas automobilísticas. Foi assim que ele educou e formou os filhos.

Apesar de ele ter recebido somente formação escolar básica e de sempre ter trabalhado como operário na indústria metalúrgica, tinha conhecimentos práticos que o faziam ser respeitado por muitos engenheiros titulados. Mesmo sem muita instrução, ele sabia o suficiente para me ajudar a entender alguns aspectos de Engenharia necessários para que eu cumprisse a disciplina Introdução à Tecnologia de Satélites. Nunca me esqueço da felicidade dele por ter conseguido me auxiliar em uma disciplina de um curso de doutorado – naquele dia, ele se sentiu muito importante. Acredito que, esteja onde estiver, ele está torcendo por mim.

AGRADECIMENTOS

Agradeço primeiramente a meu marido Celso Chokiti Miyashiro, que tanto colaborou com meu doutorado assumindo atividades que eu deveria desenvolver em minha família, além de contribuir diretamente na elaboração desta tese. Agradeço a meus filhos Jorge, Catarina e Carolina o entendimento e o apoio dados durante todo o programa, aceitando minha ausência e ajudando o pai nas tarefas que deixei de executar em casa.

Agradeço principalmente aos meus orientadores, Professor Doutor Maurício Ferreira e Professor Doutor Nilson Sant’Ana, a orientação e a confiança, o apoio incondicional e o direcionamento para o desenvolvimento desta tese. Gostaria de agradecer de forma especial ao Professor Doutor José Demísio Simões da Silva, meu primeiro contato no programa de Pós-graduação do INPE, que me recebeu e confiou em mim, aceitando minha inscrição como aluna de matéria isolada e depois me orientando no programa de mestrado. Com isso, ele me abriu as portas do INPE, possibilitando-me a conquista do título de mestre e viabilizando a defesa do título de Doutora.

Foram muitas as pessoas que, de forma indireta, me auxiliaram na conclusão desta tese, entre elas meus sogros, meus cunhados e o Padre Gennaro, por meio de suas orações.

Agradeço especialmente ao meu cunhado Tomás Shoji Miyashiro, que em conjunto com o meu marido auxiliou na revisão da tese para a banca de qualificação, e ao meu grande amigo Marcelo Succi, que me ajudou com a revisão técnica.

Agradeço também a minha professora de Inglês, Patrícia Petroni, da escola de Idiomas Move Up. Além das dedicadas aulas, que resultaram em uma apresentação bem-sucedida em um congresso em Londres, Patrícia revisou o texto “Processo de desenvolvimento de sistemas embarcados” (Apêndice B desta tese), que precisou ser versado para o inglês antes de ser apresentado em reunião com a Agência Espacial Europeia (ESA) e disponibilizado para a *University of Huddersfield – School of Computing and Engineering (Electronic and Electrical Engineering)* Dr. Violeta Holmes, *HPC Research*

Group Leader, para uso no desenvolvimento de sistema embarcado para a indústria automobilística, a ser realizado por um aluno de seu grupo de pesquisa.

Agradeço à Andréa Vidal, profissional responsável por fazer toda a revisão do texto final desta tese, transformando-a assim em um trabalho com qualificação para ser publicado e representar o Instituto Nacional de Pesquisas Espaciais (INPE).

Agradeço ao Antônio Cassiano Júlio Filho, do Laboratório de *Hardware e Software* da Divisão de Desenvolvimento de Sistemas de Solo (DSS) do INPE, e ao Rafael Corsi Ferrão, do Núcleo de Sistemas Eletrônicos Embarcados (NSEE) do Instituto Mauá de Tecnologia (IMT), pelo auxílio na realização do estudo de caso.

Agradeço à Casa Perseverança por sempre oferecer recursos para me manter no caminho da integridade e da moral.

Deixo por último o mais importante agradecimento: obrigada a Deus por ter me dado toda a força necessária para a realização deste trabalho quando pensei, inúmeras vezes, que não conseguiria.

RESUMO

Hoje, com o aumento do uso de produtos que utilizam programas de computador para executar algumas de suas funcionalidades, torna-se essencial o desenvolvimento de *softwares* de qualidade, em especial nos *softwares* que podem colocar em risco a vida dos seres humanos. Os *softwares* embarcados são componentes inseridos em um sistema maior que, em geral, realiza atividades específicas. Eles são mais utilizados pela indústria, em aplicações que vão desde equipamentos pessoais/domésticos e brinquedos infantis até dispositivos médicos, controles de aviação, órbitas de satélites e missões científicas. Os sistemas embarcados são compostos de uma parte física (*hardware*), que já é produzida com maior nível de maturidade, e uma parte lógica (*software*), para cujo desenvolvimento ainda se busca melhorias em seus níveis de maturidade. Como essa condição ainda não foi alcançada, há certa instabilidade nesse processo de desenvolvimento, o que aumenta a criticidade dos componentes com tais características. Esta tese propõe a elaboração de um processo cíclico para o desenvolvimento de um sistema embarcado com fases, atividades e *templates* bem definidos, que atendam às exigências equivalentes ao nível 2 de maturidade do CMMI-DEV. Esse processo utiliza-se do conceito de individualização das partes do componente embarcado (*hardware* e *software*) de forma que elas sejam desenvolvidas com a integração que o componente embarcado requer, atentas às particularidades das partes *hardware* e *software* do componente embarcado resultante ao fim de um ciclo.

Palavras-chave: qualidade de *software*, *software* embarcado, sistemas embarcados, sistemas críticos.

**AN APPROACH TO THE PROCESS OF EMBEDDED SYSTEM
DEVELOPMENT THAT ATTENDS THE LEVEL OF 2 CMMI-DEV
MATURITY**

ABSTRACT

Today, with the increased use of products using computer programs to perform some of its functionality, it is essential to the development of quality software, especially in software that can endanger the lives of human beings. The embedded software components are inserted into a larger system that, in general, performs specific activities. They are mostly used by the industry in applications ranging from personal / household equipment and children's toys to medical devices, aviation controls, satellite orbits and scientific missions. The embedded systems are composed of a physical part (hardware), which is already produced with a higher level of maturity, and a logical part (software), to whose development still seeking improvements in their maturity levels. As this condition has not been met, there is some instability in the development process, which increases the criticality of components having such characteristics. This thesis proposes the development of a cyclic process for the development of an embedded system with phases, activities and well-defined templates that meet requirements equivalent to Level 2 of maturity CMMI-DEV. This process uses the concept of individualization of parts of the embedded component (hardware and software) so that they are developed by integrating the embedded component requires attentive to the particular party hardware and embedded software component resulting after a cycle.

Keywords: quality of software, embedded software, embedded systems, critical systems.

LISTA DE FIGURAS

Figura 1.1 – Representação de um sistema computacional convencional.....	1
Figura 1.2 – Resultado do <i>Chaos Manifesto</i> entre 2004 e 2012.....	2
Figura 1.3 – Representação de um sistema embarcado.....	2
Figura 2.1 – Representação modelo clássico.....	18
Figura 2.2 – Dimensões críticas de um processo	20
Figura 2.3 – Estrutura do TSP/PSP	23
Figura 2.4 – Funcionamento do TSP	24
Figura 2.5 – Estrutura do SCRUM.....	25
Figura 2.6 – <i>Tower Bridge</i> , em Londres.....	27
Figura 2.7 – Fatores de qualidade.....	29
Figura 3.1 – Qualidade de <i>software</i> durante seu desenvolvimento	34
Figura 3.2 – Componentes do modelo CMMI (SEI, 2010).....	36
Figura 3.3 – Visibilidade do CMMI.....	38
Figura 3.4 – Áreas de processo do CMMI.....	39
Figura 3.5 – Modelos de CMMI de representação contínua e por estágios	40
Figura 3.6 – Organização da representação contínua.....	41
Figura 3.7 – Organização da representação por estágios.....	43
Figura 3.8 – Componentes do MPS.BR	59
Figura 3.9 – Modelo de processo brasileiro	60
Figura 4.1 – Processo de definição de um sistema espacial	69
Figura 4.2 – Elementos do ciclo de vida de um projeto na área espacial, padrão ECSS- M-ST-10C.....	70
Figura 4.3 – Elementos que constituem os segmentos de um sistema espacial	71

Figura 4.4 – Elementos dos segmentos de um sistema espacial.....	72
Figura 4.5 – Disciplinas do padrão ECSS	74
Figura 4.6 – Relacionamento dos processos de <i>software</i> no padrão ECSS.....	76
Figura 4.7 – Estrutura da norma ECSS-E-ST-40	77
Figura 4.8 – Estrutura da norma ECSS-Q-80	79
Figura 5.1 – Fluxo simplificado do projeto de sistemas embarcados.....	83
Figura 5.2 – Relação entre <i>software</i> embarcado e <i>software</i> geral.....	86
Figura 5.3 – Processo do ciclo de vida do <i>software</i> embarcado.....	91
Figura 6.1 – Interação entre as atividades de projeto	95
Figura 6.2 – Fluxo de funcionamento.....	98
Figura 6.3 – Fases do modelo baseado em codesign.....	99
Figura 6.4 – Esquema do método Volere	100
Figura 6.5 – Arquitetura de processos	102
Figura 6.6 – V-Model	103
Figura 6.7 – Organização SCRUM para sistema embarcado	104
Figura 6.8 – Visão geral do <i>software</i> produzido	105
Figura 6.9 – Modelo de processo proposto	107
Figura 6.10 – <i>Model Driven</i>	109
Figura 6.11 – Abordagem MDEReq.....	111
Figura 7.1 – Nível 2 CMMI gerenciado	121
Figura 7.2 – Processo proposto para sistemas embarcados.	122
Figura 7.3 – Funcionamento do Processo Cíclico.	123
Figura 7.4 – Componentes do processo cíclico.	124
Figura 7.5 – Atividades comuns – Fase.....	125
Figura 7.6 – Engenharia – Ciclo.....	126

Figura 7.7 – Engenharia – Fase.	127
Figura 7.8 – Requisitos – Fase.....	128
Figura 7.9 – Requisitos do produto – ciclo.....	129
Figura 7.10 – Análise e projeto – fase.	130
Figura 7.11 – Análise e projeto – ciclo.....	131
Figura 7.12 – Implementação – fase.....	132
Figura 7.13 – Integração – fase.	133
Figura 7.14 – Implementação e integração – ciclo.....	134
Figura 7.15 – Validação e verificação – Fase.....	135
Figura 7.16 – Verificação e validação – ciclo.	136
Figura 7.17 – Avaliação do ciclo – fase.	137
Figura 7.18 – Avaliação do ciclo – ciclo.	137
Figura 7.19 – Funcionamento da arquitetura.	141
Figura 7.20 – Linha do tempo dos artefatos.	141
Figura 7.21 – Representação gráfica do fluxograma.	142
Figura 7.22 – Fluxograma geral de realização do processo – parte 1.	143
Figura 7.23 – Fluxograma geral de realização do processo – parte 2.	144
Figura 7.24 – Notação para o fluxograma das fases.....	145
Figura 7.25 – Fluxograma da fase de engenharia.	146
Figura 7.26 – Fluxograma da fase de análise de requisitos.	147
Figura 7.27 – Fluxograma da fase de análise e projeto.	148
Figura 7.28 – Implementação e integração.....	149
Figura 7.29 – Fluxograma da fase de verificação e validação.....	150
Figura 7.30 – Fluxograma da fase de avaliação do ciclo.....	151
Figura 7.31 – Gráfico de atendimento por área de processo.	180

Figura 7.32 – Gráfico de atendimento por nível.....	181
Figura 8.1 – Representação da relação do processo desta tese com os trabalhos correlatos.....	188

LISTA DE TABELAS

Tabela 2-1 – Categorias de criticidade do software (adaptada de ECSS, 2009b)	15
Tabela 3.1 – Comparação entre as representações adaptado.....	46
Tabela 3.2 – Gerenciamento de requisitos.....	47
Tabela 3.3 – Planejamento do projeto	47
Tabela 3.4 – Monitoramento e controle do projeto	48
Tabela 3.5 – Gerenciamento de acordos com fornecedores	48
Tabela 3.6 – Medições e análises	49
Tabela 3.7 – Garantia da qualidade do processo e do produto	49
Tabela 3.8 – Gerenciamento de configurações.....	50
Tabela 3.9 – Desenvolvimento de requisitos.....	50
Tabela 3.10 – Solução técnica	51
Tabela 3.11 – Integração de produto	51
Tabela 3.12 – Verificação.....	52
Tabela 3.13 – Validação	52
Tabela 3.14 – Foco no processo da organização (OPF)	53
Tabela 3.15 – Definição do processo da organização	53
Tabela 3.16 – Treinamento organizacional	54
Tabela 3.17 – Gerência integrada de projeto.....	54
Tabela 3.18 – Gerência de risco	55
Tabela 3.19 – Análise de decisão e resolução	55
Tabela 3.20 – Desempenho do processo organizacional.....	56
Tabela 3.21 – Gerência quantitativa de projeto.....	56
Tabela 3.22 – Análise causal e resolução	57

Tabela 3.23 – Gestão do processo organizacional.....	57
Tabela 3.24 – Práticas genéricas.....	58
Tabela 3.25 – Similaridade CMMI × MPSBr	61
Tabela 5.1 – Fatores a serem considerados em sistemas embarcados.....	89
Tabela 6.1 – Tarefas do desenvolvimento de sistemas embarcados.....	96
Tabela 6.2 – Gerência do desenvolvimento de sistema embarcado	97
Tabela 6.3 – Tabela de Correlação	113
Tabela 7.1 – Práticas genéricas.....	153
Tabela 7.2 – Gerência de requisitos.....	154
Tabela 7.3 – Medição e Análise	156
Tabela 7.4 – Acompanhamento e controle de projeto	157
Tabela 7.5 – Gerência de acordos com fornecedores	159
Tabela 7.6 – Planejamento de projeto	160
Tabela 7.7 – Gerência de configuração	162
Tabela 7.8 – Garantia da qualidade de processo e produto	163
Tabela 7.9 – Desenvolvimento de requisitos.....	164
Tabela 7.10 – Solução técnica	165
Tabela 7.11 – Integração de produto	167
Tabela 7.12 – Verificação.....	168
Tabela 7.13 – Validação	170
Tabela 7.14 – Foco no processo da organização	171
Tabela 7.15 – Definição do processo da organização	171
Tabela 7.16 – Treinamento organizacional	172
Tabela 7.17 – Gerência integrada de projeto.....	173
Tabela 7.18 – Gerência de risco	175

Tabela 7.19 – Análise de decisão e resolução	176
Tabela 7.20 – Desempenho do processo organizacional.....	177
Tabela 7.21 – Gerência quantitativa de projeto.....	177
Tabela 7.22 – Análise causal e resolução	178
Tabela 7.23 – Gestão do processo organizacional.....	179
Tabela 7.24 – ECSS-Q-80 - Garantia de Produto Espacial / Garantia de Produto de <i>Software</i>	182
Tabela 7.25 – ECSS-E-ST-40- Engenharia de Software Espacial	182

SUMÁRIO

1. INTRODUÇÃO	1
1.1 Sistemas embarcados.....	3
1.2 Motivação.....	5
1.2.1 Dificuldade de definição de sistemas embarcados.....	6
1.2.2 Reutilização de componentes de sistema	6
1.2.3 Problema.....	7
1.3 Objetivo geral	8
1.3.1 Objetivos específicos.....	10
1.4 Metodologia	11
1.4.1 Pesquisa.....	11
1.4.2 Elaboração da tese	11
1.5 Estrutura da tese	12
2 FUNDAMENTOS CONCEITUAIS.....	13
2.1 Termos comuns e definições	13
2.2 Desenvolvimento de <i>software</i>	17
2.2.1 Descrição das fases.....	18
2.3 Processo.....	19
2.3.1 Processo de desenvolvimento de <i>software</i>	20
2.3.2 Ciclos de vida de desenvolvimento de software	21
2.4 Processo de <i>Software</i> para equipes - <i>Team Software Process (TSP)</i>	22
2.5 SCRUM.....	24
2.6 Qualidade de <i>software</i>	26
2.6.1 Qualidade do produto	28

2.6.2	Qualidade do processo	30
3	MODELOS DE QUALIDADE	33
3.1	Qualidade de processo de software	35
3.2	CMMI V1.3 (<i>Capability Maturity Model Integration</i>)	35
3.2.1	Componentes do modelo CMMI V1.3	37
3.2.2	Representações do modelo CMMI.....	39
3.2.3	Comparação entre as representações.....	45
3.2.4	Práticas a serem realizadas em cada área de processo	47
3.3	MPS.BR.....	58
3.4	Modelos de maturidade e os sistemas embarcados	62
3.5	Técnicas, ferramentas e padrões que auxiliam a qualidade do processo	64
3.5.1	Técnicas e ferramentas utilizadas na área espacial (exemplos)	64
3.5.2	Modelagem funcional.....	65
3.5.3	Modelagem orientada a objetos.....	65
4	ÁREA ESPACIAL	67
4.1	Exploração espacial.....	67
4.1.1	Missão espacial	68
4.1.2	Sistema espacial	69
4.1.3	Distribuição das responsabilidades entre os segmentos	71
4.2	Requisitos para o desenvolvimento de sistemas embarcados na área espacial ..	73
4.2.1	Engenharia de Software Espacial	76
4.2.2	Garantia de Produto Espacial – Garantia de Produto de Software.....	78
5	SISTEMAS EMBARCADOS	81
5.1	<i>Software</i> embarcado	85
5.2	<i>Hardware</i> embarcado	87

5.3	Desenvolvimento de sistemas embarcados	87
5.3.1	Desenvolvimento de <i>software</i>	88
5.3.2	Ciclos de vida do software embarcado.....	90
5.4	Disponibilidade	91
5.5	Acidentes	92
6	TRABALHOS CORRELATOS	95
6.1	Diretrizes para o desenvolvimento de <i>software</i> de sistemas embarcados.....	95
6.1.1	Conceito geral da proposta.....	95
6.1.2	Atividades gerais do processo	95
6.1.3	Atividades de gerenciamento de sistema embarcado.....	96
6.2	Desenvolvimento de sistemas embarcados para aplicações espaciais	97
6.2.1	Conceito geral da proposta.....	98
6.3	Model-Based Codesign	99
6.3.1	Conceito geral da proposta.....	99
6.4	Estudo de requisitos do <i>software</i> embarcado no segmento da telemedicina....	100
6.4.1	Conceito geral da proposta.....	100
6.4.2	Atividades gerais do processo	101
6.5	Definição de um processo de engenharia de requisitos.....	101
6.5.1	Conceito geral da proposta.....	101
6.5.2	Atividades gerais do processo	102
6.6	Metodologia de desenvolvimento ágil em sistemas embarcados.....	103
6.7	Metodologia de projeto de <i>software</i> embarcada voltada ao teste.....	104
6.7.1	Conceito geral da proposta.....	104
6.7.2	Atividades gerais do processo	105
6.8	Diretrizes para desenvolvimento de <i>software</i> para sistemas embarcados.....	106

6.8.1	Conceito geral da proposta	106
6.8.2	Atividades gerais do processo	107
6.9	<i>Model-Driven Engineering of Complex Embedded Systems</i>	108
6.9.1	Conceito geral da proposta	108
6.9.2	Atividades gerais do processo	109
6.10	MDEReq.....	110
6.10.1	Conceito geral da proposta	110
6.10.2	Atividades gerais do processo	111
6.11	Tabelas de correlação	112
7	ARQUITETURA DO PROCESSO	117
7.1	Processo proposto.....	119
7.1.1	Ciclo de desenvolvimento	123
7.1.2	Atividades comuns (FC).....	124
7.1.3	Engenharia (F1).....	126
7.1.4	Análise de requisitos (F2)	128
7.1.5	Análise e projeto (F3).....	129
7.1.6	Implementação e integração (F4).....	131
7.1.7	Verificação e validação do sistema (F5)	134
7.1.8	Avaliação do ciclo (F6)	136
7.1.9	Descrição dos documentos a serem produzidos durante o processo.	138
7.2	Representação da arquitetura.....	140
7.3	Fluxograma do processo.....	142
7.3.1	Fluxograma geral do processo.....	143
7.3.2	Fluxograma de cada fase do processo	145
7.3.2.1	Fluxograma da fase de engenharia – F1	146

7.3.2.2	Fluxograma da fase de análise de requisito – F2.....	147
7.3.2.3	Fluxograma da fase da análise e projeto – F3	148
7.3.2.4	Fluxograma da fase de implementação e integração – F4	149
7.3.2.5	Fluxograma da fase da fase de verificação e validação – F5	150
7.3.2.6	Fluxograma da fase da fase de avaliação do ciclo – F6	151
7.4	Maturidade do processo proposto	152
7.4.1	Atendimento às práticas do CMMI	152
7.4.1.1	Genéricas	152
7.4.2	Específicas de nível 2	154
7.4.2.1	Gerência de requisitos	154
7.4.2.2	Medição e análise	155
7.4.2.3	Acompanhamento e controle de projeto.....	157
7.4.2.4	Gerência de acordo com fornecedores	158
7.4.2.5	Planejamento de projeto.	160
7.4.2.6	Gerência da configuração.....	162
7.4.2.7	Garantia da qualidade	163
7.4.3	Específicas de nível 3	163
7.4.3.1	Desenvolvimento de requisitos	163
7.4.3.2	Solução técnica.....	165
7.4.3.3	Integração de produto.....	166
7.4.3.4	Verificação	168
7.4.3.5	Validação.....	169
7.4.3.6	Foco no processo da organização.....	170
7.4.3.7	Definição do processo da organização.	171
7.4.3.8	Treinamento organizacional.	172

7.4.3.9	Gerência integrada de projeto	173
7.4.3.10	Gerência de risco	174
7.4.3.11	Análise de decisão e resolução.....	176
7.4.4	Específicas de nível 4	176
7.4.4.1	Desempenho do processo organizacional	176
7.4.4.2	Gerência quantitativa de projeto.....	177
7.4.5	Específicas de nível 5	178
7.4.5.1	Análise causal e resolução.....	178
7.4.5.2	Gestão do processo organizacional	178
7.5	Estatística da cobertura do processo ao modelo CMMI-DEV	179
7.6	Atendimento as normas ECSS	181
7.7	Estudo de caso	184
8	CONSIDERAÇÕES FINAIS	187
8.1	Trabalhos correlatos e o processo apresentado	187
8.2	Objetivos da tese x resultados apresentados	188
8.2.1	Com relação ao objetivo geral.....	188
8.2.2	Com relação aos objetivos específicos.....	189
8.3	Contribuição acadêmica	193
8.4	Conclusão	194
8.5	Publicações.....	196
8.6	Trabalhos futuros.....	197
	REFERÊNCIAS BIBLIOGRÁFICAS	199
	Apêndice A – Glossário	
	Apêndice B – Processo de desenvolvimento – descrição	
	Apêndice C – Plano de desenvolvimento	

Apêndice D – Especificação do produto

Apêndice E – Especificação do componente

Apêndice F – Projeto de teste

Apêndice G – Homologação

Apêndice H – Configuração

Apêndice I – Processo completo versado em inglês

Anexo A – Estudo de caso – INPE

Anexo B – Estudo de caso – Mauá

Anexo C – Estudo de caso – Mauá

1. INTRODUÇÃO

Na década de 1960, com o aumento do número de sistemas baseados em computadores e a consequente dependência da sociedade ao uso de sistemas de *software*, a complexidade do desenvolvimento desses *softwares* cresceu, dando origem à chamada *crise do software* (PRESSMAN, 2011).

A *crise do software* foi um período em que os produtos de *software* eram feitos de forma artesanal e a engenharia de *software* era praticamente inexistente. Refere-se a um conjunto de problemas encontrados no desenvolvimento de *softwares* e suas atividades de manutenção. A expressão representou as dificuldades de desenvolvimento de *softwares* diante do avanço da demanda e da ampliação da complexidade dos problemas a serem resolvidos, agravados pela inexistência de técnicas de desenvolvimento que funcionassem adequadamente e pudessem ser validadas.

Os sistemas computacionais são compostos de *software* e *hardware*. O *hardware* é a parte física do sistema, os circuitos, os microprocessadores etc.; o *software* é a parte lógica, um conjunto de instruções a serem processadas pelos circuitos e microprocessadores eletrônicos do *hardware* para atingir um objetivo, conforme representado na Figura 1.1. Pode-se também considerar as pessoas como mais um componente dos sistemas, pela importância que têm em seu funcionamento.

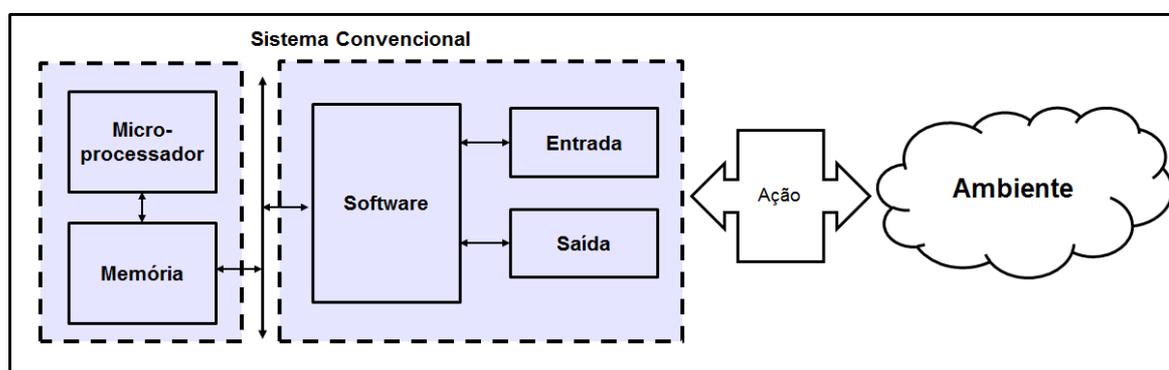


Figura 1.1 – Representação de um sistema computacional convencional

Fonte: adaptado de Luiz (2010).

Nos cinquenta anos após o início dessa crise, houve uma grande evolução na área de desenvolvimento de *softwares*, com o surgimento de novos métodos, ferramentas e técnicas. No entanto, mesmo com o uso de tais recursos, as dificuldades não mudaram conforme as expectativas, como apresentou o resultado da pesquisa do *Chaos Reporter* entre 1994 e 2010 (Figura 1.2).

RESOLUTION					
	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Project resolution results from CHAOS research for years 2004 to 2012.

Figura 1.2 – Resultado do *Chaos Manifesto* entre 2004 e 2012

Fonte: CHAOS MANIFESTO (2013).

Uma das categorias de *software* que passaram por uma evolução foi a dos *sistemas embarcado* que inicialmente foram definidos como sistemas ou componentes compostos de *hardware* e *software* integrados que realizam tarefas específicas e geralmente com grandes restrições. Seu funcionamento deve ser de alta confiabilidade e qualidade. Na maioria das vezes, tais sistemas não funcionam isoladamente; fazem parte de um sistema maior e reagem a seus estímulos (CURTIS, 2000; DIB, 2008 apud BARROSO, 2010), conforme mostrado na Figura 1.3.

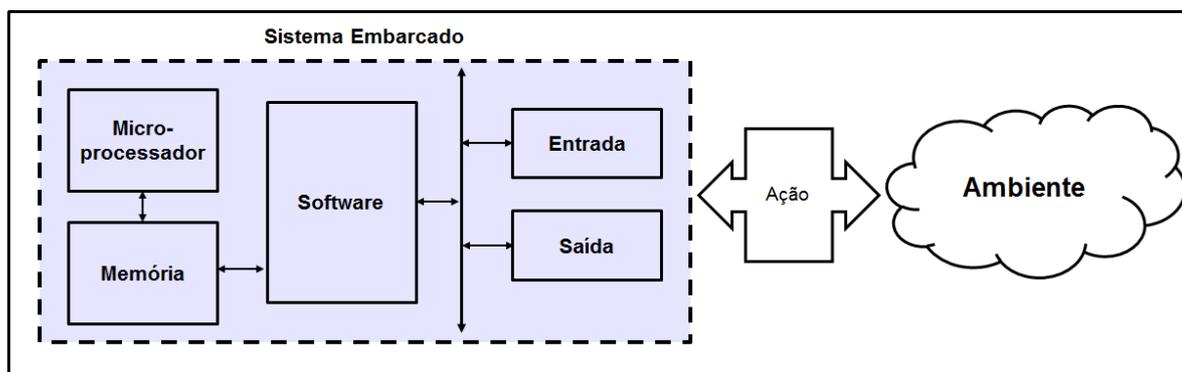


Figura 1.3 – Representação de um sistema embarcado

Fonte: adaptada Luiz (2010).

1.1 Sistemas embarcados

Um sistema embarcado é um produto instalado em um computador que, em geral, é dedicado a um dispositivo com objetivos que realiza um conjunto de tarefas predefinidas com requisitos específicos. Sua execução se dá em computadores formados, fundamentalmente, pelos mesmos componentes de um computador convencional, como processador, memória, algum dispositivo de armazenamento, interfaces, entre outros, com uma considerável diferença nos recursos computacionais, que são limitados.

Os sistemas embarcados surgiram com a função de executar com eficiência uma única tarefa de forma contínua, sem pausas ou interrupções indevidas.

Quando se aborda o tema *qualidade de software*, seja na área acadêmica, seja na aplicação em empresas, se pensa em vários modelos e metodologias que são aplicados no processo de desenvolvimento. Porém, quando se fala em desenvolver *software* e *hardware* juntos, questiona-se: como integrar essas duas realidades garantindo a qualidade? (BARROSO, 2010)

Segundo (SPINOLA, 1998), os estudos relativos à caracterização e ao desenvolvimento desses sistemas têm evoluído desde o início dos anos 1980, iniciando-se com os equipamentos militares, integrando-se aos sistemas de supervisão e controle de processos, e generalizando-se nos últimos anos para as mais diversas aplicações cotidianas (BALARIN, 1997; KUMAR et al., 1996 apud SPINOLA; PESSÔA, 1997; SPINOLA, 1998), exemplificados em usos tais como eletrodomésticos, sistemas PABX, automóveis, aviões, brinquedos etc. Tal fato implica exigências cada vez maiores e diferenciadas, ampliando, de forma geral, as dificuldades de desenvolvimento de sistemas embarcados, tanto em quantidade quanto em importância (SPINOLA, 1998).

Hoje, graças à dependência do uso de sistemas de *software*, é comum encontrar aplicações de sistemas embarcados em diversas áreas, como a de sistemas automotivos, periféricos computacionais, robôs, aeronaves e até espaciais (LABROSSE, 2002; VÉRAS, 2007 apud MIRACHI; CARVALHO, 2009), assim como equipamentos para uso na medicina, equipamentos domésticos e dispositivos pessoais. Com o desenvolvimento e a evolução da tecnologia, os sistemas embarcados também têm se tornado cada vez mais sofisticados e

complexos, o que influencia diretamente seu processo de desenvolvimento, tornando-o uma atividade custosa e propensa a erros (VÉRAS, 2007 apud MIRACHI; CARVALHO, 2009).

O aumento do uso de sistemas embarcados se deu principalmente em equipamentos que realizam funções críticas em sistemas que afetam diretamente a vida das pessoas, como é o caso de *softwares* utilizados na aviação, em equipamentos médicos e de outros tipos dos quais dependem vidas. Um produto de *software* é considerado crítico quando, em caso de falha, pode levar um sistema a um estado perigoso, isto é, quando o *software* pode provocar desastres ou fatalidades (FALLA, 1998 apud PÁSCOA, 2002).

Os problemas gerados pelo desenvolvimento de *softwares* complexos levaram à conclusão de que, apesar de todos os esforços realizados, esses *softwares* ainda podem ser construídos com algumas falhas que comprometam seu funcionamento. Isso pode ser observado pelo alto número de defeitos apresentados pelos programas (PERRY, 1992 apud SPINOLA, 1998), que levaram ao que originou a crise do *software* (GIBBS, 1994 apud ALONSO, 1998).

Exemplos clássicos demonstram que *softwares* embarcados desenvolvidos para desempenhar sistemas críticos muitas vezes falham, como nos casos do foguete Ariane 5, que explodiu 37 segundos após o lançamento, e da máquina de radioterapia Therac-25, que fez seis vítimas fatais, entre outros.

Esses fatos mostram a necessidade de estudar o papel do *software* no produto final, a relação entre os problemas do usuário de sistemas embarcados e sua origem, pois, em geral, seus projetos são planejados e construídos visando às funcionalidades gerais do produto, e não observando separadamente as funcionalidades a serem realizadas pelo *software* e pelo *hardware*, o que significa que, de modo geral, os projetos não são elaborados apropriadamente para o *software*.

Um *software* falha porque apresenta defeito (MOSSBERG, 1995; DEMARCO, 1989 apud SPINOLA, 1998), que ocorre quando um requisito não é corretamente implementado no sistema, isto é, o defeito se dá pela falta total ou parcial de um requisito. A falha é a manifestação do defeito existente. Considera-se que uma falha ocorre quando o sistema não realiza a tarefa

requerida ou esperada durante uma operação (STRAUSS; EBENAU, 1994; LANNINO et al., 1984; CRESPO et al., 1997 apud SOUZA, 2002).

Dadas as dificuldades para remover os defeitos, sua prevenção constitui-se na alternativa mais consistente para a obtenção de um *software* de alta qualidade (SPINOLA, 1998).

Uma vez que os sistemas embarcados atuam na maioria dos sistemas eletrônicos utilizados em equipamentos dos mais diversos tipos e para numerosos fins, eles constituem um desafio para a indústria de *softwares* embarcados, que necessitam de componentes desenvolvidos com alto nível de qualidade.

O primeiro sistema embarcado reconhecido foi o Apollo Guidance Computer, projeto desenvolvido por Charles Stark Draper no MIT. O computador de guia, que operava em tempo real, foi considerado o item mais arriscado do projeto Apollo (LOUREIRO, 2010).

O primeiro sistema embarcado em linha de produção foi o computador guia do LGM-30 *Missil Minuteman*, míssil nuclear lançado em 1961 que, originalmente, dispunha de um disco rígido para a memória principal. Na versão seguinte, foi substituído por um computador com circuitos integrados. Assim, o uso da tecnologia desse projeto reduziu o valor dos circuitos integrados, o que permitiu seu uso em sistemas comerciais (LOUREIRO, 2010).

1.2 Motivação

A crescente utilização de sistemas embarcados e a identificação de diversos acidentes provocados por falhas em sistemas com essas características estimularam o estudo e a pesquisa de propostas que pudessem contribuir para suprimir tais falhas ou pelo menos minimizá-las.

A importância de *software* de computador na nossa sociedade continuará seu crescimento sem paralelo, com potencial para um prejuízo e insatisfação do cliente também sem paralelo, gerado por um processo de qualidade pobre. (DUNN; ULLMAN, 1994 apud SICKLE, 1997).

1.2.1 Dificuldade de definição de sistemas embarcados

Embora seja amplamente utilizado pelas organizações desenvolvedoras de *software*, é difícil encontrar na literatura, tanto na acadêmica quanto na técnica, uma definição clara e precisa do que é um sistema embarcado.

Conforme será apresentado com maior rigor no Capítulo 5, um sistema embarcado é composto de um conjunto de componentes, alguns dos quais são *hardwares* e outros são *softwares*; o *hardware* é físico e seu ciclo de desenvolvimento em geral, já conquistou um nível de maturidade mais evoluído, enquanto o *software* é lógico e devido a suas particularidades ainda busca por um ciclo de vida estável. Apesar dessas particularidades, os diferentes componentes de um sistema embarcado são tão próximos entre si que, em geral, damos a eles o mesmo tipo de tratamento.

Um estudo desenvolvido pelo *Carnegie Mellon University's Software Engineering Institute (CMU/SEI)* sobre as tendências da indústria no uso de componentes de *software* apontou, entre outras fragilidades, que na engenharia de *software* faltam métodos para produzir sistemas de qualidade a partir de componentes (CARVALHO, 2010). Pesquisas recentes confirmam a deficiência na disponibilidade de processos, métodos, técnicas e ferramentas para que se avalie a qualidade de componentes de *software*, ainda mais especificamente para o *software* embarcado (CARVALHO et al., 2009; SOMMERVILLE, 2007; WALLNAU, 2004; ÁLVARO, 2005 apud CARVALHO et al., 2009; LUCRÉDIO et al., 2007 apud CARVALHO et al., 2009; VOAS, 1998 apud CARVALHO et al., 2009).

1.2.2 Reutilização de componentes de sistema

Um requisito importante para o desenvolvimento de sistemas embarcados com qualidade é a redução do custo e do tempo de desenvolvimento (CARVALHO, 2010). Uma abordagem que vem sendo investigada em diversas pesquisas para atender a esse requisito é a reusabilidade de componentes, isto é, o reúso de componentes externos, conhecidos e validados em projetos anteriores. Isso motivou a indústria de produtos embarcados a incorporar abordagens baseadas em componentes com uma premissa de reutilização.

1.2.3 Problema

A dificuldade de definição de sistemas embarcados reflete na indeterminação ou dificuldade da implementação de diretrizes eficientes para a obtenção de uma boa qualidade do sistema embarcado, isto é, o total atendimento dos requisitos do produto.

Um sistema embarcado, em geral, é um conjunto de subsistemas ou componentes formados por parte *hardware* e parte *software*. O menor componente de um sistema embarcado ainda pode ser considerado sistema quando é composto por *software* e *hardware*, que, juntos, vão realizar determinadas atividades. Cada parte desse sistema pode ser identificada na Figura 1.4.

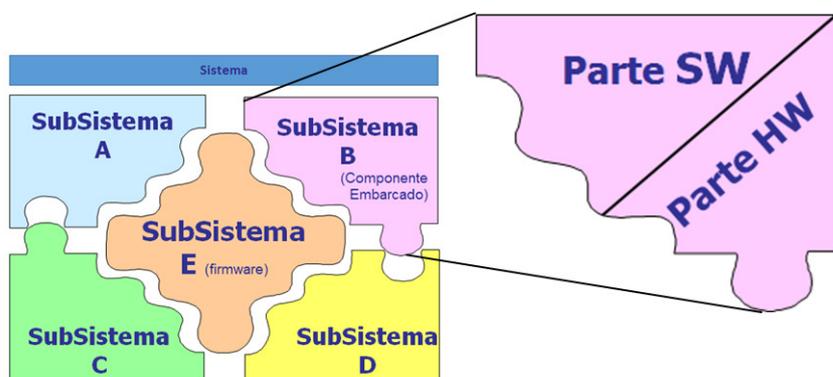


Figura 1.4 – Representação das partes de um sistema embarcado.

Em geral, os processos de desenvolvimento de *software* e os modelos de qualidade de *software* são descritos de forma genérica para atender a todas as categorias/classificações e sugerem que sejam feitas adaptações ou ajustes para que determinadas características sejam assimiladas. Estes, por vezes, são muito grandes e resumem-se a itens ou documentos que não são aplicados, sendo raramente incluídos os documentos necessários, o que dificulta a aplicação dos processos para os softwares embarcados.

A indústria de *hardware*, (manufatura) por ser mais antiga e com processos mais consolidados, há muito já tem um nível maior de maturidade para o desenvolvimento de seus produtos, enquanto a produção do *software*, apesar de sua evolução, ainda está no início desse desenvolvimento, precisa de mais estudos e aplicações para conseguir atingir essa maturidade.

Sem contar que o *hardware*, de modo geral, são módulos genéricos, enquanto os softwares necessitam de implementação de requisitos mais específicos. (SPINOLA, 1998)

Com o grande consumo e a utilização de sistemas embarcados, a estabilidade do *hardware* não é suficiente para garantir a qualidade do sistema como um todo, uma vez que o desenvolvimento dos componentes de *software* desse sistema costuma apresentar fragilidades que podem causar falhas de execução.

O *software* de componente embarcado está conquistando mais espaço e importância, o que aumenta a demanda por soluções e serviços de sistemas embarcados (ACCENTURE, 2014). A maioria dos esforços envolvidos na produção de componentes para sistemas embarcados está sendo direcionada para o desenvolvimento de softwares. Segundo ACCENTURE (2014), 62% do orçamento de pesquisa e desenvolvimento e 67% do custo de um componente embarcado são utilizados no desenvolvimento de softwares, o que indica sua importância em um componente (ACCENTURE, 2014). Outro dado significativo mostra que 33% dos dispositivos produzidos não atendem aos requisitos de funcionalidade, nem de desempenho do produto, e que 80% do esforço de desenvolvimento são gastos na correção de erros não identificados durante as fases anteriores de sua produção (ACCENTURE, 2014). Além disso, 80% dos motivos de falha em sistemas embarcados advêm de problemas no *software*, e não no *hardware* (GOMES, 2010).

As empresas que desenvolvem (ou adquirem) e embarcam *softwares* em seus produtos muitas vezes encontram dificuldades para a utilização de processos para tais atividades, o que tem estimulado o tratamento das partes *software* e *hardware* do componente de forma diferenciada e isolada, contribuindo para dificultar a estruturação dos processos de maneira integrada (SPINOLA, 1998).

1.3 Objetivo geral

O principal objetivo deste trabalho é identificar mecanismos e propor um processo flexível para o desenvolvimento de um sistema embarcado que atenda às necessidades particulares e aos aspectos individuais de software e hardware. Suas necessidades e particularidades podem ser projetadas de forma independente, uma vez que suas características são diferentes por

definição: o hardware é físico e o software é lógico. Contudo, pela proximidade ou até mesmo pela unicidade de suas funções, para atingir o resultado final, seu desenvolvimento deve ser feito simultaneamente.

O grande desafio para o desenvolvimento de sistemas embarcados está em atender requisitos funcionais e não funcionais de cada componente dos sistemas integrados, além de suas características específicas.

O desenvolvimento deste trabalho envolve aspectos práticos e metodológicos que buscam, pro meio da qualidade do processo, conquistar a qualidade do produto (componente embarcado) final.

A produção de um componente embarcado envolve hardware e software, e assim foi considerado para a elaboração do processo, em cujas diretrizes e especificação foram determinadas ações para os dois aspectos, porém com especial atenção à parte software do componente.

Os estímulos para este trabalho são a dificuldade e a grande necessidade de utilização de métodos e técnicas de engenharia de software para o desenvolvimento de sistemas embarcados, sendo os aspectos principais:

- i. O que é e quais são as características de sistemas embarcados?
- ii. Quais as dificuldades encontradas para o desenvolvimento de sistemas embarcados?
- iii. Qual é a fronteira entre sistemas, componentes, software e hardware embarcados?
- iv. Que atividades devem estar presentes em um processo para o desenvolvimento de sistemas embarcados e qual é o nível de detalhamento ideal?
- v. É possível utilizar processos para o desenvolvimento de sistemas embarcados que atendam ao modelo de qualidade CMMI-DEV?
- vi. Como utilizar o processo em desenvolvimento de sistemas críticos embarcados em satélites em projetos de missões espaciais?

Os aspectos apontados visam estruturar o desenvolvimento de sistemas embarcados e estabelecer um percurso adaptável de acordo com as necessidades e as características de cada empresa e de cada sistema/componente a ser desenvolvido.

1.3.1 Objetivos específicos

- i. Definir e caracterizar conceitualmente sistemas, componentes, *software* e *hardware* embarcados.
- ii. Definir as características e fronteiras de um processo para o desenvolvimento de sistemas embarcados.
- iii. Definir um processo de desenvolvimento de sistemas embarcados que ofereça atividades individualizadas para as partes *software* e *hardware* dos componentes do sistema embarcado.
- iv. Definir um processo que particione os componentes dos sistemas embarcados para que sejam produzidos individualmente.
- v. Definir um processo que atenda aos modelos de qualidade CMMI e ao modelo MPS.Br.
- vi. Definir um processo que seja absorvido pelas normas ECSS-E-ST-40C e ECSS-Q-ST-80 do padrão ECSS.
- vii. Definir mecanismos de modelagem prévia do componente, controle e integração dos componentes de *software* e de *hardware* do sistema embarcado para minimizar problemas após a implementação.
- viii. Definir um processo adaptável para o desenvolvimento de componentes para sistemas embarcados, em que cada componente do sistema seja produzido em um ciclo independente e cada execução do ciclo possa adaptar o processo às características do novo componente a ser produzido.
- ix. Definir um processo em que os documentos produzidos sejam resultado das boas práticas das atividades realizadas em todo o processo.

- x. Aplicar os resultados da pesquisa em casos reais.

1.4 Metodologia

A organização apresentada envolve estrutura de desenvolvimento da tese, pois está direcionada para seus objetivos.

1.4.1 Pesquisa

- i. Revisão bibliográfica reunindo as principais contribuições das áreas de sistemas embarcados e qualidade de *software*.
- ii. Definição dos conceitos de sistema embarcado e de *software* embarcado a serem adotados nesta tese.
- iii. Identificação da fronteira entre *software* e *hardware* em um sistema embarcado.
- iv. Estudo do ciclo de vida em sistemas embarcados para a identificação de mecanismos de união entre *hardware* e *software* no processo de desenvolvimento de sistemas embarcados.
- v. Utilização de processos de desenvolvimento de sistemas embarcados pela indústria.

1.4.2 Elaboração da tese

- i. Elaboração do texto.
- ii. Elaboração de um processo para o desenvolvimento de sistemas embarcados.
- iii. Detalhamento das fases do processo de desenvolvimento de sistemas embarcados.
- iv. Descrição das atividades do processo.
- v. Elaboração dos artefatos do processo.
- vi. Elaboração dos fluxogramas das fases e das atividades do processo.
- vii. Atendimento ao modelo CMMI-DEV pelo processo.

- viii. Atendimento aos documentos ECSS-E-ST-40 Engenharia espacial e ECSS-Q-ST-80C Garantia do Produto Espacial – Garantia do Produto de *Software* do padrão ECSS pelo processo.
- ix. Providência e análise dos estudos de caso e apresentação de seus resultados.

1.5 Estrutura da tese

O Capítulo 2 apresenta os fundamentos conceituais que influenciarão o entendimento e a proposta apresentada; o Capítulo 3 descreve modelos de qualidade, em especial o CMMI e sua aderência ao MPSBr; o Capítulo 4 apresenta, de forma introdutória, a área espacial; o Capítulo 5 apresenta os sistemas embarcados e alguns problemas clássicos; o Capítulo 6 apresenta os trabalhos correlatos; o Capítulo 7 apresenta o trabalho realizado por esta tese; e o Capítulo 8 apresenta as considerações finais.

As figuras e tabelas apresentadas nesta tese sem indicação de fonte foram produzidas pelo autor.

O aumento do uso de produtos que contêm algum componente de software embarcado e o número de produtos que utilizam tais tecnologias demandaram o desenvolvimento de produtos de maior qualidade. Esta tese se propõe realizar estudos para a elaboração de um processo para o desenvolvimento de sistemas embarcados que atendam às características específicas dos sistemas embarcados, buscando a melhoria da qualidade do produto final utilizando boas práticas para sua construção/desenvolvimento.

O estudo dos fundamentos conceituais apresentados a seguir é imprescindível para identificar e definir os conceitos aqui utilizados.

2 FUNDAMENTOS CONCEITUAIS

A busca pela qualidade dos produtos é antiga, por isso, muitos estudos foram realizados com essa finalidade. Para o desenvolvimento desta tese, foram realizadas a identificação dos conceitos básicos necessários para a definição da estrutura e a organização dos fundamentos nos quais se baseia este estudo e que serão utilizados em todas as etapas do projeto.

2.1 Termos comuns e definições

Os termos e conceitos norteadores desta tese serão apresentados a seguir e utilizados ao longo do trabalho, tendo como objetivo normalizar suas definições básicas. Os demais conceitos estão dispostos no Apêndice A.

Projeto

Um projeto é um empreendimento temporário cujo objetivo é criar um produto ou serviço único. Neste caso, por *temporário* entende-se que todo projeto tem um início e um fim definidos, o que significa que o produto ou serviço é diferente de todos os produtos ou serviços executados anteriormente (PMBOK, 2004).

Sistema

Um sistema é composto de equipamentos, habilidades e técnicas capazes de desempenhar ou prover suporte a um papel operacional. Incluem instalações, equipamentos, materiais, serviços, *software*, dados técnicos e pessoais para sua operação até o ponto de poder ser considerado uma unidade autossuficiente (LOUREIRO, 2008).

Modelagem de sistemas

Modelagem de sistemas é a abstração de um sistema que o transforma em partes com o objetivo de promover melhor entendimento e identificação de independência dessas partes,

auxiliando a análise e o design, favorecendo a comunicação e a compreensão das declarações textuais e diagramáticas (SOUZA, 2006).

Sistema embarcado

Sistema embarcado é um produto formado por um conjunto de componentes encapsulados ou dedicados ao dispositivo ou sistema que ele controla, tendo componentes de *hardware* e componentes de *software* (HERBSTER, 2006).

Componente embarcado¹

Componente embarcado é um dispositivo que pode ser item de um sistema embarcado, que, muitas vezes, tem em sua concepção parte *hardware* e parte *software*.

Componentes externos²

Um componente externo é um item que foi ou está sendo produzido externamente ao projeto corrente da organização ou por uma empresa contratada.

Software embarcado³

Entende-se *software* como sinônimo de produto de *software*, que é o conjunto de programas de computador, procedimentos e possível documentação e dados associados (SOFTEX, 2010).

Software embarcado é a parte de um componente embarcado desenvolvida com recurso de programação computacional.

¹ Definido pelo autor para ser utilizado nesta tese

² Definido pelo autor para ser utilizado nesta tese

³ Definido pelo autor para ser utilizado nesta tese

Sistema crítico

Característica e condições de um produto ou sistema cujos parâmetros, requisitos ou itens exigem atenção especial para que seus objetivos possam ser cumpridos. A Norma ECSS define, em seu anexo D, Categorias de *Software* Crítico, apresentadas na Tabela 2.1.

Tabela 2-1 – Categorias de criticidade do software (adaptada de ECSS, 2009b)

Categoria	Definição
A	<i>Softwares que se não forem executados, ou se não forem corretamente executadas, ou cujo comportamento anômalo pode causar ou contribuir para uma falha no sistema, resultando em: consequências catastróficas.</i>
B	<i>Softwares que se não forem executados, ou se não forem corretamente executadas, ou cujo comportamento anômalo pode causar ou contribuir para uma falha no sistema, resultando em consequências críticas.</i>
C	<i>Softwares que se não forem executados, ou se não forem corretamente executadas, ou cujo comportamento anômalo pode causar ou contribuir para uma falha no sistema, resultando em principais consequências.</i>
D	<i>Softwares que se não forem executados, ou se não forem corretamente executadas, ou cujo comportamento anômalo pode causar ou contribuir para uma falha no sistema, resultando em consequências menores ou negligenciáveis.</i>

Fonte: adaptada ECSS (2009).

Defeito

Uma característica de um sistema que poderá levá-lo a um erro (SOMMERVILLE, 2011).

Um problema de qualidade encontrado depois de o *software* ser liberado para o uso (PRESSMAN, 2010).

Uma falta é atribuída à suposta causa do erro (SOUZA, 2002).

É um ato inconsistente cometido por um indivíduo ao tentar entender determinada informação, resolver um problema ou utilizar um método ou uma ferramenta (DEV MEDIA, 2013).

Um passo incorreto, processo ou definição de dados em um programa de computador (IEEE, 1990).

Erro

Um problema de qualidade encontrado antes de o *software* ser liberado para o uso (PRESSMAN, 2010).

Um erro ocorre quando parte do estado do sistema pode levar à falha (SOUZA, 2002).

É uma manifestação concreta de um defeito num artefato de *software*. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro (DEVMEDIA, 2013).

Um passo, processo ou definição de dados incorretos, um resultado incorreto. Uma ação humana que produz um resultado incorreto (IEEE, 1990).

Falha

É a falta de conformidade com os requisitos de *software* (PRESSMAN, 2010).

Um evento que ocorre em algum momento em que o sistema não fornece o resultado de um serviço como esperado (SOMMERVILLE, 2011).

Uma falha ocorre quando o comportamento do sistema desvia-se da especificação (SOUZA, 2002).

É o comportamento operacional do *software* diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros, e alguns erros podem nunca causar uma falha (DEVMEDIA, 2013).

A inabilidade de um sistema ou componente para desempenhar as funções necessárias dentro de requisitos de desempenho especificados. Uma física ou funcional manifestação de uma falha que resulta na incapacidade completa de um sistema ou componente para funcionar (IEEE, 1990).

Modelo

Um modelo é considerado um guia ou manual das melhores práticas encontradas por organizações de sucesso, dada uma área de aplicação. Ele simplesmente recomenda ações, não define passos ou ferramentas para a execução de uma atividade. Um modelo não é um programa de melhoria de processos; ele apresenta as observações que devem ser feitas na elaboração de um programa. (SOUZA, 2006).

Modelo de processo ou modelo de processo de desenvolvimento de *software*

Um modelo de processo de desenvolvimento de *software*, ou simplesmente modelo de processo, é uma representação ou abstração simplificada do processo de *software*, com os objetos e atividades envolvidos com perspectivas e abrangência para representar o gerenciamento do processo de *software* e, conseqüentemente, o progresso do projeto. (ARRUDA; COUTO; SOUZA, 2008)

2.2 Desenvolvimento de *software*

O processo de desenvolvimento de *software* é um conjunto de atividades coerentes e consistentes utilizadas para especificar, projetar, implementar e testar um *software*. O modelo de processo de desenvolvimento é uma representação abstrata de como deverá ser realizada a construção do *software*. Os diferentes conjuntos de etapas são denominados *ciclos de vida*, e cada uma delas conta com diferentes atividades e uma estrutura básica comum

O modelo mais apresentado academicamente é o *cascata*, também denominado *clássico* ou *linear*, como apresentado na Figura 2.1.

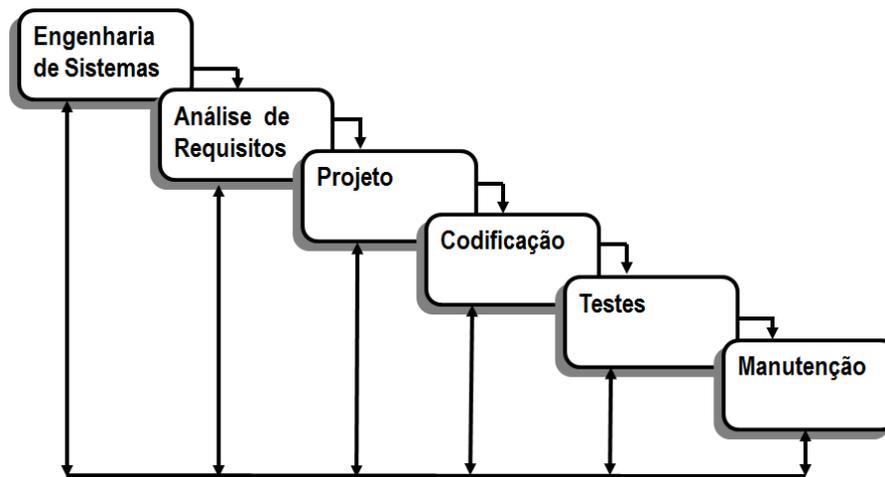


Figura 2.1 – Representação modelo clássico

Fonte: adaptado de PRESSMAN (2006).

2.2.1 Descrição das fases

- **Análise e a engenharia de sistemas:** envolvem a coleta de requisitos em nível do sistema, pequena quantidade de projeto e análise de alto nível que proporcionam uma visão essencial, identificando no *software* quem deve fazer interface com outros elementos (*hardware*, pessoas e banco de dados). Nesta fase, elabora-se o plano de desenvolvimento de *software*.
- **Análise de requisitos de *software*:** nesta fase, o processo de coleta dos requisitos é intensificado e concentrado mais especificamente nos requisitos do *software*. Deve-se ter o domínio da informação, a função, o desempenho e interfaces em que os requisitos de sistema e de *software* são documentados e revistos com o cliente.
- **Projeto:** é a tradução dos requisitos do *software* para um conjunto de representações que podem ser avaliadas quanto à qualidade antes que a codificação se inicie. Em geral, se concentra em quatro atributos do *software*: estrutura de dados, arquitetura de *software*, detalhes procedimentais e caracterização de interfaces.
- **Codificação:** nesta fase, faz-se a tradução das representações do projeto para uma linguagem “artificial”, resultando em instruções executáveis pelo computador, ou seja, o programa ou *software*.

- **Testes:** concentram-se em dois aspectos primordiais: aspectos lógicos internos do *software*, nos quais se verifica se todas as instruções foram testadas; aspectos funcionais externos, que direcionam os esforços para descobrir erros e garantir que a entrada definida produza os resultados esperados.
- **Manutenção:** controla e suporta as mudanças que o *software* deverá sofrer depois que for entregue ao cliente. Em geral, as causas das mudanças são os erros, a necessidade de adaptação do *software* para acomodar alterações sem seu ambiente externo e a exigência do cliente quanto a acréscimos funcionais e de desempenho.

O próprio significado do termo *engenharia* faz inferências aos conceitos das fases da engenharia convencional: *criação, construção, análise, desenvolvimento e manutenção*, que direcionam a concentração das atividades da engenharia de *software* para os aspectos da produção de um sistema de *software*.

Com o aumento da necessidade de sistemas em sua maioria embarcados, mais sofisticados e complexos, e a grande interação dos *softwares* com os seres humanos, em muitos casos até substituindo-os em atividades de risco, os *softwares* precisam ser cada vez mais seguros e confiáveis.

Alguns problemas de projeto estão relacionados com a engenharia de *software*, em que são apontados problemas de sistema e de *software* que remetem às práticas e ao uso inadequado de modelos e ao processo de desenvolvimento de sistemas.

2.3 Processo

Um processo é uma sequência de passos realizados para um dado propósito. Colocado de maneira mais simples, processo é aquilo que você faz. [...] Processo é aquilo que as pessoas fazem usando procedimentos, métodos, ferramentas e equipamentos para transformar matéria-prima (entrada) em produto (saída) que tenha valor para o cliente (PAULK et al., 1995 apud SPINOLA e PESSÔA, 1997).

2.3.1 Processo de desenvolvimento de *software*

Um processo de desenvolvimento de *software* é um conjunto de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter *softwares* e seus produtos relacionados (SEI, 2010).

Há outras definições de processo:

Maneiras pelas quais se realiza uma operação segundo determinadas normas (PADUA, 2009).

Um arcabouço para as tarefas que são necessárias para construir *softwares* de alta qualidade (PRESSMAN, 2011).

Descrição documentada de um conjunto de atividades executadas para alcançar um determinado objetivo (PRESSMAN, 2011).

O *Software Engineering Institute (SEI)* identifica três dimensões críticas, segundo as quais as organizações devem direcionar seus esforços, a saber pessoas, procedimentos e métodos, ferramentas e equipamentos, conforme apresentado na Figura 2.2.

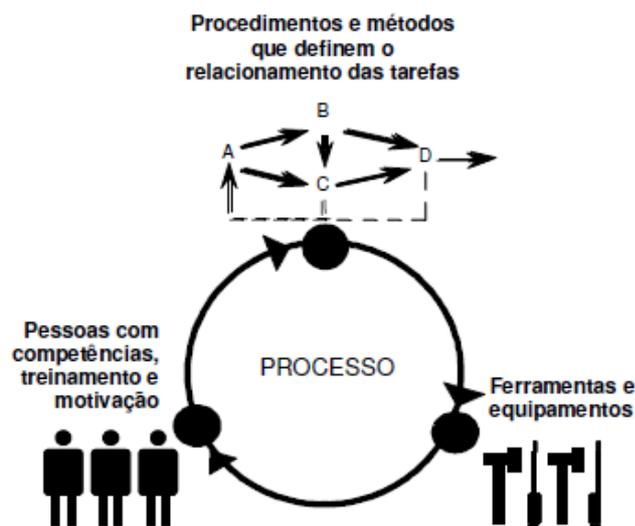


Figura 2.2 – Dimensões críticas de um processo

Fonte: SEI (2010).

As três dimensões críticas são:

- **Pessoas com competências, treinamento e motivação:** todo processo para conseguir o resultado almejado deve ser aceito, entendido e estimulado pelas pessoas nele envolvidas, pois, de maneira geral, só se pode desenvolver com sucesso aquilo em que se acredita.
- **Ferramentas e equipamentos:** de forma controlada e evolutiva, deve ser inserido o uso de ferramentas que auxiliem na execução do processo, com a responsabilidade de fazê-las ser dominadas por pessoas e utilizadas nessa execução sem que tal uso desestime o conhecimento ou o envolvimento das pessoas com o referido processo.
- **Processos e métodos que definem o relacionamento entre as tarefas:** o processo, para ser instituído em uma empresa, deve ser elaborado de forma a direcionar naturalmente para a realização das boas práticas. Deve ser minimamente burocratizado, além de permitir certa flexibilidade.

Atualmente, existem diversos modelos de processo de desenvolvimento de *software*, que têm como objetivo auxiliar as empresas na organização e na disciplina dessas atividades de desenvolvimento, visando aumentar a qualidade de seus produtos e de sua produtividade.

A evolução do produto de *software* no tempo é distribuída em fases, iniciando quando um produto é idealizado e concebido, e terminando quando o produto não está mais disponível para uso.

2.3.2 Ciclos de vida de desenvolvimento de software

Ciclo de vida é a estrutura que contém processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de *software*, abrangendo a vida do sistema desde a definição de seus requisitos até o término de seu uso (ISO/IEC 12207, 2008).

O modelo de ciclo de vida do *software* consiste em todas as atividades, desde sua criação até sua “aposentadoria”. A definição de um modelo de ciclo de vida influencia diretamente no sucesso do desenvolvimento do produto, pois sua estrutura, quando bem elaborada, permite a

identificação, a especificação e o controle dos requisitos solicitados pelo cliente, que é um dos mais fortes fatores de avaliação da qualidade do produto, além de minimizar os riscos inerentes ao processo, ou seja, é a subdivisão do tempo de vida de um produto ou projeto em fases.

Em geral, produtos de *software* são desenvolvidos para diferentes empresas, sendo por isso importante a adequação do ciclo de vida para cada caso (SEI, 2010).

A distribuição do ciclo de vida do *software* em diferentes atividades, quando estas são executadas de forma estruturada, produz em todas elas artefatos controláveis, além de permitir o gerenciamento do desenvolvimento do produto (GUERRA; COLOMBO, 2008).

2.4 Processo de *Software* para equipes - *Team Software Process (TSP)*

Com o aumento da demanda por produtos de *software*, a qualidade do *software* é uma disciplina que está conquistando grande importância na engenharia de *software*. Nos primeiros estágios, seu foco estava em melhorar os processos da organização e, na sequência, passou a se preocupar com as equipes de desenvolvimento.

O *Team Software Process (TSP)* foi criado pelo *Software Engineering Institute (SEI)* para auxiliar as equipes a melhorar a gestão dos processos de desenvolvimento e, assim, garantir a qualidade de seus produtos de *software* (AMARAL, 2009; AMARAL; FARIA, 2010).

Sua versão inicial foi desenvolvida em 1996, por Watts Humphrey, com o objetivo de fornecer um processo operacional, de orientar os engenheiros a sempre desenvolver um trabalho de qualidade. É baseado no *Capability Maturity Model (CMM, Modelo de Maturidade da Capacidade)*, estrutura criada para promover a melhoria quantitativa de processo de *software* que ajuda equipes a desenvolver produtos de *software* de maneira eficaz.

O *TSP* pertence a uma estrutura integrada com os métodos *Personal Software Process (PSP)* e *CMM* para a produção de equipes eficazes. Conceitualmente, o *CMM* e *PSP* fornecem o contexto de habilidades de engenharia eficazes, enquanto o *TSP* fornece um guia de trabalho eficiente, conforme apresentando na Figura 2.3.

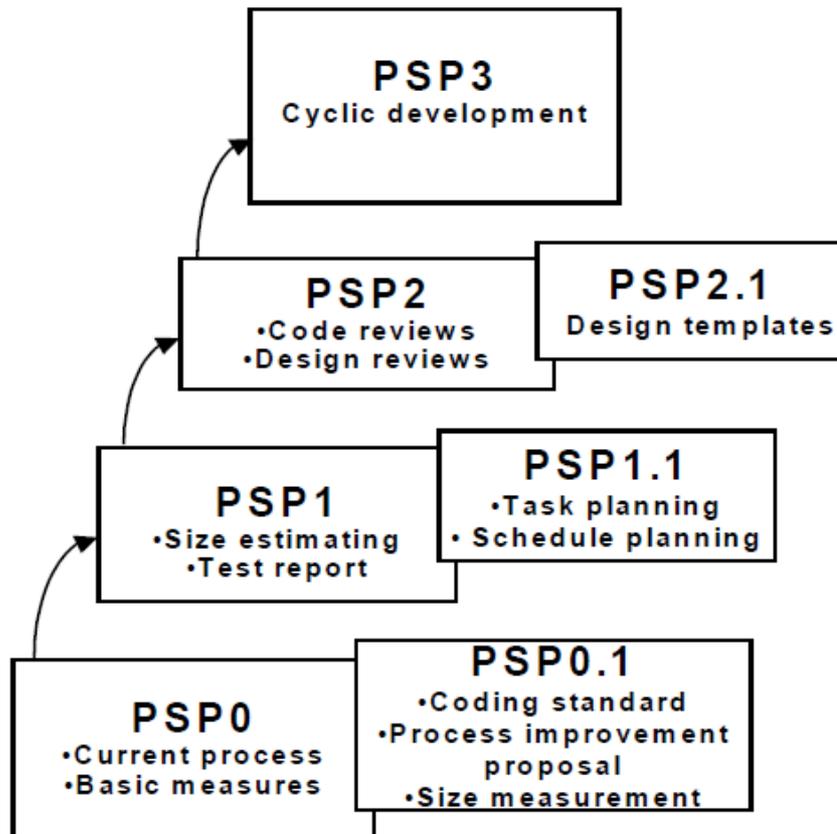


Figura 2.3 – Estrutura do TSP/PSP

Fonte: HUMPHREY (2000).

O *PSP* baseia-se no fato de que a equipe é dona de seus processos e pode mudá-los sempre que necessário, e de que tais processos são calcados na experiência, no conhecimento e na maturidade dos membros da equipe. As equipes devem aplicar as práticas de CMM (HUMPHREY, 2000).

O *TSP* provê um conjunto de *scripts* de processos, formulários, métodos e métricas que guiam os desenvolvedores na criação de equipes capazes.

A estrutura do *TSP* é simples. Ela é constituída de desenvolvimento incremental e métricas padronizadas de qualidade e *performance* para indivíduos e equipes, definição e avaliações de papéis e de time, e processo disciplinado. Essa estrutura está representada na Figura 2.4.

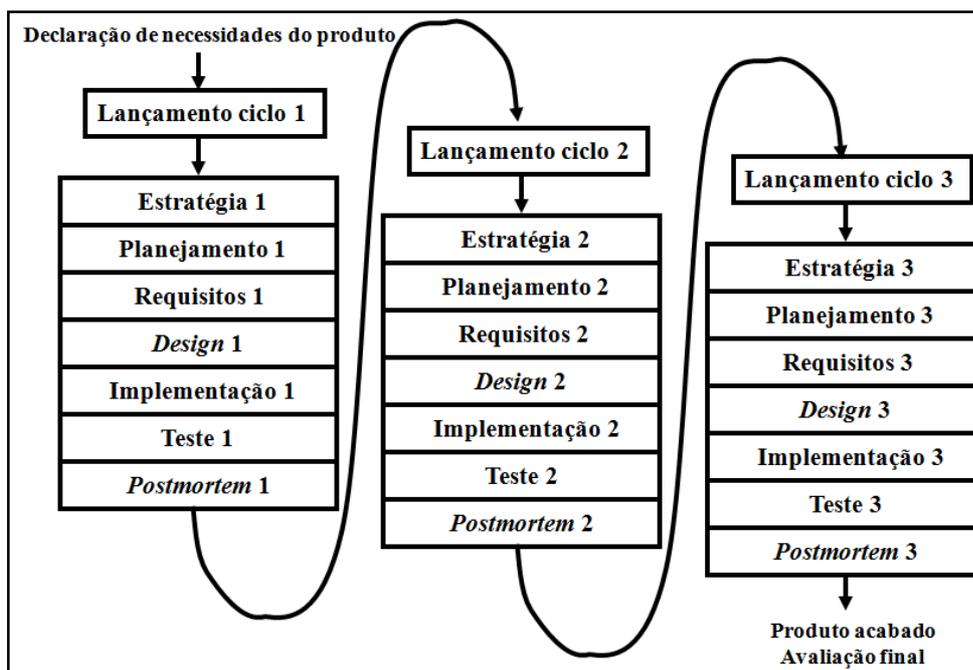


Figura 2.4 – Funcionamento do TSP

Fonte: BELLOQUIN (2005).

Com a globalização, a terceirização e o desenvolvimento distribuído de *software* estão se tornando mais comuns. Nesse contexto, o *TSP* pode auxiliar as equipes de desenvolvimento de produtos de *software* a melhorar os níveis de qualidade e produtividade (LÓPEZ et al., 2010).

2.5 SCRUM

O SCRUM é um processo de desenvolvimento de *software* iterativo e incremental para o gerenciamento de projeto se o desenvolvimento de *software* ágil. Não se trata de um processo completo; é um *framework* que faz transparecer suas práticas de desenvolvimento com a intenção de melhorá-las. O nome desse processo vem de uma jogada do rúgbi que mostra que o trabalho em equipe é fundamental.

É sustentado por três pilares de controle de processos empíricos:

- **Transparência:** garante que os aspectos do processo que afetam o resultado sejam visíveis para aqueles que gerenciam os resultados.

- **Inspeção:** os diversos aspectos do processo devem ser inspecionados com frequência suficiente para que variações inaceitáveis no processo possam ser detectadas.
- **Adaptação:** identificação e ajustes por meio da inspeção de aspectos do processo que não atendam aos limites aceitáveis do resultado (BELNUOVO et al., 2010).

O SCRUM é baseado em pequenas equipes e permite melhor comunicação entre seus membros, conforme a estrutura apresentada na Figura 2.5.

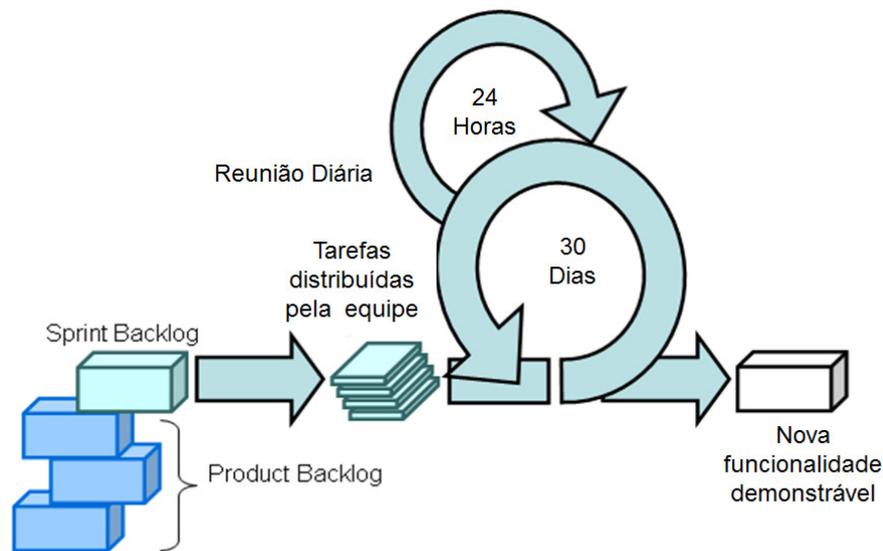


Figura 2.5 – Estrutura do SCRUM

Fonte: adaptado BELNUOVO et al (2010).

- **Sprint:** unidade básica de desenvolvimento; dura, geralmente, de uma semana a um mês, dependendo do planejamento.
- **Planejamento:** momento de identificação e organização das tarefas para o *sprint*, para estimativa e compromisso para sua realização.
- **Reunião:** pode ser de revisão ou de retrospectiva. Nela, o progresso é analisado e revisto, e as lições aprendidas para os próximos *sprints* são identificadas e registradas.
- **Product backlog:** conjunto de funcionalidades definidas pelas partes envolvidas.

- **Backlog:** funcionalidade a ser desenvolvida em um *sprint* e pertence ao *product backlog*.
- **Product owner:** usuário envolvido com a equipe e o desenvolvimento. Qualidade de *software*.

2.6 Qualidade de *software*

A definição para este conceito, segundo a ISO 8402, é “a totalidade das características que sustentam sua habilidade em satisfazer necessidades declaradas ou implícitas” (GUERRA, 2009).

A qualidade de *software* é uma das áreas de estudo da engenharia de *software* e busca estruturar os processos de desenvolvimento de *software* para que possam atender à premissa definida pela ISO 8402, garantindo dessa forma que especificações e necessidades explícitas e implícitas estejam presentes no produto desenvolvido, empregando processos normatizados, desde sua concepção até sua finalização (GUERRA, 2009).

A preocupação com o uso de padrões teve origem com a *Tower Bridge*, ponte basculante projetada por Sir John Wolfe-Barry (Figura 2.6), o que estimulou o estabelecimento de um comitê para a normatização em janeiro de 1901. Dessa data até 1914, a normatização foi aplicada aos perfis de ferrovias. Nos anos 1920, foi introduzida em países como Canadá, Austrália, África do Sul e Nova Zelândia, tendo sido identificado interesse também da Alemanha e dos Estados Unidos nesse assunto. Seu crescimento deu-se durante a Primeira Guerra Mundial (de 1914 a 1945), quando a normatização foi utilizada pela Marinha Britânica e pelo Comando de Guerra, entre outros.

Entre os anos 1950 e 1960, o uso de normas foi adotado pelo mercado de consumo com o objetivo de atender aos interesses dos consumidores finais. Em 1987, a ISO publicou a Série 9000 e, em 2005, a ISO 27001, com foco na segurança da informação, além da ISO 20000, para Gestão de Serviços de TI (BSI BRASIL, 2006).



Figura 2.6 – *Tower Bridge*, em Londres.

Em 1968, motivado pelo aumento da importância de sistemas de *software* de computador em muitas atividades da sociedade, foi criado o *Nato Science Committee* para acompanhar os aspectos técnicos e gerais relativos ao impacto da engenharia de *software* na sociedade.

A maior causa da crise do *software* é que as máquinas tornaram-se várias ordens de magnitude mais potentes. Em termos diretos, enquanto não havia máquinas, programar não era um problema; quando tivemos computadores fracos, isso se tornou um problema pequeno, e agora que temos computadores gigantescos, programar tornou-se um problema gigantesco (DIJKSTRA, 1972 apud BERGMANN, 2008; KOSCIANSKI, 2007).

Nos anos 1980, auge da crise do *software*, o Departamento de Defesa dos Estados Unidos (DoD) estimulou a *Software Engineering Institute (SEI)* a delinear as características de um *software* maduro. O objetivo era fornecer parâmetros para que as empresas de *software* avaliassem sua capacidade de realizar seus processos e também para que as organizações contratantes, como o governo dos Estados Unidos, analisassem a capacidade dos subcontratados. Para atender a essa infraestrutura do processo, o SEI desenvolveu o *Capability*

Maturity Model para *Software* (CMM), que delinea as características de um processo de *software* maduro e capaz. A estrutura e o conceito do modelo foram desenvolvidos por Watts Humphrey (PAULK et al., 1993) e esse objetivo foi cumprido com o estímulo de programas de pesquisa, treinamentos e desenvolvimento profissional (PAULK et al., 1993).

A primeira definição de engenharia de *software* foi proposta por Fritz Bauer em 1969, como segue:

O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um *software* que seja confiável e que funcione eficientemente em máquinas reais (NATO, 1969).

2.6.1 Qualidade do produto

A qualidade do produto é aferida por meio de um conjunto de parâmetros que define determinadas características a serem alcançadas para que o produto atenda às expectativas de seus usuários. Assim, se faz necessária a escolha de um modelo que organize os atributos e permita avaliar a qualidade do *software* (ROCHA; MALDONADO; WEBER, 2001 apud CARRO; WAGNER, 2003).

A qualidade de produto de *software* baseia-se nos fatores da qualidade em duas etapas: a primeira se inicia com a identificação dos fatores que definem a qualidade desejada; a segunda, dos fatores a serem identificados (HAYES, 1998 apud SPINOLA, 1998).

Toda empresa desenvolvedora de *software* necessita de mecanismos que lhe permitam acompanhar o nível de satisfação de seus clientes, bem como as tendências do mercado de forma geral; por isso, necessita conhecer os parâmetros atuais que definem qualidade e seus processos.

A engenharia de *software* oferece três aspectos básicos que devem ser considerados ao se avaliar a qualidade de um produto de *software*: suas características operacionais (operação do produto), sua facilidade para ser adaptado (revisão do produto) e sua ambientação (transição do produto). Esses aspectos são apresentados na Figura 2.7 (MCCALL; RICHARDS; WALTERS, 1977 apud PRESSMAN, 2011; PÁSCOA, 2002; GUERRA, 2009).

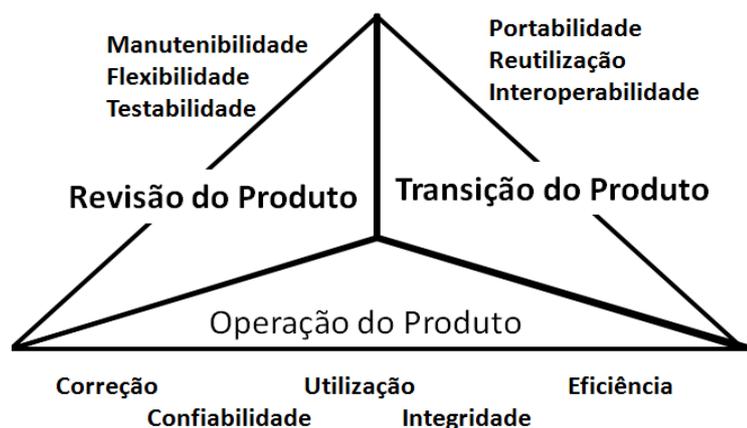


Figura 2.7 – Fatores de qualidade

Fonte: MCCALL, RICHARDS e WALTER, (1997) apud PRESSMAN (2011).

Assim, os fatores de qualidade são:

- **Correção (corretude):** o quanto o *software* atende às especificações do cliente.
- **Confiabilidade:** o quanto o *software* executa as funções desejadas da forma esperada.
- **Eficiência:** quanto de recurso computacional é utilizado para que o *software* realize sua função.
- **Integridade:** em que medida o acesso ao *software* ou aos dados pode ser controlado.
- **Utilização (usabilidade):** qual a facilidade, isto é, quanto de dedicação é necessário, para o aprendizado na utilização completa do *software*.
- **Manutenibilidade:** quanto de dedicação é necessário para identificar e corrigir um erro no *software*.
- **Flexibilidade:** quanto de dedicação é necessário para modificar uma funcionalidade do *software*.
- **Testabilidade:** quanto de dedicação é necessário para testar o *software* para garantir a realização de suas funcionalidades.

- **Portabilidade:** quanto de dedicação é necessário para modificar a plataforma do *software*.
- **Reusabilidade:** quanto o *software* (ou partes dele) pode ser utilizado ou acoplado a outras aplicações.
- **Interoperabilidade:** quanto de dedicação é necessário para interfaceá-lo com outro *software*.

De maneira geral, a qualidade de um produto é medida a partir de sua conformidade com requisitos e da satisfação com seu desempenho e a ausência de defeitos, erros ou falhas. Qualidade total é atender aos requisitos do cliente com eficiência (GUERRA, 2009).

2.6.2 Qualidade do processo

Com o avanço dos recursos computacionais e a facilidade para adquiri-los, os produtos que dispõem desses recursos passaram a fazer parte de nossa rotina, desencadeando um aumento na fabricação de produtos de *software*. Os sistemas baseados em computador estão sendo incorporados às mais diversas áreas do cotidiano e, sem que percebamos, muitas vezes em situações nas quais não pode haver falhas. Estima-se que se um sistema de uso global, isto é, do qual a humanidade seja dependente, deixar de funcionar, 40% da população mundial sofrerá as consequências do problema (RED, 2000 apud CAROSIA, 2004).

Assim, melhorar a qualidade do *software* é o principal objetivo da engenharia de *software*. Observou-se, devido às características peculiares dos produtos de *software*, que postergar as práticas de qualidade do *software* para a avaliação do produto final pode ser comprometedora e perigosa e, em muitos casos, crítico. Para alcançar seu objetivo, a tendência da engenharia de *software* é prevenir as falhas no processo de desenvolvimento do *software* em vez de consertá-las no produto final. Por isso, a qualidade do produto de *software* passou a ser responsabilidade desse processo.

Ao ser desenvolvido, todo produto deve ter descritas as características de seus requisitos de funcionamento e seus requisitos de qualidade desde o início de sua concepção, de modo a permitir que tais características sejam consideradas no produto final.

A falta de conformidade com os requisitos é falta de qualidade. Além desses requisitos, existem padrões específicos que definem os critérios de desenvolvimento que servem de diretriz para o *software* que está sendo produzido. Se esses critérios não forem seguidos, resultará na falta de qualidade (PRESSMAN, 2011).

Por definição, processo de *software* é um conjunto de atividades, métodos, práticas e ferramentas utilizados para desenvolver e manter produtos de *software*. Em geral, esse processo é realizado ou controlado por pessoas com formação, comportamentos, habilidades e experiências diferentes. Isso reforça a necessidade de definição de atividades que aconteçam de forma sistemática e passível de repetição, independentemente de quem as execute, exigindo, portanto, a criação e a adoção de padrões de processo.

Os padrões de desenvolvimento de uma organização devem seguir critérios institucionais, com objetivos claros e estratégias que permitam o compartilhamento das experiências e lições aprendidas. Conforme esses critérios vão sendo mais bem definidos, com implementação mais fácil, estruturada e mais consistente em toda a organização, o processo alcança padrões de maturidade, a saber:

- **Capabilidade do processo:** descreve um conjunto de resultados que podem ser alcançados com a utilização do processo, tornando-os previsíveis.
- **Desempenho do processo:** representa os resultados alcançados pela utilização do processo.
- **Maturidade do processo:** representa o potencial de evolução da capacidade e seu uso consistente em todos os projetos (GONÇALVES, 2001).

3 MODELOS DE QUALIDADE

Como vimos no Capítulo 2, a qualidade do produto é aquilo que buscamos e a qualidade do processo é o que fazemos para que isso aconteça. Esta tese busca a qualidade do produto por meio da utilização de processos para o desenvolvimento de componentes para sistemas embarcados, portanto, a organização e a apresentação dos conceitos fundamentais de aspecto geral utilizados para o desenvolvimento são fundamentais para normalizar o entendimento de seu uso e de sua influência nos modelos de qualidade apresentados neste capítulo.

Segundo a norma NBR ISO 8402, *qualidade* pode ser definida como “totalidade de características de um produto de *software* que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas” (ABNT01, 1994).

Necessidades explícitas são aquelas solicitadas pelos clientes na definição dos requisitos. Refletem as características e as condições para o produto ser utilizado, atendendo aos objetivos, às funções e ao desempenho esperado. *Necessidades implícitas* são os requisitos que não foram solicitados pelo cliente, por serem transparentes ao seu conhecimento ou serem óbvias a sua necessidade, mas precisam ser atendidas pelo *software* (IEEE, 1990).

Segundo Koscianski (2007), a qualidade é relativa. O que é qualidade para uma pessoa pode ser falta de qualidade para outra. Já para Sommerville (2011), o controle de qualidade se dá com a definição e a aprovação de processos para que os procedimentos e padrões de qualidade sejam seguidos pelos desenvolvedores do *software*, enquanto para Pressman (2011), a garantia de qualidade de *software* envolve atividades como a aplicação de métodos técnicos, a realização de revisões técnicas formais e de atividades de testes de *software*, a aplicação de padrões, o controle de mudanças, a medição e a manutenção de registros e reportagem.

Ocorrida durante a década de 1990, a *crise do software* foi um período de grande preocupação que resultou em estudos e na aplicação de modelos de melhoria do processo para o desenvolvimento de produto de *software*.

Com a finalidade de construir *softwares* dentro do prazo estimado, com custo adequado e atendendo às expectativas de funcionamento do solicitante, foram desenvolvidos alguns modelos e normas (Figura 3.1), que apresenta os principais componentes que devem ser observados em um modelo de qualidade de um produto de *software*.

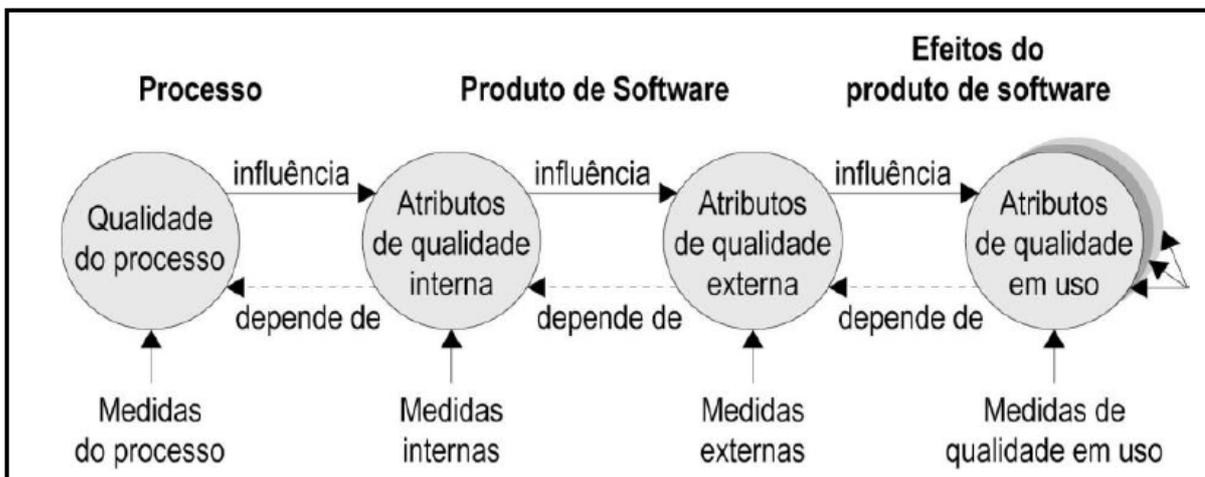


Figura 3.1 – Qualidade de *software* durante seu desenvolvimento

Fonte: ISO 9126 (2003) apud GUERRA (2009), REY (2003).

Conforme apresentado no Capítulo 2, os modelos desenvolvidos estão organizados em duas categorias:

- **Qualidade de produto de *software***: tem como objetivo a garantia da qualidade do produto gerado. Entre outros, podem ser citados ISO/IEC 9126, ISO/IEC 12119, ISO/IEC 14598, Projeto SQuaRE.
- **Qualidade de processo de *software***: definição, avaliação e melhoria dos processos de desenvolvimento de *software*. Entre eles, podemos citar o *Capability Maturity Model (CMM)*, *Capability Maturity Model Integrated-CMMI*, *Software Process Improvement and Capability dEtermination (SPICE, Padrão Internacional para Avaliação de Processos de Software)* (CARVALHO et al., 2009).

3.1 Qualidade de processo de software

Atualmente, as pesquisas e os estudos para alcançar a qualidade de um *software* estão voltados para seus processos de desenvolvimento. Isso significa que garantir a qualidade do processo é um passo importante para obter um produto de qualidade. Os processos podem ser definidos para satisfazer atividades como desenvolvimento, manutenção, aquisição e contratação de *software*.

Muitas vezes, a qualidade de *software* no mundo está diretamente relacionada ao CMMI, e, no Brasil, remete também ao MPS.BR, ambos inspirados nas normas ISO/IEC 12207 e ISO/IEC 15504.

A norma ISO/IEC 12207 estabelece uma estrutura comum para os processos de ciclo de vida de *software*, como orientar a compreensão de todos os componentes presentes na aquisição e fornecimento de *software* (ISO 12207, 2008); já a norma ISO/IEC 15504 é utilizada como estrutura para modelos e métodos de avaliação de processo (ISO 15504, 2008).

3.2 CMMI V1.3 (*Capability Maturity Model Integration – Integração de Modelos de Maturidade da Capacidade*)

O modelo *Capability Maturity Model (CMM)* surgiu na década de 1980, por solicitação do Departamento de Defesa Norte-Americano para avaliar a capacidade dos fornecedores de *software* contratados por eles. Foi desenvolvido pelo *Software Engineering Institute (SEI)* utilizando os conceitos de qualidade total instituídos por Humphrey (2000) e sua equipe no SEI.

O CMMI é uma integração de três modelos do CMM: o SW-CMM, o *System Engineering Capability Model (SECM)* e o *Integrated Product Development Capability Maturity Model (IPD-CMM)*. Reduz custos e inconsistências, garantindo aderência à norma ISO/IEC 15504. Esse modelo é composto de cinco níveis de maturidade, e as empresas que não utilizam processos para seu desenvolvimento ou cujos processos são artesanais estão no nível 1. Conforme os processos são utilizados e dedicam-se a sua melhoria, são conquistados novos níveis de classificação, o que representa um amadurecimento de seu processo de desenvolvimento de *software*, começando pelo nível 2 até conquistar o nível 5, que é o mais

alto na escala do CMMI. Todos os níveis estão divididos em áreas de processos, organizadas de forma a proporcionar o alcance do objetivo de cada nível.

O *Software Engineering Institute (SEI)*, administrado pela Universidade de *Carnegie Mellon*, em *Pittsburgh*, nos Estados Unidos, e patrocinado pelo Departamento de Defesa Norte-Americano (DoD), tem por objetivo oferecer condições para o desenvolvimento das boas práticas da engenharia de *software* e, nos projetos desenvolvidos por seus fornecedores, conquistar o mesmo nível de repetibilidade, sucesso e controle encontrado em outros setores da atividade industrial.

O CMMI baseia-se no conceito de maturidade dos processos de *software*. Foi inspirado nas técnicas de *Total Quality Management (TQM)* e é compatível com as normas ISO/IEC 15505 e ISO/IEC 12207.

Tem cinco níveis de maturidade, em uma escala crescente de controle e visibilidade que permite a classificação de processos, dos resultados técnicos e gerências do projeto de *software*. Sua estrutura é apresentada na Figura 3.2.

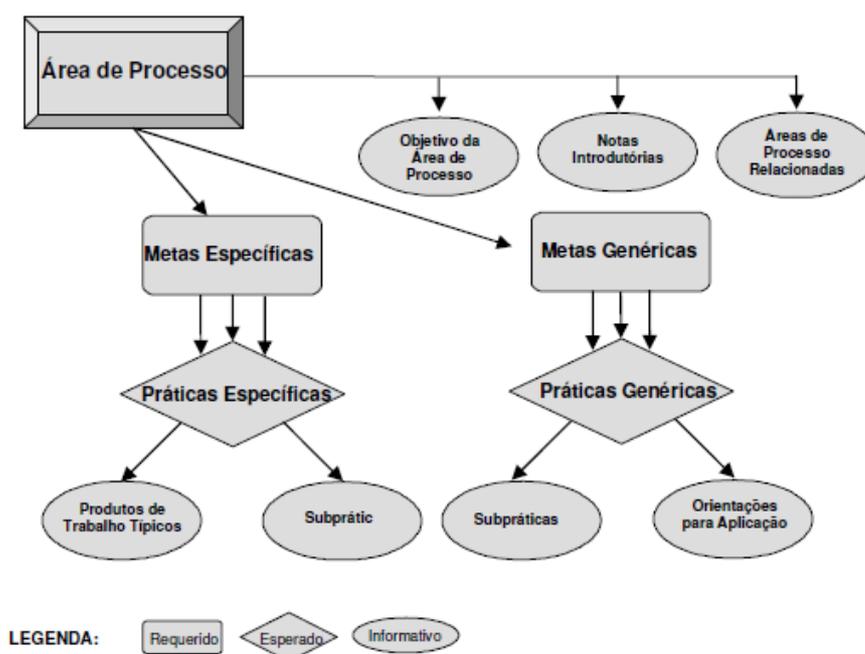


Figura 3.2 – Componentes do modelo CMMI (SEI, 2010)

Fonte: SEI (2010).

3.2.1 Componentes do modelo CMMI V1.3

Os componentes do modelo CMMI V1.3 (ver Figura 3.2) cumprem a seguinte finalidade:

- **Níveis de maturidade:** representam um caminho para o processo de melhoria, indicando quais áreas de processos devem ser implantadas para se alcançar cada nível, ilustrando assim a evolução da melhoria para toda a organização. Eles fornecem uma maneira de controlar ou estruturar o desempenho da organização dentro de uma dada disciplina ou um conjunto de disciplinas. São estágios evolutivos bem definidos em busca de um processo maduro. Cada nível estabelece uma parte importante do processo da organização.

Os níveis do modelo CMMI descrevem um caminho evolutivo recomendado para uma organização que deseja melhorar os processos utilizados para desenvolver e manter seus produtos e serviços. Também podem resultar de classificações obtidas por meio de avaliações realizadas em organizações, compreendendo a empresa toda (geralmente pequenas) ou grupos menores, tais como um grupo de projetos ou uma divisão de uma empresa.

- **Metas específicas:** aplicam-se a uma área de processo (PA) e descrevem o que deve ser implementado para satisfazer essa área. São usadas em avaliações para ajudar a determinar se uma área de processo está estabelecida.
- **Práticas específicas:** são atividades consideradas importantes no estabelecimento da meta específica associada. Descrevem as atividades esperadas para resultar no estabelecimento das metas específicas de uma área de processo.
- **Metas genéricas:** são assim denominadas porque a mesma meta aparece em múltiplas áreas de processo. A satisfação de uma meta genérica para a área de um processo significa maior controle no planejamento e na implantação dos processos a ela associados, indicando se esses processos serão eficazes, duradouros e se vão se repetir. Essas metas são usadas nas avaliações para determinar se uma área de processo é satisfeita.

- **Áreas de processo:** é um conjunto de práticas relatadas em uma área que, quando estabelecidas coletivamente, satisfaz a um conjunto de metas consideradas importantes para obter uma melhoria significativa nessa área. Elas descrevem aspectos de cada processo, mas não descrevem como um processo eficaz é executado, e sim como organizações, usando um processo eficaz, o fazem e por que o fazem. Isso quer dizer que as áreas de processos descrevem o que deve ser feito, mas não definem como elas estão agrupadas em níveis, de forma a atingir os objetivos de cada nível de maturidade, conforme apresentado na Figura 3.3.

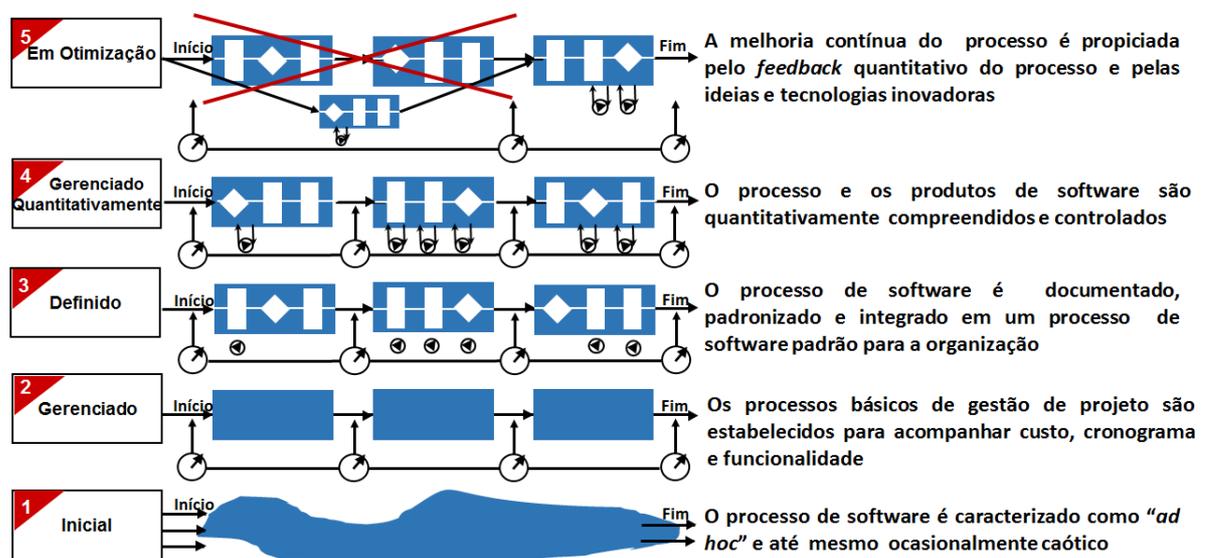


Figura 3.3 – Visibilidade do CMMI

Fonte: adaptada de COUTO (2007).

Além de estarem organizadas em níveis, as áreas de processos estão agrupadas em categorias (por exemplo, Gestão de Processo, Gestão de Projeto, Engenharia e Suporte), de forma a atingir níveis de capacidade, conforme representado na Figura 3.4.

	Nível 2	Nível 3	Nível 4	Nível 5
Engenharia	<ul style="list-style-type: none"> Gestão de Requisitos 	<ul style="list-style-type: none"> Desenvolvimento de Requisitos Solução Técnica Integração de Produto <ul style="list-style-type: none"> Verificação Validação 		
Projetos	<ul style="list-style-type: none"> Planejamento de Projeto Monitoramento e Controle de Projeto Gestão de Contrato com Fornecedores 	<ul style="list-style-type: none"> Gestão Integrada de Projeto Gestão de Riscos 	<ul style="list-style-type: none"> Gestão Quantitativa de Projeto 	
Processos		<ul style="list-style-type: none"> Foco nos Processos da Organização Definição dos Processos da Organização <ul style="list-style-type: none"> Treinamento da Organização 	<ul style="list-style-type: none"> Desempenho dos Processos da Organização 	<ul style="list-style-type: none"> Gerenciamento de Desempenho Organizacional
Suporte	<ul style="list-style-type: none"> Medição e Análise Garantia da Qualidade de Processo e Produto Gestão de Configuração 	<ul style="list-style-type: none"> Análise e Tomada de Decisões 		<ul style="list-style-type: none"> Análise e Resolução de Causas

Figura 3.4 – Áreas de processo do CMMI.

- Uma **subprática** (ver Figura 3.2) é uma descrição detalhada que fornece uma orientação para a interpretação e implementação de uma prática específica ou genérica.
- Os **produtos de trabalho típicos** (ver Figura 3.2) listam exemplos de resultados (*outputs*) de uma prática específica.

3.2.2 Representações do modelo CMMI

A definição de um CMMI permite que a comunidade desenvolva modelos que apoiem diferentes abordagens para a melhoria de processo. O CMMI possibilita abordar essa melhoria e a avaliação de processos utilizando duas representações diferentes: *contínua* e *por estágios*.

A *representação contínua* permite que a organização escolha determinada área de processo (ou grupo de áreas de processo) e melhore processos a ela relacionados. Essa representação utiliza níveis de capacidade para caracterizar a melhoria associada a certa área de processo.

A *representação por estágios* provém de uma sequência de melhorias, iniciada com as práticas de gerenciamento básicas e progredindo por meio de um caminho predefinido de níveis sucessivos, cada qual servindo de fundamento para o próximo. Dessa forma, permite

comparações por meio de organizações que utilizam os níveis de maturidade como base de comparação.

Um dos caminhos permite que as organizações melhorem, de forma incremental, os processos correspondentes a uma ou mais áreas de processo, individualmente selecionadas pela organização. O outro permite que elas melhorem um conjunto de processos inter-relacionados e, de forma incremental, tratem sucessivos conjuntos de áreas de processo.

A estrutura do CMMI é suficiente para suportar as duas representações (MIYASHIRO, 2007), conforme representado na Figura 3.5.

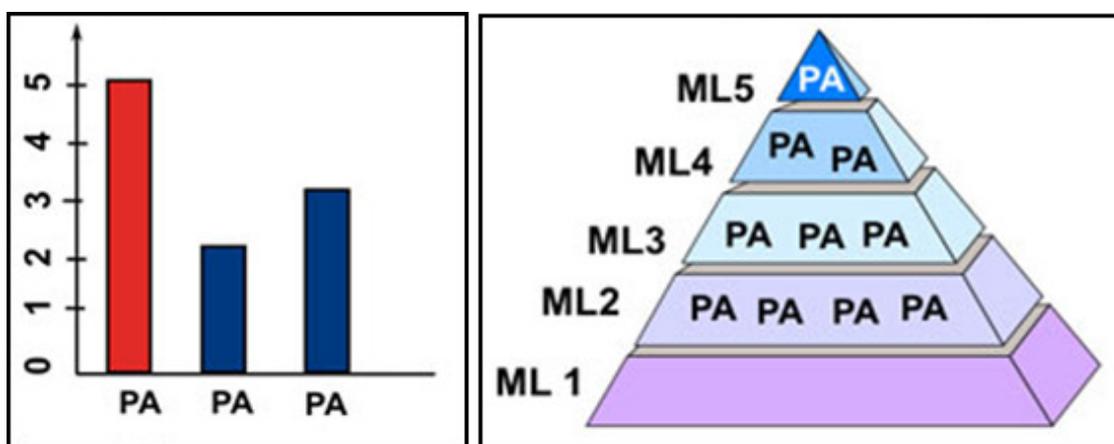


Figura 3.5 – Modelos de CMMI de representação contínua e por estágios

Fonte: MEDEIROS (2012).

- A representação contínua viabiliza a escolha da ordem das tarefas a serem realizadas que melhor atendam aos objetivos da organização, além de permitir verificações entre áreas de processos da organização e áreas básicas de processos. Essa representação do modelo CMMI-DEV adere aos modelos *Electronic Industries Alliance Interim Standard (EIA/IS) 731* e *International Organization for Standardization and International Electrotechnical Commission (ISO/IEC 15504)*, uma vez que sua organização segue o padrão da ISO/IEC 15504 (SEI, 2010).
- A representação contínua avalia o processo por áreas de processo, que definidas ou escolhidas de acordo com as necessidades da organização em conjunto com as

características básicas do CMMI, permite a identificação da capacidade de um processo para satisfazer suas necessidades. Isso indica que determinadas áreas de processo, que têm características básicas, devem ser cobertas desde o início do procedimento, porque são base e sustentação para a implementação das demais PAs, ou seja, a representação contínua do CMMI (Figura 3.6) é, de fato, uma estratégia de implantação, visto que apresenta uma solução ordenada para organizações que pretendem conquistar a classificação, porém sem pressa ou sem recursos para isso (SEI, 2010).

Uma organização, quando opta por utilizar uma representação contínua, deve saber que vai conquistar melhorias em níveis de capacidade que não terão classificação dentro do modelo CMMI, mas isso servirá de base para uma avaliação posterior, na qual será avaliada a maturidade do processo.

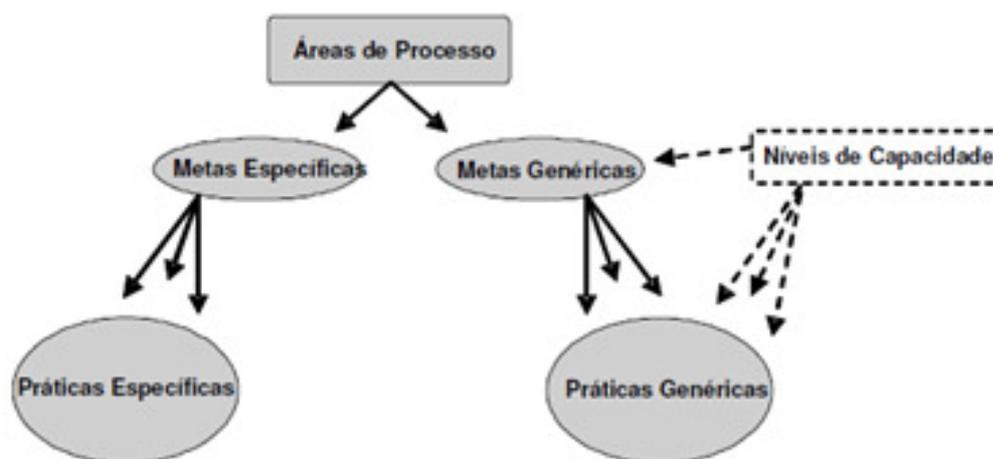


Figura 3.6 – Organização da representação contínua

Fonte: SEI (2010).

Para a conquista de níveis de capacidade por determinada área de processo, a representação contínua requer uma definição do nível de capacidade desejado para essa área, por isso, será necessária identificação se um processo for *executado* ou estiver *incompleto*. Portanto, o ponto de partida da representação contínua é chamado de *incompleto*.

Os níveis de capacidade são refletidos à medida que suas metas são satisfeitas e que os benefícios da melhoria de processo para determinada área de processo são colhidos.

Conforme já apresentado, os níveis de capacidade proporcionados pela representação contínua referem-se à melhoria de processo em áreas de processos individuais, de forma incremental, com seis níveis de capacidade:

- **Nível de capacidade 0: Incompleto**

Um processo é considerado incompleto quando não é executado ou quando é executado parcialmente, isto é, quando apenas algumas metas específicas da área de processo não são satisfeitas.

- **Nível de capacidade 1: Executado**

Um processo é considerado executado quando satisfaz metas e práticas específicas da área de processo. Esta representação não exige que as metas e práticas genéricas sejam cumpridas, porém, tal processo não é institucionalizado.

- **Nível de capacidade 2: Gerenciado**

Um processo é considerado gerenciado quando atende ao nível de capacidade 1 e realiza atividades de planejamento e execução de acordo com uma política organizacional. Neste nível, as práticas genéricas e específicas são realizadas e mantidas.

- **Nível de capacidade 3: Definido**

Um processo é considerado definido quando tem nível de capacidade 2 e segue a estrutura de diretrizes organizacionais, quando seus produtos de trabalho são medidos e mantidos como ativos de processo da organização. Apresenta padrões, processos e procedimentos que, geralmente, são descritos com mais rigor e adaptados de forma controlada por meio de diretrizes organizacionais.

Um processo de nível 3 estabelece claramente o objetivo, as entradas, os critérios de entrada, as atividades, os papéis, as medidas, as etapas de verificação, as saídas e os critérios de saída. Seu gerenciamento é feito segundo as atividades do processo.

- **Nível de capacidade 4: Gerenciado quantitativamente**

Um processo é considerado gerenciado quantitativamente quando tem nível de capacidade 3, é acompanhado e medido por meio de técnicas estatísticas e quantitativas. Seus objetivos de desempenho de processo são estabelecidos e utilizados como critérios na gestão de processo ao longo desse processo.

- **Nível de capacidade 5: Em otimização**

Um processo é considerado em otimização porque tem nível de capacidade 4 e é otimizado de acordo com as medições do processo, buscando a melhoria contínua do desempenho desse processo.

A representação por estágios distingue-se pela utilização do conjunto de áreas de processo associadas ao nível de maturidade a ser atingido para definir o caminho necessário para alcançar a melhoria desejada. Cada nível de maturidade é composto de um conjunto de áreas de processos predefinidas, e esses níveis auxiliam na previsão do resultado de futuros projetos, conforme representado na Figura 3.7.

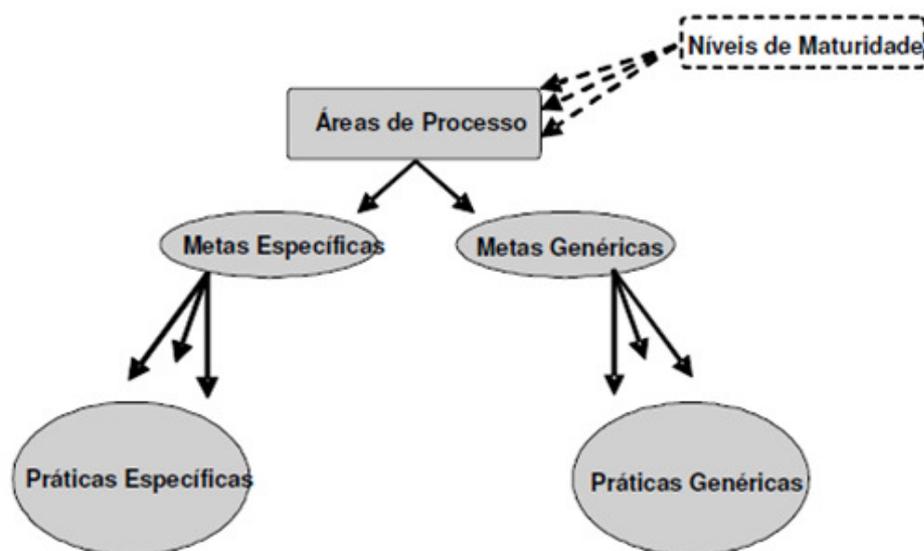


Figura 3.7 – Organização da representação por estágios

Fonte: SEI (2010).

O nível de maturidade é uma indicação do desempenho da organização em um conjunto de disciplinas, uma base evolutiva para alcançar o próximo nível de maturidade. Sua medição se dá pelo atendimento das metas pertinentes a cada conjunto. Os níveis de maturidade são seriados de 1 a 5: 1. Inicial; 2. Gerenciado; 3. Definido; 4. Gerenciado quantitativamente; 5. Em otimização.

- **Nível de maturidade 1: Inicial**

Um processo de maturidade nível geralmente é *ad hoc*⁴ e caótico, não fornece um ambiente estável para os processos. Nesse tipo de processo, o sucesso depende da experiência, da competência e do heroísmo das pessoas, e não do uso de processos formalizados.

- **Nível de maturidade 2: Gerenciado**

Um processo de maturidade nível 2 tem planejamento e segue uma política de execução. Seus projetos, em geral, contam com recursos de mão de obra e técnicos adequados para o alcance dos objetivos. São monitorados, controlados e revisados periodicamente, de acordo com o planejado. Além disso, são avaliados e têm sua aderência verificada em relação à descrição do processo pelas partes interessadas relevantes. Os projetos são executados, gerenciados e documentados de forma a assegurar que suas práticas sejam mantidas de acordo com seus planos.

- **Nível de maturidade 3: Definido**

Um processo de maturidade nível 3 deve ser bem definido e entendido. Segue padrões, procedimentos, ferramentas e métodos, e sua adaptação dá-se conforme o conjunto de processos padrão da organização, de acordo com as diretrizes para a adaptação. Os processos costumam ser descritos rigorosamente e estabelecem claramente o objetivo, as entradas, os critérios de

⁴ Na Engenharia de Software, a expressão *ad hoc* é utilizada para referenciar ciclos completos de construção de *software* que não foram projetados.

entrada, as atividades, os papéis, as medidas, as etapas de verificação, as saídas e os critérios de saída.

- **Nível de maturidade 4: Gerenciado quantitativamente**

Um processo de maturidade nível 4 estabelece os objetivos quantitativos para atingir a qualidade e o desempenho definido no processo para ser utilizado na gestão de processos. O desempenho dos processos é controlado por meio de técnicas estatísticas e técnicas quantitativas e deve ser quantitativamente previsível.

- **Nível de maturidade 5: Em otimização**

Um processo de maturidade nível 5 requer a melhora contínua de seus processos. Conta com entendimento quantitativo de sua execução e das causas comuns de variação inerentes ao processo. Busca o aperfeiçoamento constante do desempenho por meio de melhorias incrementais e inovadoras do processo e da tecnologia, a fim de melhorar seu desempenho.

3.2.3 Comparação entre as representações

No CMMI, tanto os níveis de capacidade como os níveis de maturidade proporcionam procedimentos para medir a evolução dos processos das organizações e promovem resultados equivalentes (SEI, 2010). Entretanto, a representação contínua foca a capacidade individual das áreas de processos em implementar requisitos, enquanto a representação por estágios evidencia a maturidade da organização medida pelos níveis de maturidade, em que cada nível necessita da implementação coletiva das áreas de processos.

Para a representação contínua, emprega-se a expressão *nível de capacidade*; para a representação por estágios, a expressão *nível de maturidade*.

O conceito de *nível* é utilizado nas duas representações, que caracterizam melhorias contínuas a partir de um estado em que processos não têm definição, um estado que utilize informações quantitativas para regulamentar e gerenciar as melhorias necessárias para atender os objetivos da organização.

Para alcançar o nível desejado, uma organização deve atender a todas as metas associadas à área de processo ou ao conjunto de áreas de processos daquele nível, independentemente de ser nível de capacidade ou nível de maturidade.

Tanto a representação contínua quanto a representação por estágios permitem a implementação de melhorias de processo e apresenta substancialmente a mesma filosofia, utilizando os mesmos componentes do modelo. A Tabela 3.1 apresenta as principais diferenças entre as representações.

Tabela 3.1 – Comparação entre as representações adaptado

Contínua	Por estágios
Permite livre escolha da sequência de melhorias, de forma a melhor satisfazer aos objetivos estratégicos e mitigar as áreas de risco da organização.	Permite que as organizações tenham um caminho de melhoria predefinido e testado.
Permite visibilidade crescente da capacidade alcançada em cada área de processo.	Foca em um conjunto de processos que fornece à organização uma capacidade específica caracterizada por cada nível de maturidade.
Permite que melhorias em diferentes processos sejam realizadas em diferentes níveis.	Resume os resultados de melhoria de processo em uma forma simples: um único número que representa o nível de maturidade.
Reflete uma abordagem mais recente que ainda não dispõe de dados para demonstrar seu retorno do investimento.	Baseia-se em uma história relativamente longa de utilização, com estudos de casos e dados que demonstram o retorno do investimento.
Áreas de processo organizadas por categoria.	Áreas de processo organizadas por nível.
Seis níveis de capacidade (0 a 5).	Cinco níveis de maturidade (1 a 5).
Melhoria é medida usando níveis de capacidade que refletem na implementação incremental de uma área de processo em particular.	Melhoria é medida usando níveis de maturidade que refletem a implementação simultânea de diferentes áreas de processo.
Apêndice adicional descreve a equivalência de estágios, permite tradução de um objetivo em um nível de maturidade.	Não possui perfil de equivalência para ir de um nível de maturidade para um objetivo.

Fonte: SEI (2010) e SPIN (2010).

Apesar de a representação contínua não oferecer condições para a conquista de uma classificação no modelo CMMI-DEV, pode ser utilizada de forma pertinente, como estratégia de implantação, para atingir os níveis desejados.

3.2.4 Práticas a serem realizadas em cada área de processo

Áreas de processo do nível 2

- **Gerenciamento de requisitos (REQM):** viabiliza o gerenciamento de requisitos dos produtos e componentes de produto do projeto e garante o alinhamento entre esses requisitos, os planos e produtos de trabalho do projeto. Suas práticas estão descritas na Tabela 3.2.

Tabela 3.2 – Gerenciamento de requisitos

Código	Nome da prática	Descrição
SP1.1	Entender requisitos.	Trabalhar com os provedores de requisitos para obter um melhor entendimento do significado dos requisitos.
SP1.2	Obter comprometimento com os requisitos.	Obter comprometimento dos participantes do projeto com requisitos.
SP1.3	Gerenciar mudanças nos requisitos.	Gerenciar mudanças nos requisitos à medida que evoluem durante o projeto.
SP1.4	Manter rastreabilidade bidirecional dos requisitos.	Manter a rastreabilidade bidirecional entre requisitos e produtos de trabalho.
SP1.5	Garantir alinhamento entre produtos de trabalho, planos de projeto e requisitos.	Garantir que os planos de projetos e produtos de trabalho se mantêm alinhados com os requisitos.

Fonte: SEI (2010).

- **Planejamento do projeto (PP):** estabelece e mantém planos para definir as atividades de projeto. Suas práticas estão descritas na Tabela 3.3.

Tabela 3.3 – Planejamento do projeto

Código	Nome da prática	Descrição
SP1.1	Estimar o escopo do projeto.	Estabelecer uma estrutura analítica de projeto Work Breakdown Structure (WBS), de alto nível, para estimar o escopo do projeto.
SP1.2	Estabelecer estimativas para atributos de produtos de trabalho e de tarefas.	Estabelecer e manter estimativas para produtos de trabalho e atributos de tarefas.
SP1.3	Definir fases do ciclo de vida do projeto.	Definir fases do ciclo de vida do projeto para fins de planejamento.
SP1.4	Estimativas de esforço e custo.	Estimar esforço do projeto e custo para produtos de trabalho e as tarefas com base no raciocínio utilizado na estimativa.
SP2.1	Estabelecer orçamento e cronograma.	Estabelecer e manter o orçamento e o cronograma do projeto.
SP2.2	Identificar riscos do projeto.	Identificar e analisar os riscos do projeto.
SP2.3	Plano de gerenciamento de dados.	Plano de gestão de dados do projeto.
SP2.4	Planejar recursos do projeto.	Plano de recursos para executar o projeto.
SP2.5	Planejar habilidades e conhecimentos necessários.	Planejar habilidades e conhecimento necessários para a execução do projeto.
SP2.6	Planejar o envolvimento das partes interessadas.	Planejar o envolvimento das partes interessadas identificadas.
SP2.7	Estabelecer o plano do projeto.	Estabelecer e manter o plano global do projeto.
SP3.1	Revisar planos que afetam o projeto.	Revisar todos os planos que afetam o projeto para entender os compromissos do projeto.
SP3.2	Conciliar carga de trabalho e recursos.	Conciliar o plano do projeto com os recursos estimados e disponíveis. Ajustar o plano do projeto para conciliar os recursos disponíveis e estimados.
SP3.3	Obter comprometimento com o plano.	Obter o comprometimento das partes interessadas relevantes responsáveis pela execução e pelo apoio à execução do plano.

Fonte: SEI (2010).

- **Monitoramento e controle do projeto (PMC):** fornece subsídios para proporcionar o acompanhamento do progresso do projeto, de forma que as ações corretivas possam ser implementadas quando o desempenho do projeto desviar do plano. Suas práticas estão descritas na Tabela 3.4.

Tabela 3.4 – Monitoramento e controle do projeto

Código	Nome da prática	Descrição
SP1.1	Monitorar os parâmetros de planejamento do projeto.	Monitorar valores reais dos parâmetros de planejamento de projeto em relação ao plano de projeto.
SP1.2	Monitorar compromissos.	Monitorar os compromissos com relação aos identificados no plano de projeto.
SP1.3	Monitorar riscos do projeto.	Monitorar os riscos em relação àqueles identificados no plano de projeto.
SP1.4	Monitorar a gestão de dados.	Monitorar a gestão de dados do projeto com relação ao plano de projeto.
SP1.5	Monitorar o envolvimento das partes interessadas.	Monitorar o envolvimento das partes interessadas em relação ao plano de projeto.
SP1.6	Conduzir revisões de progresso.	Revisar periodicamente o progresso, o desempenho e as questões críticas do projeto.
SP1.7	Conduzir revisões de macro.	Revisar as realizações do projeto e os resultados em macros do projeto selecionado.
SP2.1	Analisar as questões.	Identificar e analisar questões e determinar ações corretivas para revê-las.
SP2.2	Implementar ações corretivas.	Implementar ações corretivas para tratar as questões críticas identificadas.
SP2.3	Gerenciar ações corretivas.	Gerenciar ações corretivas até sua conclusão.

Fonte: SEI (2010).

- **Gerenciamento de acordos com fornecedores (SAM):** fornece subsídios para gerenciar a aquisição de produtos e serviços de fornecedores. Suas práticas estão descritas na Tabela 3.5.

Tabela 3.5 – Gerenciamento de acordos com fornecedores

Código	Nome da prática	Descrição
SP1.1	Determinar o tipo de aquisição.	Determinar o tipo de aquisição para cada produto ou componente de produto a ser adquirido.
SP1.2	Selecionar fornecedores.	Selecionar fornecedores com base na avaliação de suas capacidades para satisfazer os requisitos especificados e critérios estabelecidos.
SP1.3	Estabelecer contratos com fornecedores.	Estabelecer e manter acordos com fornecedores.
SP2.1	Executar contrato com fornecedor.	Executar atividades com o fornecedor conforme especificado no contrato.
SP2.2	Aceitar o produto adquirido.	Assegurar que o contrato com o fornecedor seja cumprido antes de aceitar o produto adquirido.
SP2.3	Garantir a transição de produtos.	Assegurar a transição de produtos adquiridos do fornecedor.

Fonte: SEI (2010).

- **Medições e análises (MA):** fornece subsídios para desenvolver e manter uma medição usada para apoiar as necessidades de informação de gestão. Suas práticas estão descritas na Tabela 3.6.

Tabela 3.6 – Medições e análises

Código	Nome da prática	Descrição
SP1.1	Estabelecer objetivos de medição.	Estabelecer e manter objetivos de medição derivados de necessidades de informação e objetivos identificados.
SP1.2	Especificar medidas.	Especificar medidas para satisfazer aos objetivos de medição.
SP1.3	Especificar procedimentos de coleta e armazenamento de dados.	Especificar como os dados resultantes de medição são obtidos e armazenados.
SP1.4	Especificar procedimento de análise.	Especificar como os dados resultantes de medição são analisados e comunicados.
SP2.1	Obter dados resultantes de medição.	Obter dados resultantes de medição especificados.
SP2.2	Analisar dados resultantes de medição.	Analisar e interpretar dados resultantes de medição.
SP2.3	Armazenar dados resultantes.	Gerenciar e armazenar dados resultantes de medição, especificações de medição e resultados de análise.
SP2.4	Comunicar resultados.	Comunicar resultados das atividades de medição e análise para todas as partes interessadas relevantes.

Fonte: SEI (2010).

- **Garantia da qualidade do processo e do produto (PPQA):** dá à equipe e à gerência visibilidade dos processos e produtos de trabalho associados. Suas práticas estão descritas na Tabela 3.7.

Tabela 3.7 – Garantia da qualidade do processo e do produto

Código	Nome da prática	Descrição
SP1.1	Avaliar objetivamente os processos.	Avaliar objetivamente os processos selecionados contra a descrição de processos, padrões e procedimentos.
SP1.2	Avaliar objetivamente produtos de trabalho.	Avaliar objetivamente os produtos de trabalho selecionados contra a descrição de processos, normas e procedimentos.
SP2.1	Comunicar e resolver problemas de descumprimento.	Comunicar as questões de qualidade e garantir a solução de problemas de não conformidade com a equipe e com os gerentes.
SP2.2	Estabelecer registros.	Estabelecer e manter registros das atividades de garantia da qualidade.

Fonte: SEI (2010).

- **Gerenciamento de configurações (CM):** fornece subsídios para estabelecer e manter a integridade dos produtos de trabalho, utilizando identificação de configuração, controle de configuração, balanço das atividades de configuração e auditorias de configuração. Suas práticas estão descritas na Tabela 3.8.

Tabela 3.8 – Gerenciamento de configurações

SP1.1	Identificar itens de configuração.	Identificar itens de configuração, componentes e produtos de trabalho relacionados a serem colocados sob gestão de configuração.
SP1.2	Estabelecer um sistema de gestão de configuração.	Estabelecer e manter um sistema de gestão de configuração e de gestão de mudanças para controlar os produtos de trabalho.
SP1.3	Criar ou liberar baselines.	Criar ou liberar baselines para uso interno e para entrega ao cliente.
SP2.1	Acompanhar solicitações de mudanças.	Acompanhar as solicitações de mudança para os itens de configuração.
SP2.2	Controlar itens de configurações.	Controlar mudanças para os itens de configuração.
SP3.1	Estabelecer registros de gestão de configuração.	Estabelecer e manter registros que descrevam os itens de configuração.
SP3.2	Realizar auditorias de configuração.	Executar auditorias de configuração para manter a integridade dos baselines de configuração.

Fonte: SEI (2010).

Áreas de processo do nível 3

- **Desenvolvimento de requisitos (RD):** fornece subsídios para obter, analisar e estabelecer cliente, produto, produtos e requisitos do componente de produto. Suas práticas estão descritas na Tabela 3.9.

Tabela 3.9 – Desenvolvimento de requisitos

Código	Nome da prática	Descrição
SP1.1	Levantar necessidades.	Levantar as necessidades das partes interessadas, suas expectativas, restrições e interfaces para todas as fases do ciclo de vida do produto.
SP1.2	Transformar as necessidades dos envolvidos em requisitos do cliente.	Transformar as necessidades dos envolvidos, expectativas, restrições e interfaces para os requisitos dos clientes priorizados.
SP2.1	Estabelecer requisitos de produto e de componente de produto.	Estabelecer e manter os requisitos de produto e de componente de produto com base nos requisitos do cliente.
SP2.2	Alocar requisitos de componente de produto.	Alocar os requisitos a cada componente de produto.
SP2.3	Identificar requisitos de interface.	Identificar os requisitos de interface.
SP3.1	Estabelecer conceitos operacionais e cenários.	Estabelecer e manter os conceitos operacionais e cenários associados.
SP3.2	Estabelecer uma definição da funcionalidade requerida e atributos de qualidade.	Estabelecer e manter uma definição da funcionalidade requerida e atributos de qualidade.
SP3.3	Analisar requisitos.	Analisar os requisitos para assegurar que são necessários e suficientes.
SP3.4	Analisar requisitos visando ao balanceamento.	Analisar os requisitos para balancear as necessidades e as restrições das partes interessadas.
SP3.5	Validar requisitos.	Validar os requisitos para assegurar que o produto resultante vai funcionar como pretendido no ambiente do usuário final.

Fonte: SEI (2010).

- **Solução técnica (TS):** fornece subsídios para selecionar, projetar e implementar soluções para os requisitos. Suas práticas estão descritas na Tabela 3.10.

Tabela 3.10 – Solução técnica

Código	Nome da prática	Descrição
SP1.1	Desenvolver soluções alternativas e critérios de seleção.	Desenvolver soluções alternativas e critérios de seleção.
SP1.2	Selecionar soluções de componentes de produto.	Selecionar soluções de componentes de produto baseado nos critérios de avaliação.
SP2.1	Desenvolver o design do produto ou dos componentes de produto.	Desenvolver um design para o produto ou componente de produto.
SP2.2	Estabelecer um pacote de dados técnicos.	Estabelecer e manter um pacote de dados técnicos.
SP2.3	Projetar interfaces utilizando critérios.	Projetar as interfaces dos componentes do produto a partir dos critérios estabelecidos e mantidos.
SP2.4	Analisar alternativas: desenvolver, comprar ou reusar.	Avaliar se os componentes do produto devem ser desenvolvidos, comprados ou reusados com base em critérios estabelecidos.
SP3.1	Implementar design.	Implementar os designs dos componentes de produto.
SP3.2	Elaborar a documentação de suporte ao produto.	Elaborar e manter a documentação para o usuário final.

Fonte: SEI (2010).

- **Integração de produto (PI):** fornece subsídios para montar o produto a partir de seus componentes, certificando-se de que o produto integrado comporta-se e funciona adequadamente, além de subsídios para a entrega desse produto. Suas práticas estão descritas na Tabela 3.11.

Tabela 3.11 – Integração de produto

Código	Nome da prática	Descrição
SP1.1	Estabelecer uma estratégia de integração.	Estabelecer e manter a estratégia de integração do produto.
SP1.2	Estabelecer ambiente de integração do produto.	Estabelecer e manter o ambiente necessário para dar suporte à integração dos componentes do produto.
SP1.3	Estabelecer procedimentos e critérios para integração do produto.	Estabelecer e manter os procedimentos e critérios para a integração dos componentes do produto.
SP2.1	Revisar descrições de interfaces para assegurar completude.	Revisar as descrições das interfaces, visando assegurar cobertura e completude.
SP2.2	Gerenciar interfaces.	Gerenciar as definições, designs e mudanças das interfaces internas e externas entre o produto e os componentes do produto.
SP3.1	Confirmar se os componentes do produto estão prontos para serem integrados.	Confirmar, antes da montagem, se cada componente de produto necessário foi identificado corretamente, se comporta de acordo com sua descrição e se as interfaces estão em conformidade com suas descrições.
SP3.2	Montar componentes do produto.	Montar os componentes do produto de acordo com a estratégia de integração e os procedimentos.
SP3.3	Avaliar componentes de produto montados.	Avaliar os componentes de produto montados quanto à compatibilidade de interface.
SP3.4	Empacotar e entregar produto ou componente de produto.	Empacotar o produto ou o componente de produto e entregá-lo ao cliente.

Fonte: SEI (2010).

- **Verificação (VER):** fornece subsídios para assegurar que os produtos de trabalho selecionados satisfaçam os requisitos especificados. Suas práticas estão descritas na Tabela 3.12.

Tabela 3.12 – Verificação

Código	Nome da prática	Descrição
SP1.1	Selecionar produtos de trabalho para verificação.	Selecionar produtos de trabalho a serem verificados e métodos de verificação a serem utilizados.
SP1.2	Estabelecer ambiente de verificação.	Estabelecer e manter o ambiente necessário para dar suporte à verificação.
SP1.3	Estabelecer procedimentos e critérios de verificação.	Estabelecer e manter procedimentos e critérios de verificação para os produtos de trabalho selecionados.
SP2.1	Preparar-se para revisão por pares.	Preparar-se para a revisão dos produtos de trabalho selecionados, feita pelos pares.
SP2.2	Conduzir revisão por pares.	Realizar análises comparativas dos produtos de trabalho selecionados e identificar dúvidas resultantes da revisão.
SP2.3	Analisar dados de revisão por pares.	Analisar dados sobre a preparação, condução e resultados de revisão feita pelos pares.
SP3.1	Realizar verificação.	Realizar a verificação nos produtos de trabalho selecionados.
SP3.2	Analisar resultados da verificação.	Analisar os resultados de todas as atividades de verificação.

Fonte: SEI (2010).

- **Validação (VAL):** fornece subsídios para demonstrar que um produto ou componente de produto satisfaz a seu uso pretendido quando colocado em seu ambiente pretendido. Suas práticas estão descritas na Tabela 3.13.

Tabela 3.13 – Validação

Código	Nome da prática	Descrição
SP1.1	Selecionar produtos para validação.	Selecionar os produtos e componentes de produto a serem validados e os métodos de validação a serem utilizados.
SP1.2	Estabelecer ambiente de validação.	Estabelecer e manter o ambiente necessário para a validação.
SP1.3	Estabelecer procedimentos e critérios de validação.	Estabelecer e manter os procedimentos e critérios de validação.
SP2.1	Realizar validação	Realizar a validação dos produtos e componentes de produto selecionados.
SP2.2	Analisar resultados de validação.	Analisar os resultados das atividades de validação.

Fonte: SEI (2010).

- **Foco no processo da organização (OPF):** fornece subsídios para planejar, implementar e implantar melhorias nos processos da organização com base na compreensão dos pontos fortes e fracos desses processos e dos ativos de processo da organização. Suas práticas estão descritas na Tabela 3.14.

Tabela 3.14 – Foco no processo da organização (OPF)

Código	Nome da prática	Descrição
SP1.1	Estabelecer necessidades de processo da organização.	Estabelecer e manter a descrição das necessidades e dos objetivos de processo da organização.
SP1.2	Avaliar os processos da organização.	Avaliar os processos da organização periodicamente, e conforme necessário, para conhecer seus pontos fortes e pontos fracos.
SP1.3	Identificar melhorias para os processos da organização.	Identificar melhorias para os processos e ativos de processo da organização.
SP2.1	Estabelecer planos de ação de processo.	Estabelecer e manter planos de ação de processo para promover melhorias nos processos e ativos de processo da organização.
SP2.2	Implementar planos de ação de processo.	Implementar planos de ação de processo.
SP3.1	Implantar ativos de processo da organização.	Implantar ativos de processos organizacionais em toda a organização.
SP3.2	Implantar processos padrão.	Implantar o conjunto de processos padrão nos projetos desde o startup e implementar mudanças nesses processos ao longo do ciclo de vida de cada projeto, conforme apropriado.
SP3.3	Monitorar a implementação.	Monitorar a implementação do conjunto de processos padrão da organização e o uso dos ativos de processo em todos os projetos.
SP3.4	Incorporar experiências em ativos de processos organizacionais.	Incorporar, nos ativos de processo da organização, os produtos de trabalho, as medidas e as informações para melhoria relacionados a processo que foram derivados do planejamento e da execução dos processos. Incorporar experiências relacionadas ao planejamento e à execução de processo em ativos de processos organizacionais.

Fonte: SEI (2010).

- **Definição do processo da organização (OPD):** fornece subsídios para estabelecer e manter um conjunto utilizável de ativos de processos organizacionais, padrões de ambiente de trabalho
- e as regras e diretrizes para as equipes. Suas práticas estão descritas na Tabela 3.15.

Tabela 3.15 – Definição do processo da organização

Código	Nome da prática	Descrição
SP1.1	Estabelecer processos padrão.	Estabelecer e manter o conjunto de processos padrão da organização.
SP1.2	Estabelecer descrições de modelos de ciclo de vida.	Estabelecer e manter as descrições dos modelos de ciclo de vida aprovado para uso na organização.
SP1.3	Estabelecer critérios e diretrizes para adaptação.	Estabelecer e manter critérios e diretrizes para adaptação do conjunto de processos padrão da organização.
SP1.4	Estabelecer o repositório de medições da organização.	Estabelecer e manter o repositório de medições da organização.
SP1.5	Estabelecer a biblioteca de ativos de processo da organização.	Estabelecer e manter a biblioteca de ativos de processo da organização.
SP1.6	Estabelecer padrões de ambiente de trabalho.	Estabelecer e manter padrões de ambiente de trabalho.
SP1.7	Estabelecer regras e orientações para equipes.	Estabelecer e manter regras de organização e diretrizes para a estrutura, formação e operação de equipes.

Fonte: SEI (2010).

- **Treinamento organizacional (OT):** fornece subsídios para desenvolver as habilidades e o conhecimento das pessoas para que elas possam desempenhar seus papéis de forma eficiente e eficaz. Suas práticas estão descritas na Tabela 3.16.

Tabela 3.16 – Treinamento organizacional

Código	Nome da prática	Descrição
SP1.1	Estabelecer necessidades estratégicas de treinamento.	Estabelecer e manter as necessidades estratégicas de treinamento da organização.
SP1.2	Identificar as necessidades de treinamento sob responsabilidade da organização.	Identificar quais necessidades de treinamento são de responsabilidade da organização e quais devem ser atribuídas a cada projeto ou grupo de suporte.
SP1.3	Estabelecer um plano tático de treinamento na organização.	Estabelecer e manter um plano tático de treinamento na organização.
SP1.4	Estabelecer a capacidade de treinamento.	Estabelecer e manter a capacidade de treinamento para tratar as necessidades de treinamento na organização.
SP2.1	Fornecer treinamento.	Fornecer treinamentos de acordo com o plano tático de treinamento na organização.
SP2.2	Estabelecer registros de treinamento.	Estabelecer e manter registros dos treinamentos organizacionais.
SP2.3	Avaliar a eficácia dos treinamentos.	Avaliar a eficácia do programa de treinamento da organização.

Fonte: SEI (2010).

- **Gerência integrada de projeto (IPM):** fornece subsídios para estabelecer e gerenciar o projeto e o envolvimento das partes interessadas relevantes, de acordo com um processo definido e integrado que é adaptado a partir do conjunto de processos padrão da organização. Suas práticas estão descritas na Tabela 3.17.

Tabela 3.17 – Gerência integrada de projeto

Código	Nome da prática	Descrição
SP1.1	Estabelecer o processo definido para o projeto.	Estabelecer e manter o processo definido para o projeto, desde o startup até o fim.
SP1.2	Utilizar os ativos de processo organizacional para planejar as atividades do projeto.	Utilizar os ativos de processo e o repositório de medições organizacionais para estimar e planejar as atividades do projeto.
SP1.3	Estabelecer o ambiente de trabalho do projeto.	Estabelecer e manter o ambiente de trabalho do projeto com base nos padrões de ambiente de trabalho da organização.
SP1.4	Integrar planos.	Integrar o plano do projeto com os outros planos que afetam o projeto de forma alinhada com o processo definido para o projeto.
SP1.5	Gerenciar o projeto utilizando planos integrados.	Gerenciar o projeto utilizando o plano de projeto, outros planos que afetam o projeto e o processo definido para ele.
SP1.6	Estabelecer equipes.	Estabelecer e manter equipes.
SP1.7	Contribuir para os ativos de processos organizacionais.	Contribuir com os ativos de processos organizacionais com experiências relatadas do processo.
SP2.1	Gerenciar o envolvimento das partes interessadas.	Gerenciar o envolvimento das partes interessadas relevantes no projeto.
SP2.2	Gerenciar dependências.	Participar, com as partes interessadas relevantes, da identificação, da negociação e do acompanhamento de dependências críticas.
SP2.3	Solucionar questões críticas de coordenação.	Solucionar questões críticas de coordenação com as partes interessadas relevantes.

Fonte: SEI (2010).

- **Gerência de risco (RSKM):** fornece subsídios para identificar potenciais problemas, de forma que atividades de tratamento de riscos possam ser planejadas e colocadas em prática, quando necessário, para mitigar impactos indesejáveis que comprometam a realização dos objetivos. Suas práticas estão descritas na Tabela 3.18.

Tabela 3.18 – Gerência de risco

Código	Nome da prática	Descrição
SP1.1	Determinar fontes e categorias de riscos.	Determinar as fontes e as categorias de riscos.
SP1.2	Definir parâmetros para riscos.	Definir parâmetros utilizados para analisar e categorizar os riscos e para controlar a atividade de gestão de riscos.
SP1.3	Estabelecer uma estratégia para gestão de riscos.	Estabelecer e manter a estratégia a ser utilizada para a gestão de riscos.
SP2.1	Identificar riscos.	Identificar e documentar riscos.
SP2.2	Avaliar, categorizar e priorizar riscos.	Avaliar e categorizar cada risco identificado utilizando categorias e os parâmetros definidos para riscos, e determinar suas prioridades relativas.
SP3.1	Elaborar planos de mitigação de riscos.	Desenvolver um plano de mitigação de riscos de acordo com a estratégia para gestão de riscos.
SP3.2	Implementar planos de mitigação de riscos.	Monitorar periodicamente o status de cada risco e implementar o plano de mitigação quando apropriado.

Fonte: SEI (2010).

- **Análise de decisão e resolução (DAR):** fornece subsídios para tomar decisões com base em um processo formal de avaliação de alternativas identificadas em relação a critérios estabelecidos. Suas práticas estão descritas na Tabela 3.19.

Tabela 3.19 – Análise de decisão e resolução

Código	Nome da prática	Descrição
SP1.1	Estabelecer diretrizes para análise e decisão.	Estabelecer e manter diretrizes para determinar quais questões críticas estão sujeitas a um processo formal para avaliação de alternativas.
SP1.2	Estabelecer critérios de avaliação.	Estabelecer e manter critério para avaliar as alternativas e para classificá-los de forma relativa.
SP1.3	Identificar soluções alternativas.	Identificar soluções alternativas para tratar questões críticas.
SP1.4	Selecionar métodos de avaliação.	Selecionar métodos de avaliação.
SP1.5	Avaliar soluções alternativas.	Avaliar soluções alternativas utilizando os critérios e os métodos estabelecidos.
SP1.6	Selecionar soluções.	Selecionar as soluções entre as alternativas, com base nos critérios de avaliação.

Fonte: SEI (2010).

Áreas de processo do nível 4

- **Desempenho do processo organizacional (OPP):** fornece subsídios para estabelecer e manter um entendimento quantitativo do desempenho dos processos selecionados da organização no apoio aos objetivos para alcançar qualidade e desempenho de processo, além de prover dados, *baselines* e modelos de desempenho de processo para gerenciar quantitativamente os projetos da organização. Suas práticas estão descritas na Tabela 3.20.

Tabela 3.20 – Desempenho do processo organizacional

Código	Nome da prática	Descrição
SP1.1	Estabelecer objetivos para qualidade e para desempenho de processo.	Estabelecer e manter objetivos quantitativos para qualidade e para desempenho de processo na organização, que são rastreáveis aos objetivos de negócio.
SP1.2	Selecionar processos.	Selecionar processos ou subprocessos pertencentes ao conjunto de processos padrão da organização a serem incluídos nas análises de desempenho de processo e manter a rastreabilidade para os negócios objetivos da organização.
SP1.3	Estabelecer medidas de desempenho de processo.	Estabelecer e manter definições de medidas a serem incluídas nas análises de desempenho de processo da organização.
SP1.4	Analisar a performance do processo e estabelecer baselines de desempenho de processo.	Analisar a performance dos processos selecionados e estabelecer e manter os baselines de desempenho de processo da organização.
SP1.5	Estabelecer modelos de desempenho de processo.	Estabelecer e manter modelos de desempenho de processo para o conjunto de processos padrão da organização.

Fonte: SEI (2010).

• **Gerência quantitativa de projeto (QPM):** fornece subsídios para gerenciar quantitativamente o projeto, no apoio aos objetivos para alcançar qualidade e para o desempenho de processo estabelecido para o projeto. Suas práticas estão descritas na Tabela 3.21.

Tabela 3.21 – Gerência quantitativa de projeto

Código	Nome da prática	Descrição
SP1.1	Estabelecer os objetivos do projeto.	Estabelecer e manter os objetivos para a qualidade e para o desempenho de processo.
SP1.2	Compor o processo definido.	Usando estatística e outras técnicas quantitativas, compor processo definido que permite que o projeto atinja seus objetivos de qualidade e desempenho do processo.
SP1.3	Selecionar subprocessos e atributos.	Selecionar subprocessos e atributos críticos, avaliar o desempenho, ajudando a atingir a qualidade do projeto e os objetivos do desempenho do processo.
SP1.4	Selecionar medidas e técnicas analíticas.	Selecionar medidas e as técnicas analíticas a serem utilizadas na gestão quantitativa.
SP2.1	Monitorar o desempenho dos subprocessos selecionados.	Monitorar o desempenho de subprocessos selecionados utilizando estatística e outras técnicas quantitativas.
SP2.2	Gerenciar o desempenho do projeto.	Gerenciar o projeto usando estatística e outras técnicas quantitativas para determinar se os objetivos para qualidade e para desempenho de processo no projeto serão satisfeitos ou não.
SP2.3	Realizar a análise de causa raiz.	Realizar a análise de causa raiz dos problemas selecionados para tratar deficiências na obtenção da qualidade do projeto e objetivos de desempenho de processo.

Fonte: SEI (2010).

Áreas de processo do nível 5

• **Análise causal e resolução (CAR):** fornece subsídios para identificar causas de resultados selecionados e tomar medidas para melhorar o desempenho do processo. Suas práticas estão descritas na Tabela 3.22.

Tabela 3.22 – Análise causal e resolução

Código	Nome da prática	Descrição
SP1.2	Analisar causas.	Realizar a análise de causas dos resultados selecionados e propor ações para tratá-las.
SP2.1	Implementar propostas de ação.	Implementar propostas de ação selecionadas desenvolvidas durante análise de causa.
SP2.2	Avaliar efeitos de ações implementadas.	Avaliar os efeitos das ações implementadas no desempenho do processo.
SP2.3	Registrar dados.	Registrar dados de análise e resolução de causas para uso no projeto e na organização.

Fonte: SEI (2010).

- **Gestão do processo organizacional (OPM):** fornece subsídios para gerenciar proativamente os objetivos de negócio da organização. Suas práticas estão descritas na Tabela 3.23.

Tabela 3.23 – Gestão do processo organizacional

Código	Nome da prática	Descrição
SP1.1	Manter objetivos de negócio.	Manter objetivos de negócio baseado na compreensão das estratégias de negócio e resultados de desempenho atual.
SP1.2	Analisar dados de desempenho do processo.	Analisar dados de desempenho do processo para determinar a capacidade da organização de alcançar os objetivos de negócio identificados.
SP1.3	Identificar áreas com potencial para melhorias.	Identificar áreas com potencial para melhorias que poderiam contribuir para o alcance de objetivos de negócio.
SP2.1	Identificar melhorias sugeridas.	Identificar e classificar melhorias sugeridas.
SP2.2	Analisar sugestões de melhorias.	Analisar melhorias sugeridas e seu possível impacto para a obtenção da qualidade e alcance de objetivos de desempenho do processo.
SP2.3	Validar melhorias.	Validar melhorias selecionadas.
SP2.4	Selecionar e aplicar melhorias para implantação.	Selecionar e aplicar melhorias para implantação em toda a organização, baseadas na avaliação dos custos, benefícios e outros fatores.
SP3.1	Planejar implantação.	Estabelecer e manter planos para a implantação das melhorias selecionadas.
SP3.2	Gerenciar implantação.	Gerenciar a implantação das melhorias selecionadas.
SP3.3	Avaliar os efeitos de melhorias.	Avaliar os efeitos das melhorias implantadas na qualidade e no desempenho do processo utilizando estatística e outras técnicas quantitativas.

Fonte: SEI (2010).

As áreas de processos são organizadas de formas diferentes na representação por estágios e na representação contínua, sendo em níveis, como representado na Figura 3.3, e em categorias, como representado na Figura 3.4. Essas categorias representam a relação entre as áreas de processo, ou seja, permite a organização das áreas de processo, grupos com um propósito específico.

Práticas genéricas a serem realizadas

A implementação das práticas genéricas implica que a forma como o processo é executado está consolidada, existindo padronização na execução do processo e comprometimento em relação a sua execução. Suas práticas estão descritas na Tabela 3.24.

Tabela 3.24 – Práticas genéricas

Código	Nome da prática	Descrição
GP 2.1	Estabelecer uma política organizacional.	Estabelecer e manter uma política organizacional para o planejamento e a execução do processo.
GP 2.2	Planejar o processo.	Estabelecer e manter o plano para a execução do processo.
GP 2.3	Fornecer recursos.	Fornecer os recursos adequados para a execução do processo, o desenvolvimento de produtos de trabalho e o fornecimento dos serviços do processo.
GP2.4	Designar responsabilidades.	Atribuir responsabilidade e autoridade para a execução do processo, para o desenvolvimento dos produtos de trabalho e o fornecimento dos serviços do processo.
GP2.5	Treinar pessoas.	Treinar pessoas para executar ou apoiar o processo conforme necessário.
GP2.6	Controlar produtos de trabalho.	Colocar produtos de trabalho selecionados do processo sob níveis de controle apropriados.
GP2.7	Identificar e envolver as partes interessadas relevantes.	Identificar e envolver as partes interessadas relevantes do processo, conforme planejado.
GP2.8	Monitorar e controlar o processo.	Monitorar e controlar o processo em relação ao estabelecido no plano para a execução do processo e implementar ações corretivas apropriadas.
GP2.9	Avaliar objetivamente a adesão.	Avaliar objetivamente a adesão do processo e selecionar produtos de trabalho em relação à descrição do processo, aos padrões e procedimentos e tratar não conformidades.
GP2.10	Revisar status com a gerência de nível superior.	Revisar as atividades, status e os resultados do processo com a gerência de nível superior e tratar questões críticas.

Fonte: SEI (2010).

3.3 MPS.BR

A melhoria de processos do *software* brasileiro (MPS.BR) é um movimento para a melhoria e um modelo de qualidade de processo voltado para a realidade do mercado de pequenas e médias empresas de desenvolvimento de *software* no Brasil. Ele é baseado no CMMI, nas normas ISO/IEC 12207 e ISO/IEC 15504, e na realidade do mercado brasileiro.

O MPS.Br é dividido em três partes: modelo de referência para melhoria do processo de *software* (MR-MPS), método de avaliação para melhoria do processo de *software* (MA-MPS) e modelo de negócio para melhoria do processo de *software* (MN-MPS), conforme apresentado na Figura 3.8.

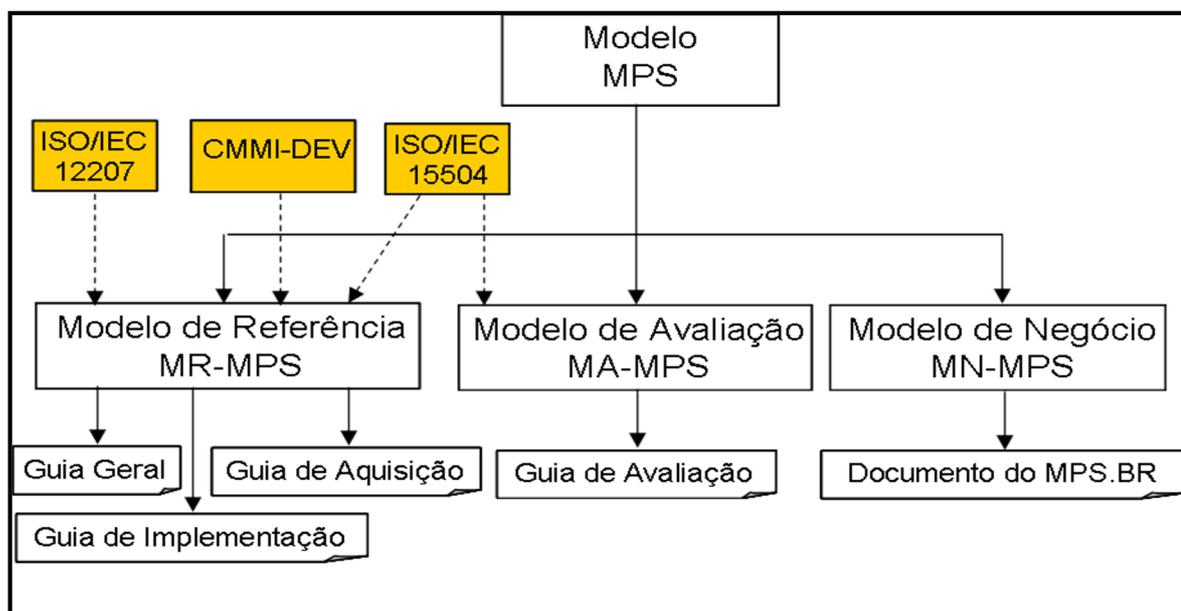


Figura 3.8 – Componentes do MPS.BR

Fonte: SOFTEX (2010).

Como um diferencial em relação aos demais padrões de processo, o MPS.BR apresenta sete níveis de maturidade: do maior para o menor nível, em otimização; gerenciado quantitativamente; definido; largamente definido; parcialmente definido; gerenciado; parcialmente gerenciado.

Cada nível de maturidade tem seus processos, nos quais são classificados os processos fundamentais (aquisição, gerência de requisitos, desenvolvimento de requisitos, solução técnica, integração do produto, instalação do produto e liberação do produto), os processos organizacionais (gerência de projeto, adaptação do processo para gerência de projeto, análise de decisão e resolução, gerência de riscos, avaliação e melhoria do processo organizacional, definição do processo organizacional, desempenho do processo organizacional, gerência quantitativa do projeto, análise e resolução de causas, inovação e implantação na organização) e os processos de apoio (garantia de qualidade, gerência de configuração, validação, medição, verificação e treinamento). Em seguida, é apresentada a capacidade com que são obtidos os resultados dos processos analisados. Cada nível de maturidade tem um número de capacidades definido, como apresenta a Figura 3.9.

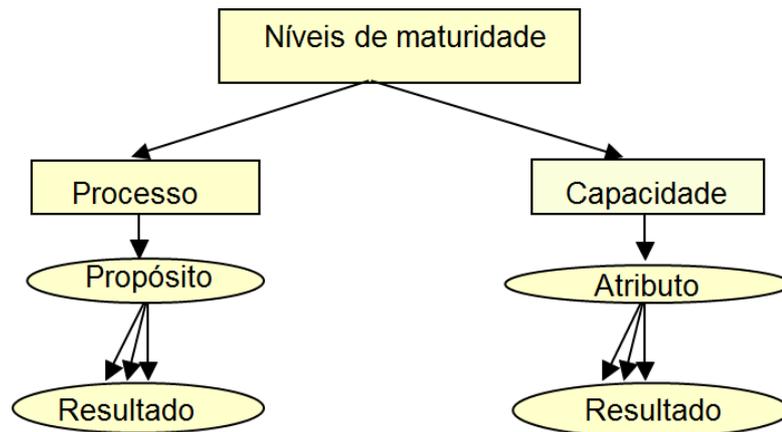


Figura 3.9 – Modelo de processo brasileiro

Fonte: SOFTEX (2010).

No Brasil, uma das vantagens indicadas para a utilização do modelo é seu custo reduzido para avaliação em relação às normas estrangeiras, sendo ideal para micros, pequenas e médias empresas.

Para reduzir os custos e estimular o aumento da qualidade dos produtos de *software*, o Ministério da Ciência, Tecnologia e Inovação (MCTI) promove, por meio do Programa Associação para Promoção da Excelência do *Software* Brasileiro (Softex), auxílio nas áreas operacional, de financiamento e de capacitação por meio da articulação de parceiros da iniciativa privada, do Governo e da academia, com o objetivo de desenvolver a inovação e a competitividade do setor brasileiro de *software* e dos serviços de tecnologia da informação.

O MPSBr tem como objetivo estender o modelo para a América Latina, incluindo o Chile, a Argentina, a Costa Rica, o Peru e o Uruguai.

Aderência entre os modelos CMMI e MPSBr

Apesar de o modelo CMMI estar estruturado em cinco níveis de maturidade e 22 áreas de processo e de o modelo MPSBr estar organizado em sete níveis de maturidade e 19 processos, existe uma perfeita aderência entre esses modelos (Tabela 3.25). Tal aderência se dá a partir das normas utilizadas como referência por ambos: a ISO/IEC 12207 e a ISO/IEC 151504.

Tabela 3.25 – Similaridade CMMI × MPSBr

CMMI		MPSBR		
Nível	Áreas de processo	Processo	Nível	
2	Gestão de requisitos	Gerência de requisitos	G	
	Planejamento de projeto	Gerência do projeto		
	Monitoramento e controle de projeto	Gerência de portfólio de projetos	F	
	Gestão de contrato com fornecedores	Aquisição		
	Medição e análise	Medição		
	Garantia da qualidade de processo e produto	Garantia da qualidade		
3	Gestão de configuração	Gerência de configuração	E	
	Solução técnica*	Gerência de reutilização		
	Foco nos processos da organização	Avaliação e melhoria do processo organizacional	E	
	Definição dos processos da organização +IPPD	Definição do processo organizacional		
	Treinamento na organização	Gerência de recursos humanos		
	Gestão integrada de projeto +IPPD	Gerência do projeto	D	
	Desenvolvimento de requisitos	Desenvolvimento de requisitos		
	Solução técnica*	Projeto e construção do produto		
	Integração de produto	Integração do produto		
	4	Verificação	Verificação	D
		Validação	Validação	
		Solução técnica*	Desenvolvimento para reutilização	C
Gestão de riscos		Gerência de riscos		
Análise e tomada de decisões		Gerência de decisões	B	
Desempenho do processo organizacional		Gerência do projeto (evolução)**		
5	Gestão quantitativa de projeto	Gerência do projeto (evolução)**	B	
	Análise e resolução de causas	Sem processos definidos		
5	Gestão do processo organizacional	Sem processos definidos	A	

Graças ao alto nível de aderência entre os modelos CMMI e o MPSBr, não existe necessidade de as organizações optarem por utilizar um ou outro modelo. É possível a um mesmo processo conquistar a classificação nos dois modelos.

É comum as organizações utilizarem o modelo MPSBr como estratégia de implantação para o CMMI, uma vez que este apresenta maior subdivisão em seus níveis, permitindo a conquista mais rápida de uma classificação. O primeiro nível do MPSBr exige a implementação de dois processos, enquanto para conquistar o primeiro nível do CMMI é necessário implementar sete áreas de processo. Outro motivo para trilhar esse caminho é poder utilizar os incentivos governamentais para o MPSBr. Os incentivos e financiamentos oferecidos pelo MCTI viabilizam a conquista de níveis de maturidade do modelo MPSBr; com ajustes e adaptações, pode-se também conquistar os níveis de maturidade do modelo CMMI.

A aderência entre os modelos pode ser observada com maior propriedade por meio do estudo da aderência entre as áreas de processos do modelo CMMI e os processos e resultados esperados

do modelo MPSBr, como apresentado na monografia “Uma abordagem para identificação da maturidade do processo de desenvolvimento de *software* utilizando o modelo MPS.BR a partir da sua aderência e mapeamento às práticas do CMMI V1.3” (QUEIROZ et al., 2011).

3.4 Modelos de maturidade e os sistemas embarcados

Muitos modelos e padrões foram criados para garantir a qualidade do produto de *software* e a qualidade do processo de desenvolvimento de *software*. Em sua maioria, esses padrões direcionam as organizações que os utilizam na realização de boas práticas para o desenvolvimento de sistemas, mais especificamente de *softwares*.

Padrões internacionais foram desenvolvidos e estão sendo utilizados com esse objetivo, mas uma de suas principais desvantagens é que eles fornecem diretrizes muito genéricas, tornando, assim, os modelos de qualidade difíceis de serem aplicados em atividades específicas, como o desenvolvimento de *softwares* embarcados.

Segundo Carvalho (2010) pesquisas atuais apontam para a necessidade de avaliação da qualidade dos componentes de *software*. Acredita-se que o uso de componentes qualificados e com critérios bem definidos durante seu desenvolvimento poderá reduzir riscos de falha, aumentando a probabilidade do sucesso da execução (CARVALHO, 2010). Muitas organizações se esforçam para identificar um componente apropriado em seu sistema. Acredita-se que uma estrutura bem definida e consistente para o desenvolvimento de componentes de *software* é essencial para a reusabilidade entre as organizações ou a utilização de componentes do mercado.

Conforme já foi dito, é crítica a falta de processos e métodos de desenvolvimento de *softwares* com diretrizes específicas para sistemas embarcados, o que inviabiliza a prevenção de seus defeitos.

Os projetos desses *softwares* estão ficando cada vez mais complexos e, com o aumento de seu uso, têm influenciado o aumento da quantidade de defeitos. Quando os defeitos ocorrem em sistemas críticos, sua fragilidade é colocada em evidência. Isso pode ser exemplificado pelo

número de acidentes aéreos em que não se consegue comprovação de falha humana como causa (CARVALHO, 2010).

Para atender às necessidades de sistema embarcado, a engenharia de *software* necessita proporcionar recursos que permitam aos processos atender às particularidades desses sistemas, o que significa, além de fornecer mecanismos de auxílio à produtividade, desenvolver mecanismos que viabilizem o atendimento e a medição dos requisitos do produto e permitam caracterizar (modelar) as necessidades de *hardware* e *software*, preservando sua unicidade.

Uma característica importante a ser considerada em um processo é o fato de que os componentes de sistema embarcados, em geral, fazem parte de um sistema maior, com estabilidade de requisitos. Isso quer dizer que, por sua especificidade, não ocorrem mudanças de requisitos, como nos sistemas convencionais e comerciais, circunstância que deve ser levada em consideração para determinar o paradigma de desenvolvimento.

Como já mencionamos, 80% dos motivos de falha em sistemas embarcados estão relacionados a problemas no *software* (ALONSO, 1998), o que demonstra a necessidade de mais estudos, pois, na literatura, as referências aos sistemas embarcados abordam o componente completo – poucas particionam o desenvolvimento ou são direcionadas para as partes de *software* do componente.

É crescente o esforço das organizações em identificar os componentes mais apropriados a serem inseridos em seu sistema e estudos vêm sendo realizados com esse objetivo. Uma estrutura bem definida e consistente para o desenvolvimento do *software* embarcado de forma a viabilizar a individualização e o compartilhamento da modelagem e da construção das partes *software* e *hardware* do componente poderá contribuir para a produção de componentes que atenda a tais necessidades.

Para atender às características dos sistemas embarcados, os desenvolvedores muitas vezes deixam de observar determinadas atividades da engenharia de *software*, uma vez que elas não levam em consideração as necessidades específicas do desenvolvimento e do produto final de um componente de sistema embarcado.

O desenvolvimento de *software* para componentes de sistemas embarcados exige que a engenharia de *software* inclua em seus controles mecanismos que permitam aperfeiçoar o produto final, além de seu processo de desenvolvimento. Assim, deve observar em seus procedimentos mecanismos para: a) melhor distribuir as funcionalidades entre o componente *hardware* e o componente *software*, em função de suas restrições; b) identificar erros em projetos antes de sua implementação; c) viabilizar o reúso de componentes; d) utilizar métricas de produto, além das métricas do projeto; e) utilizar modelos comportamentais; f) aumentar a produtividade do sistema embarcado, entre outras práticas que podem influenciar diretamente a qualidade dos sistemas embarcados (WAGNER, 2013).

Por essas razões, faz-se necessário o emprego de uma abordagem de desenvolvimento de *software* que leve em consideração as características e peculiaridades desse contexto.

3.5 Técnicas, ferramentas e padrões que auxiliam a qualidade do processo

Os padrões e as técnicas internacionais para desenvolvimento de *software* são amplos, o que, em geral, dificulta sua aplicação em atividades específicas, como o desenvolvimento de *softwares* embarcados. Entretanto, independentemente da metodologia ou técnica adotada, deve realizar as ações principais e comuns a qualquer tipo de projeto, especificação, modelagem, validação e síntese (EDWARDS; LEE; SANGIOVANNI-VINCENTELLI, 1997).

3.5.1 Técnicas e ferramentas utilizadas na área espacial (exemplos)

- ***Failure Mode and Effects Analysis (FMEA)***: método utilizado para antecipar a detecção de possíveis falhas do produto no processo de desenvolvimento.
- ***Design for Manufacturing and Assembly (DFMA)***: relacionamento do projeto de um produto com os processos de manufatura, de modo a otimizar suas operações de produção com a utilização de *softwares*.
- ***Design for Cost (DFC)***: controle de custos do projeto final do produto durante seu ciclo de vida.

- **Quality Function Deployment (QFD):** ferramentas que convertem em requisitos funcionais as solicitações do cliente, incluindo o método de comparação entre os produtos desenvolvidos e a concorrência.
- **Manufacturing Support Systems (MSS):** sistemas de apoio à manufatura, por exemplo: *Computer-Aided Design (CAD)*, *Computer-Aided Manufacturing (CAM)*, *Manufacturing Resource Planning (MRP)*, *Manufacturing Accounting Control (MAC)*, *Computer-Aided Engineering (CAE)* e *Computer-Aided Process Planning (CAPP)*.

3.5.2 Modelagem funcional

A modelagem funcional representa a estrutura dos modelos em diagramas funcionais, que podem ser utilizados nas atividades para atender às práticas exigidas, a saber:

- **Diagramas de fluxo de dados:** mostram um conjunto de funções (processos) e dados relacionados entre si, em que cada um dos processos pode ser decomposto em um nível inferior, mantendo entre os níveis consistências e balanceamento nos dados e funções.
- **Diagramas de transição de estado (DST):** são utilizados para definir o comportamento dos estados e os detalhes funcionais do comportamento dos componentes ou de um sistema. Definem a sequência de execução, reagindo ao conjunto de estímulos ambientais que interferem no funcionamento interno do sistema.
- **IDEF0:** é um método concebido para modelar as decisões, as ações e as atividades de um sistema ou componente. Organizado em uma visão funcional do sistema, auxilia na identificação das funções a serem executadas e dos fluxos de dados entre eles.

3.5.3 Modelagem orientada a objetos

A modelagem orientada a objetos é uma organização do sistema (*software*) que reflete a estrutura e o comportamento dos dados, além de ser uma alternativa para a abstração dos requisitos e modelagem, podendo representar um modelo de sistema embarcado.

Alguns diagramas da *Unified Modeling Language (UML)* são:

- **Diagrama de casos de uso (UCD):** usado para descrever a integração entre o sistema e seu ambiente. Pode representar o nível mais alto do modelo orientado a objetos.
- **Diagrama de pacotes (PD):** a partir de determinada classificação, apresenta o agrupamento dos elementos no modelo e pode ser usado para mostrar grupos de classes, grupos de componentes, grupos de processos, grupos de processadores em diagramas de implantação, entre outros.
- **Diagrama de sequência (SD):** representa a visão, em tempo sequenciado, da interação entre os atores e os objetos durante a realização de um cenário (instância), representando a troca de mensagens necessária entre seus objetos.
- **Diagrama de classes (CD):** é usado para identificar e caracterizar as classes de acordo com as suas características e seu comportamento, permitindo a definição das relações entre os objetos.
- **Diagrama de estado (SCD):** representa o comportamento de um objeto de uma classe em diferentes circunstâncias, utilizando uma máquina de estado finito que descreve como essa classe responde a diferentes estímulos.

4 ÁREA ESPACIAL

Como vimos no Capítulo 3, a utilização de um processo para o desenvolvimento de sistemas embarcados que atendam aos requisitos das áreas de processos do modelo CMMI e que estejam de acordo com os processos organizacionais da empresa poderá contribuir para institucionalizar as atividades básicas da organização, por exemplo, planejamento e gerenciamento de projeto, que são essenciais em todas as fases do processo.

Ao estudar os modelos de qualidade, vimos também que, em geral, o problema não está no *software*, mas na forma como ele é feito. Por isso, é preciso aplicar na indústria de *software* os conceitos de qualidade, respeitando as necessidades e especificidades de sua área de aplicação, como é o caso da área espacial, que vamos apresentar neste capítulo.

4.1 Exploração espacial

A Astronáutica é o estudo científico do espaço e de seus aspectos astronômicos, utilizando-se de recursos tecnológicos em naves espaciais, satélites artificiais ou sondas espaciais e, em algumas atividades, de seres humanos para desenvolver suas missões.

A criação dos foguetes foi precursora da exploração espacial, permitindo colocar em órbita satélites artificiais e astronautas para o estudo da Terra e do espaço.

Os satélites e as sondas são as partes facilmente visíveis em um conjunto de sistemas desenvolvidos para explorar o espaço, em geral com uma finalidade específica, tendo como parâmetros e requisitos exclusivamente sua função e o ambiente onde deverão operar, o que influencia diretamente em seu aspecto. São conceitualmente divididos em duas partes: *carga útil (payload)* e *plataforma (bus)*, sendo que a carga útil envolve os equipamentos necessários para o cumprimento da missão e a plataforma é composta pelos elementos necessários para suportar seu lançamento e sustentar a operação da carga útil:

- Telescópios espaciais Hubble Space Telescope INTEGRAL e XMM-Newton.

- Satélites de sensoriamento remoto CBERS.
- Satélites de telecomunicações A2100.
- Sondas interplanetárias SOHO, Rosetta, Mars Express, Voyager 1 e 2.

O Brasil conta com um discreto, porém qualificado, programa espacial coordenado pela Agência Espacial Brasileira (AEB), onde o Comando-Geral de Tecnologia Aeroespacial (CTA) é responsável pelos veículos lançadores, e o Instituto Nacional de Pesquisas Espaciais (INPE) pelas plataformas orbitais e cargas úteis.

4.1.1 Missão espacial

A missão espacial é a tarefa para a qual um sistema espacial é designado, na qual todos os esforços tecnológicos e humanos concentram-se em estruturar todos os sistemas para viabilizar sua realização.

Veja a seguir alguns exemplos de missão, segundo Souza (2008):

- **Observação da Terra:** sensoriamento remoto, observação da superfície da Terra para o estudo de elementos da geografia (natural e artificial) e da vegetação.
- **Meteorologia:** observação dos elementos relativos ao clima, tais como nuvens, ventos e temperaturas.
- **Oceanografia:** observação de elementos que caracterizam a dinâmica dos oceanos, como correntes, temperaturas e elevação da superfície.
- **Alarme e reconhecimento:** descoberta de incêndios e acidentes nucleares.
- **Observação astronômica:** posicionamento de telescópios em diversas órbitas para operar em frequências variadas.
- **Navegação por satélite** (satélites do tipo GPS).

- **Sondagem (interplanetárias ou não):** estudo da Lua, do Sol, dos diversos planetas do Sistema Solar e suas luas, asteroides e cometas.
- **Recuperáveis:** execução de experimentos de curta duração, por meio de satélites que permanecem poucas semanas em órbita e retornam à superfície terrestre.
- **Militares:** têm o propósito de contribuir, por meio de satélites, com sistemas de defesa para a execução de missões de interceptação e destruição.
- **Tripuladas:** Estação Espacial Internacional (International Space Station – ISS)

4.1.2 Sistema espacial

O sistema espacial é um conjunto de componentes que interagem de forma organizada com características diversas, mas um objetivo comum, tais como pessoas, organizações, procedimentos, *softwares*, equipamentos e instalações, como representado na Figura 4.1

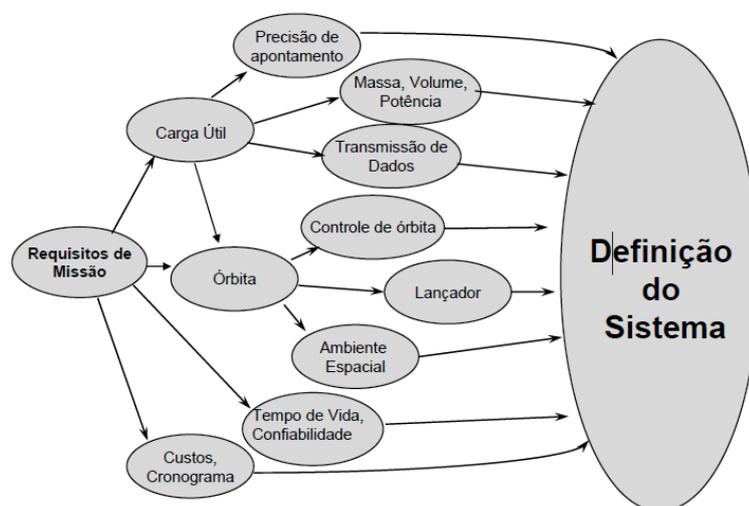


Figura 4.1 – Processo de definição de um sistema espacial

Fonte: LOUREIRO (2010).

Um programa espacial é um conjunto de atividades que combinam meios materiais, humanos e financeiros que permitem atingir objetivos de caráter técnico, científico ou de aplicação, por meio de técnicas espaciais.

Um projeto espacial é um conjunto de atividades de estudo e realização de um sistema espacial geralmente desenvolvido em fases (0, A, B, C, D, E, F), com cronograma, orçamento e alocação de recursos humanos bem definidos, conforme representado na Figura 4.2.

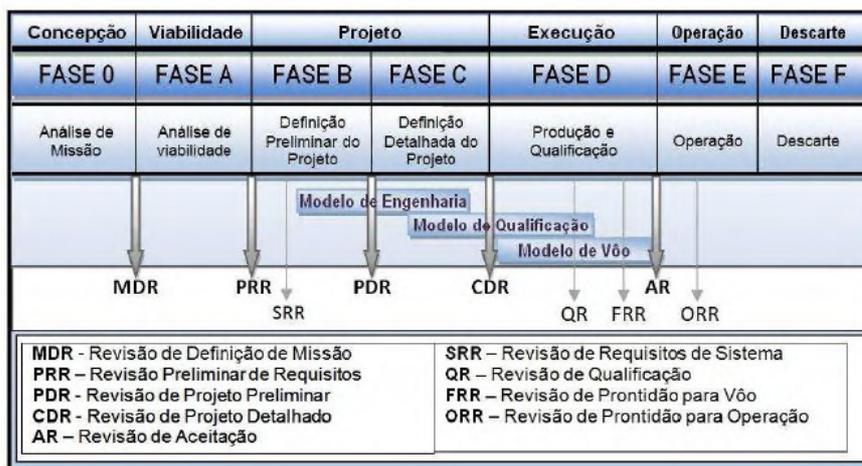


Figura 4.2 – Elementos do ciclo de vida de um projeto na área espacial, padrão ECSS-M-ST-10C

Fonte: YASSUDA (2010).

- **Fase 0 – Análise de missão:** identificação dos principais requisitos de missão, como desempenho esperado, confiabilidade e vida útil do produto, restrições e avaliação de custo contra orçamento do projeto.
- **Fase A – Análise de viabilidade:** estudo de viabilidade conceitual e tecnológica por meio da identificação das necessidades, riscos e restrições relacionados à implementação do projeto, como custos, cronogramas, organização, operação, manutenção, produção e descarte.
- **Fase B – Definição do projeto preliminar:** consolidação da proposta conceitual para as configurações de sistema e suas operações com base nas soluções técnicas escolhidas.
- **Fase C – Definição detalhada do projeto:** detalhamento conceitual do sistema e suas operações, análise de risco e detalhamento dos planos de fabricação, integração e testes para o sistema e seus componentes.

- **Fase D – Produção e qualificação:** são desenvolvidas as atividades de fabricação, integração e testes do modelo de qualificação, produzindo a versão final da documentação, configuração final dos processos e do ferramental de fabricação do modelo.
- **Fase E – Operação:** atividades de preparação para o lançamento, incluindo atividades para injeção em órbita e procedimentos iniciais para a operação do sistema, tanto as atividades relativas às operações de órbita quanto as do segmento solo.
- **Fase F – Descarte:** implementação dos planos para o descarte do sistema.

4.1.3 Distribuição das responsabilidades entre os segmentos

Toda missão espacial que utiliza satélites divide suas responsabilidades em partes chamadas de *segmento espacial*, que é a parte colocada em órbita (geralmente o satélite); o *segmento lançador*, que é a parte utilizada para a colocação do satélite em órbita-foguete; e o *segmento solo*, constituído pelas estações de rastreo e controle do satélite, sendo a parte encarregada da supervisão e do funcionamento do satélite, de seu controle e recepção dos dados de suas cargas úteis, conforme apresentado na Figura 4.3.

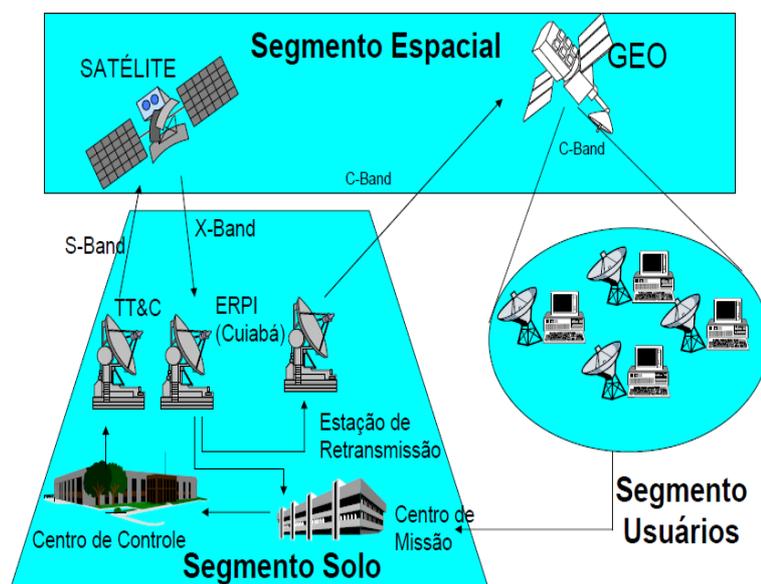


Figura 4.3 – Elementos que constituem os segmentos de um sistema espacial

Fonte: SOUZA (2008).

Em uma missão espacial, geralmente a responsabilidade pela realização das tarefas operacionais é dividida entre o segmento lançador, o segmento solo e o segmento espacial. As responsabilidades do segmento lançador têm menor duração, mas ele é o mais importante, pois é nesse momento que há maior chance de os problemas acontecerem. Um veículo lançador de satélites utiliza motores-foguetes carregados com propelente em todos os estágios, com capacidade para colocar satélites em órbita.

O segmento espacial tem a responsabilidade de executar procedimentos de rotina, como monitoramento de parâmetros, geração de alertas, envio de dados para solo, sincronizações, alguns controles de cargas úteis, entre outras atividades. A execução das atividades desenvolvidas no espaço é importante, pois promove a independência do segmento e minimiza os custos com a comunicação. O segmento solo realiza a comunicação com o satélite (telemetria e telecomandos) durante os períodos da órbita, elabora e executa os planos de voo baseados na missão aos usuários, além de tarefas que demandam algum tipo de planejamento de utilização de carga útil, análise de resultados e tomada de decisões para que se atinjam os objetivos da missão, conforme apresentado na Figura 4.4

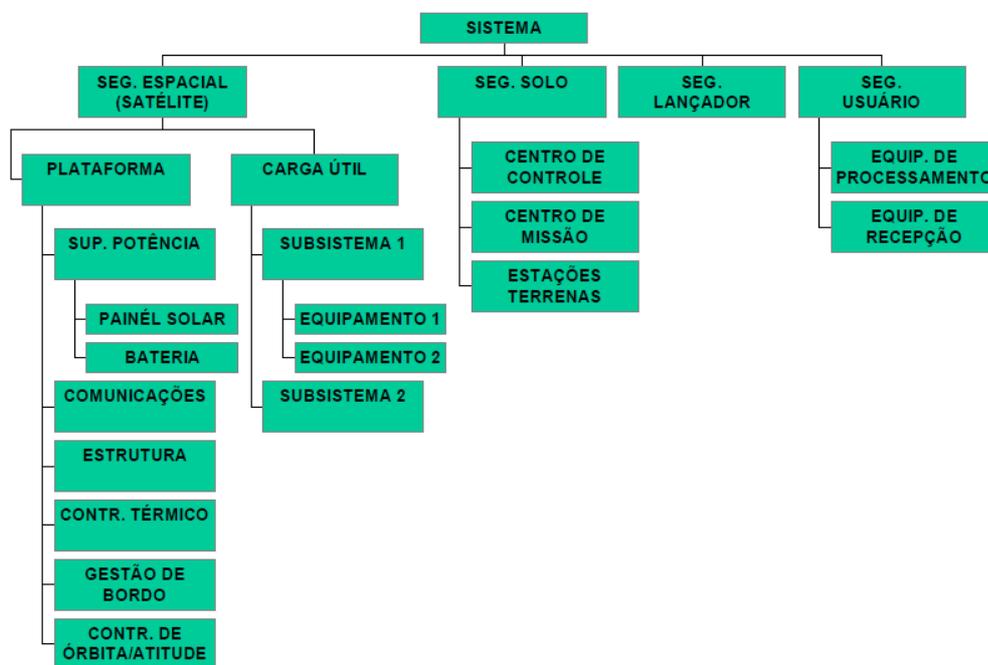


Figura 4.4 – Elementos dos segmentos de um sistema espacial

Fonte: SOUZA (2008).

4.2 Requisitos para o desenvolvimento de sistemas embarcados na área espacial segundo Modelo European Cooperation for Space Standardization (ECSS)

Um projeto espacial, em geral, é composto de componentes de alta complexidade que necessitam de tecnologia avançada para serem produzidos e, muitas vezes, geram alto custo. Em um projeto com essas características, um dos requisitos mais importantes é a confiabilidade dos componentes, pois, a partir do momento em que a fase de operação de um produto espacial se inicia, em geral a realização de manutenção desses componentes torna-se impossível.

Outra característica desse tipo de projeto refere-se às atividades de testes e aprovação de tais componentes, uma vez que é quase impossível esses testes acontecerem em ambiente real de operação.

Para garantir o sucesso desses projetos, diferentes padrões são utilizados pelas agências espaciais e empresas que desenvolvem produtos nessa área. Entre eles estão European Cooperation for Space Standardization (ECSS), National Aeronautics and Space Administration (NASA) e International Organization for Standardization (ISO 9000).

Segundo Albuquerque (2011), o padrão ECSS é largamente utilizado na área espacial, além de ser o padrão utilizado pelo INPE na realização de uma missão espacial. O modelo ECSS é uma iniciativa criada para desenvolver um conjunto de normas espaciais para utilização por toda a comunidade espacial europeia. Essa iniciativa é apoiada pelo Conselho de Deliberações da Agência Espacial Europeia (ESA) desde 1994, assumindo um papel central.

Essa parceria entre a ESA, as Agências Espaciais Nacionais e as Indústrias Europeias foi estabelecida com os seguintes objetivos:

- Aumentar a eficácia dos programas espaciais na Europa, por meio da aplicação de um conjunto integrado de normas e requisitos que possa servir de base para os projetos espaciais.
- Facilitar a comunicação entre todas as partes envolvidas em uma missão de forma adequada, com referências ou citações em documentos.

- Garantir a qualidade e a segurança dos projetos espaciais e produtos.
- Reduzir riscos e garantir a interoperabilidade e a compatibilidade de interface das aplicações.
- Assegurar que os sistemas espaciais não causam, durante todo o ciclo de vida, risco para a vida humana, o ambiente, as propriedades pública e privada.
- Permitir que a indústria esteja pronta para responder a esses requisitos.
- Melhorar a competitividade da indústria espacial europeia.

Na estrutura do padrão ECSS existem três ramos paralelos para o gerenciamento de projetos, engenharia e garantia do produto, conforme mostra a Figura 4.5. Os documentos para o gerenciamento de projetos têm prefixo “M”, os documentos para a engenharia têm prefixo “E”, e os documentos padrão de qualidade têm prefixo “Q”.

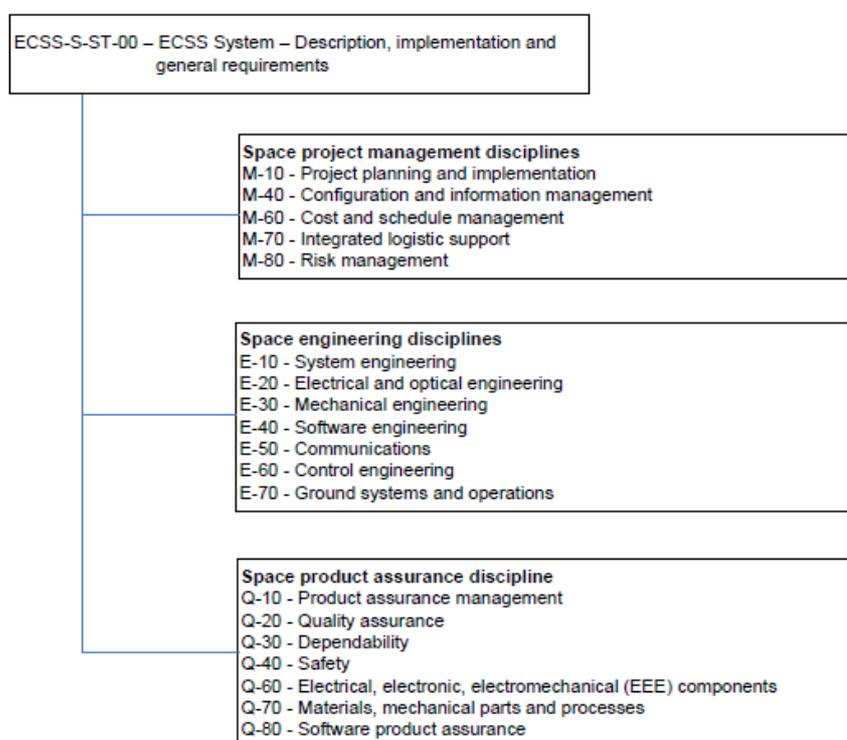


Figura 4.5 – Disciplinas do padrão ECSS

Fonte: ECSS (2009a).

Em cada ramo do padrão ECSS, as disciplinas e requisitos são cobertos por normas específicas e apoiados por manuais e memorandos.

- **Normas:** documento especificamente para uso em acordos entre empresas para a execução de atividades relacionadas à área espacial.
- **Manual:** documento não normativo que fornece informações mais detalhadas, como orientações, diretrizes, dados técnicos, pareceres ou recomendações sobre como implementar as atividades relacionadas à área espacial. Existem dois tipos de manual: as orientações e boas práticas e o tratamento de dados.
- **Memorandos técnicos:** documento não normativo que fornece informações úteis para a comunidade espacial sobre um tema específico.

Quando surgem novas exigências, são criados requerimentos adicionais, com texto justificado.

As normas que tratam de engenharia de *software* e qualidade de *software*, examinada por este trabalho, são Space Engineering *Software* (ECSS-E-ST-40) e Space Product Assurance – *Software* Product (ECSS-Q-ST-80), respectivamente.

A estrutura dessa norma de engenharia de *software* é aplicada a todos os níveis do projeto, desde as funções do sistema até *firmware*, incluindo funções críticas de segurança e de missão. Portanto, sua ênfase recai sobre a verificação e a validação de itens de *software*, sendo complementada pela garantia de produto espacial (norma ECSS-Q-ST-80).

A Figura 4.6 apresenta os diferentes processos de *software* atendidos pelas normas ECSS-E-ST-40 e ECSS-Q-ST-80, entre outras normas ECSS.

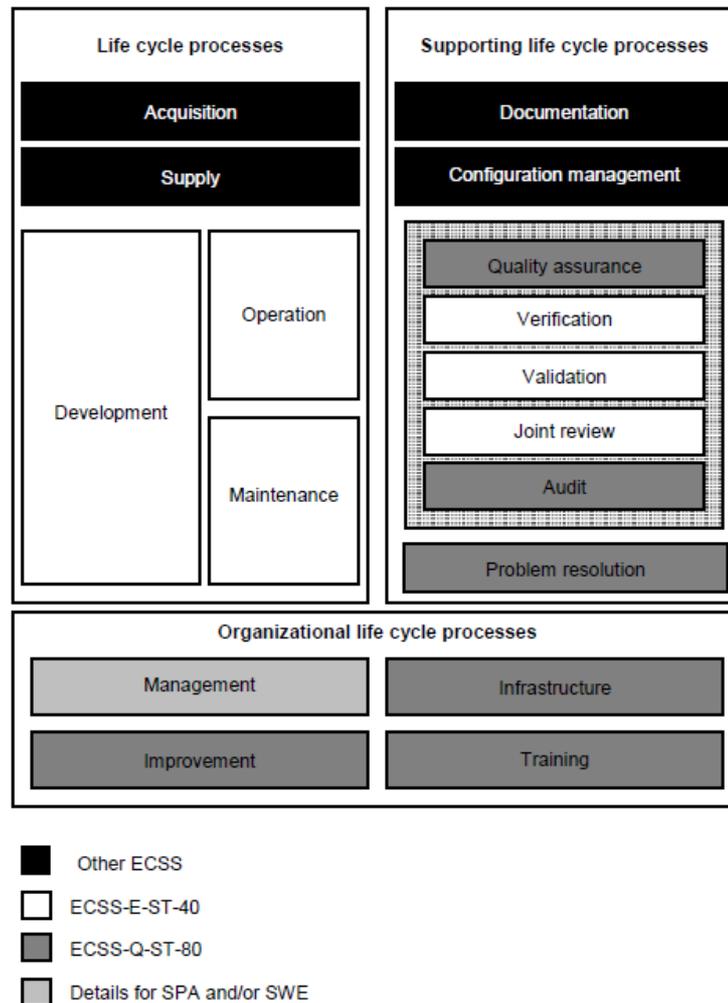


Figura 4.6 – Relacionamento dos processos de *software* no padrão ECSS

Fonte: ECSS (2009c).

4.2.1 Engenharia de Software Espacial (ECSS-E-ST-40)

Essa norma é organizada de acordo com os processos de *software* e suas atividades. Cada processo inclui atividades que devem ser decompostas em uma lista de tarefas na forma de requisitos do processo, produzindo resultados esperados. Ela não estabelece, de forma clara, quais atividades devem ser realizadas para garantir a confiabilidade do *software* em desenvolvimento. Determina que a Especificação de Requisitos de *Software* deve apontar os requisitos de confiabilidade aplicáveis ao componente (REGINATO, 2012). Sua estrutura está apresentada na Figura 4.7.

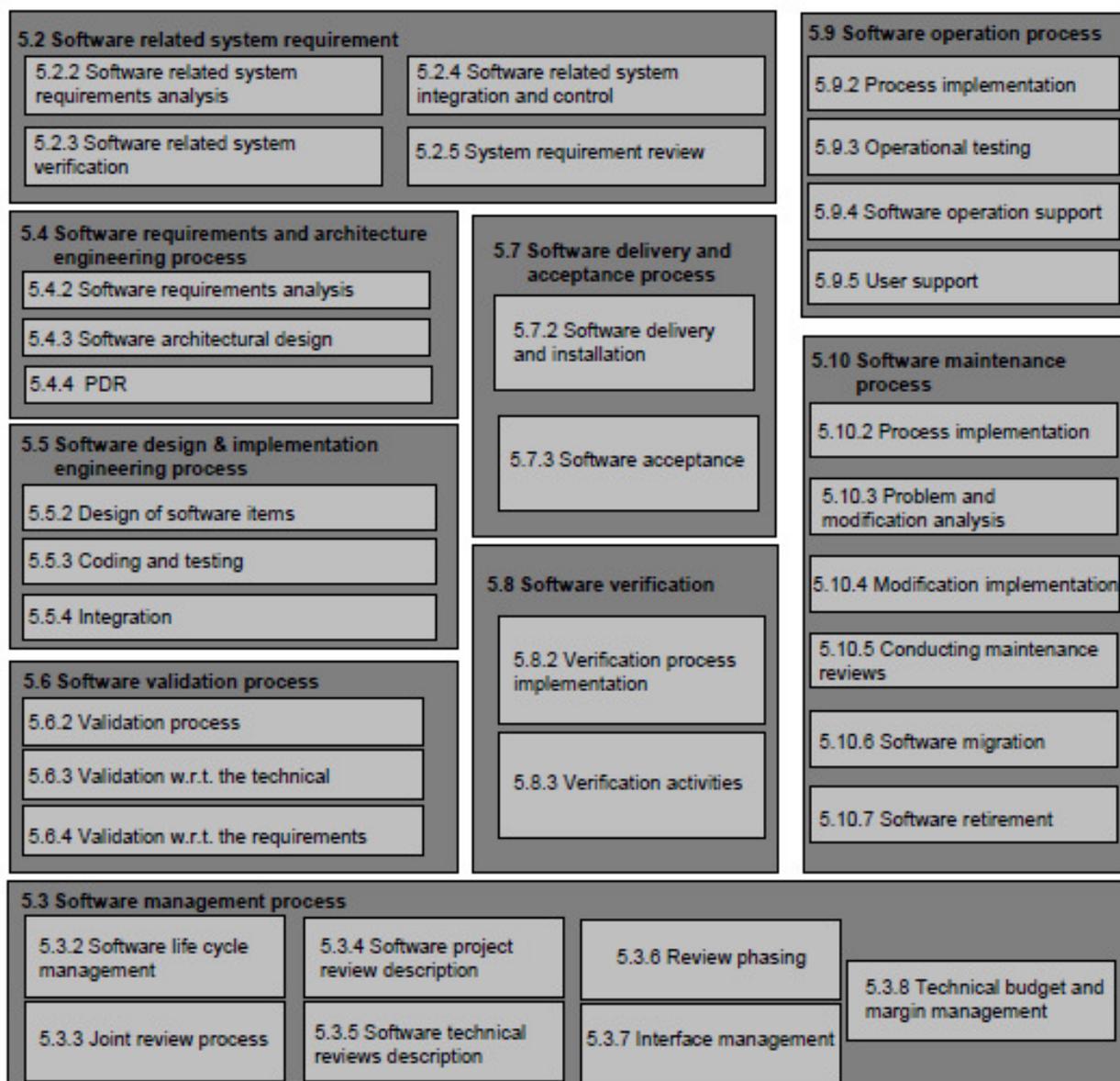


Figura 4.7 – Estrutura da norma ECSS-E-ST-40

Fonte: ECSS (2009c).

A gestão de projetos na área espacial é essencial para a geração de produtos e serviços, podendo envolver equipes de uma ou mais pessoas, de acordo com a fase, a complexidade, a criticidade e o custo do componente a ser produzido. Por isso, sua gestão exige um controle rígido, devendo acontecer em todas as fases do ciclo de vida do projeto e ser homologada por marcos pré-definidos. Uma fase só poderá ser considerada aprovada após a conclusão de sua

revisão de encerramento. Tal aprovação ocorre quando os responsáveis pela revisão consideram satisfatórios os documentos e produtos apresentados (ALBUQUERQUE, 2011).

Os requisitos para cada fase são definidos de forma que possam ser verificados e/ou medidos, a fim de comprovar que os resultados previstos para a fase atenderam ao que foi especificado.

4.2.2 Garantia de Produto Espacial – Garantia de Produto de Software (ECSS-Q-80)

O objetivo da garantia de produto do *software* é assegurar ao cliente e ao fornecedor que o *software* satisfará a suas exigências durante a vida útil do sistema e que ele foi desenvolvido para ser executado corretamente e com segurança em seu ambiente operacional. Tais exigências tratam a gerência e a estrutura de qualidade, as atividades do ciclo de vida, a definição de processo e as características de qualidade dos produtos.

A partir da definição dos requisitos de *software* de garantia de produto a serem atendidos em um projeto espacial particular, será realizada a gestão da qualidade e sua estrutura, as atividades do ciclo de vida, a definição do processo e as características dos produtos de qualidade. Essa norma aborda os aspectos fundamentais da relação cliente e fornecedor, assumida por todos. Seu principal objetivo é oferecer um nível de confiança adequada. Ela complementa a norma ECSS-E-ST-40, com os aspectos de garantia do produto integrados nos processos de engenharia de *software* do sistema espacial. Juntas, as duas normas especificam todos os processos de desenvolvimento de *software* espacial.

Software product assurance programme implementation	
5.1 Organization and responsibility	5.5 Procurement
5.2 Software product assurance programme management	5.6 Tools and supporting environment
5.3 Risk management and critical item control	5.7 Assessment and improvement process
5.4 Supplier selection and control	

Software process assurance
6.1 Software development life cycle
6.2 Requirements applicable to all software engineering processes
6.3 Requirements applicable to individual software engineering processes or activities

Software product quality assurance
7.1 Product quality objectives and metrication
7.2 Product quality requirements
7.3 Software intended for reuse
7.4 Standard ground hardware and services for operational system
7.5 Firmware

Figura 4.8 – Estrutura da norma ECSS-Q-80

Fonte: ECSS (2009b).

Essa norma é organizada em três partes principais, conforme apresentado na Figura 4.8:

- Implementação do programa de garantia de produto de *software*.
- Garantia de processo de *software*.
- Garantia de qualidade do produto de *software*.

5 SISTEMAS EMBARCADOS

Como vimos no Capítulo 4, as diferentes áreas de atuação envolvem requisitos diferentes, tanto para seu funcionamento quanto para seu desenvolvimento. Cada componente da missão espacial tem características e necessidades próprias, por isso, o uso de processos direcionados para as características desses componentes poderá contribuir para o aumento da qualidade de cada um deles. Neste capítulo serão apresentadas as características dos componentes de sistemas embarcados.

As definições de sistema embarcado e *softwares* embarcados são controversas, pois são encontradas com muitas variações na literatura (YEN; WOLF, 1996 apud SPINOLA, 1998). Em geral, os conceitos de *software* embarcado e sistema embarcado confundem-se, tornando difícil sua separação.

As contradições se iniciam a partir da nomenclatura utilizada para sua identificação: sistema embarcado, sistema embutido, *software* embutido e *software* embarcado. Neste último pode haver conflito com a identificação da parte computacional de um componente do sistema.

Dos conceitos encontrados na literatura, nesta tese foi adotado o seguinte: “um sistema embarcado é uma combinação de componentes, os quais são formados por *hardware* e *software* intimamente relacionados, projetados para realizar uma função específica” (BARR, 1999 apud AVELINO, 2009; BARROSO, 2010), no qual o *hardware* é a parte física do componente e o *software* embarcado é a parte lógica desse sistema. Tal conceito foi adotado porque é compatível com as características do sistema embarcado e porque adere à proposta apresentada.

Sistemas embarcados estão presentes em nosso cotidiano. Atualmente, a maioria dos componentes eletrônicos tem alguma parte computacional. Muitos desses componentes fazem parte de algum produto crítico, cuja falha pode levar a sérios danos, como é o caso do avião, da máquina de radioterapia, da máquina de hemodiálise, *Anti-lock Braking System (ABS)*, sistema de posicionamento global (GPS) etc. Sistemas embarcados podem estar presentes em produtos

que não oferecem tantos riscos, como brinquedos, máquinas fotográficas, entre outros. Nesses casos, em caso de falha, pode haver grande insatisfação do usuário.

Em geral, os sistemas embarcados apresentam restrições de desenvolvimento e de funcionamento, pois, para funcionar, devem atender a determinados requisitos, como consumo de energia, tamanho, peso, segurança e custo, e muitas vezes estão expostos a eventos externos.

Muitas dessas restrições dependem da utilização destinada ao sistema embarcado. Elas se tornam mais rígidas proporcionalmente à criticidade de seu uso. Por exemplo, um sistema embarcado em um marca-passo precisa ter tamanho aceitável para poder ser inserido no corpo humano de forma simples e cômoda, além de exigir tempo de resposta imediato para atender às necessidades do portador, baixo consumo de energia para minimizar substituições, baixo custo para que seu uso seja acessível e, por fim, seu funcionamento seja preciso.

Em sistemas embarcados, o atendimento aos requisitos não funcionais é fundamental para a execução e o desempenho de suas funcionalidades (requisitos funcionais), pois a eficiência depende também do tempo utilizado para se alcançar um resultado, de seu desempenho, do consumo de energia, da robustez, da confiabilidade, além do custo, para que seja viável.

A concepção do sistema inicia-se a partir de uma necessidade, formalizada pelo projetista por meio de uma especificação inicial do sistema, que, em geral, implica o estabelecimento de propriedades de componentes de *hardware* e de *software*, tais como requisitos específicos (consumo de energia, tipo de alimentação, comunicação com ambiente etc.). Com base nas especificações do sistema, as funcionalidades requeridas são mapeadas em tarefas concorrentes para serem implantadas em seus domínios de implementação (*hardware* ou *software*), conforme apresentado na Figura 5.1.

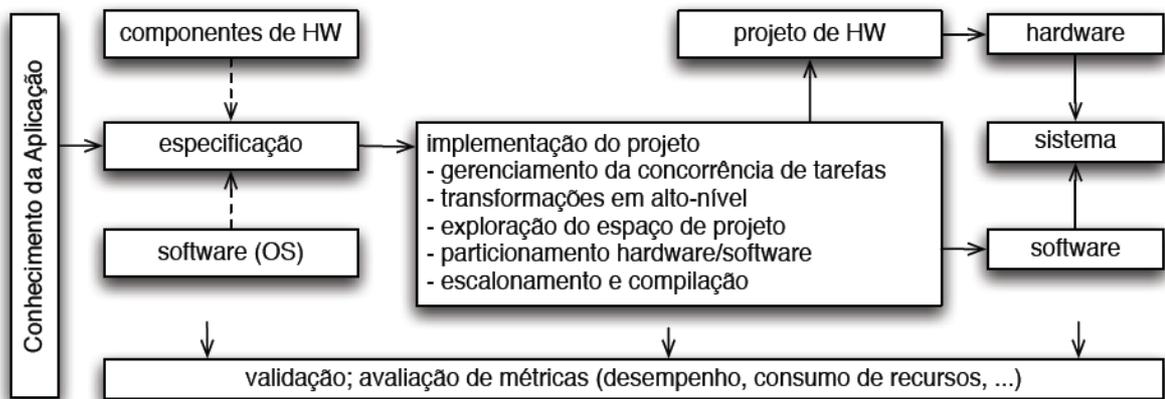


Figura 5.1 – Fluxo simplificado do projeto de sistemas embarcados

Fonte: MARWEDEL (2003) apud MARCONDES (2006).

Os sistemas embarcados têm características específicas com requisitos específicos de tecnologias e de componentes que determinam especialmente as estratégias a serem adotadas em seu processo de desenvolvimento (CARVALHO, 2010).

Um sistema embarcado é aquele cujo comportamento é definido por sua interação com seu ambiente, usualmente pelo sequenciamento entre um conjunto de modos, sendo que cada modo pode representar um estado ou algum processamento. Tais sistemas estão constantemente respondendo a eventos externos e calculando suas saídas como uma função de suas entradas e de seu estado corrente (SPINOLA, 1998).

A arquitetura do sistema embarcado é definida de acordo com os requisitos das tarefas e suas necessidades operacionais. Sua composição varia entre uma única unidade (único processador) e múltiplas unidades (vários processadores) (SOUZA, 2002).

Áreas de aplicação de sistemas computacionais embarcados (ROCHA, 2004)

- **Indústria aeronáutica**

- TCAS (Traffic Alert and Collision Avoidance System). Sistema computadorizado utilizado para evitar colisões.

- Fly-by-wire. O piloto controla as superfícies de controle de voo primárias do A320 da AIRBUS por meio de um *joystick*.

- **Indústria espacial**

- Sistemas de comando, controle e comunicação, cargas úteis, além dos serviços de solo.

- **Indústria ferroviária**

- Método B. Sistema de controle do metrô de Paris (método B do matemático Abrial).

- TVM430. Sistema de controle do trem “bala” francês TGV (Trains a Grand Vitesse).

- **Indústria automobilística**

- ABS. Sistema anti bloqueio que impede que as rodas travem numa freada, tornando-a mais segura.

- GPS. Sistema de navegação que orienta o usuário a respeito do caminho que deve seguir para chegar a determinado destino.

- **Indústria naval**

- *Electronic Chart and Display System (ECDIS)*. Sistema de navegação baseado em carta eletrônica.

- **Medicina**

- Marca-passos cardíacos eletrônicos. Dispositivo eletrônico microprocessado de alta tecnologia cuja finalidade é manter o ritmo cardíaco mínimo.

- Máquinas de radioterapia e máquinas de hemodiálise, dispositivos que tornam o procedimento seguro e eficaz.

Originalmente, os componentes eletrônicos eram estruturas de um circuito eletrônico. Com a evolução dos *softwares* e para diminuir o custo de sua produção, passou-se a utilizar *software* inserido em tais componentes, sendo esse o diferencial dos produtos. Assim, o mesmo componente eletrônico de um sistema embarcado pode assumir comportamentos diferentes, dependendo das ações realizadas pelo *software* inserido, tornando-o um amplo agregador de valores aos produtos em que estão inseridos.

Com o aumento considerável na quantidade de memória disponível e na capacidade dos processadores utilizados nas missões espaciais (KUCINSKIS, 2011), será possível o avanço no uso e na autonomia dos *softwares* embarcados, exigindo o desenvolvimento de práticas e atividades exclusivas para componentes de *software e de hardware*.

5.1 *Software* embarcado

“*Software* é: (1) Instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados; (2) estruturas de dados que permitem que os programas manipulem adequadamente a informação; e (3) documentos que descrevem a operação e o uso dos programas (PRESSMAN, 2006)”.

Por definição, todo *software* é um componente de um sistema construído com recursos de programação computacional, que pode ser um sistema total ou parte dele. Os *softwares* são desenvolvidos, mas não são manufaturados, portanto não envolvem fase de produção. Seus esforços concentram-se na engenharia e, geralmente, são feitos sob medida. O equivalente da fabricação ou produção para o *software* é a codificação. *Software* também não se desgasta, e suas falhas costumam acontecer, devido a defeitos em seu desenvolvimento ou em sua manutenção.

O *software* para sistema embarcado também pode ser um *firmware*, que é um conjunto de instruções operacionais programadas diretamente no *hardware*, no qual é armazenado em um circuito integrado ou *chip* de memória no momento de sua fabricação. Tais características do *software* embarcado se dão por questões de segurança, aplicação e, principalmente, por restrições computacionais e limites de escopo.

Geralmente, encontramos referências a *softwares* embarcados como produtos com capacidade de processamento reduzida, fazendo-os ser erroneamente confundidos com sistemas sem importância. A realidade, porém, é outra, uma vez que um *software* embarcado, a despeito de seu tamanho ou do recurso computacional utilizado, não é necessariamente um sistema de pouca importância. Ele pode estar embarcado em um brinquedo simples, no controle de foco de uma máquina fotográfica e até em sistemas que desenvolvem projeções para o mercado

financeiro, assim como no controle de todos os parâmetros necessários, minimizando o uso dos recursos da plataforma e cargas úteis em uma missão espacial.

Os *softwares* embarcados são capazes de realizar tarefas extremamente importantes dispondo de pouco recurso computacional.

A grande diferença de um *software* para sistema embarcado e um *software* convencional é a proximidade entre o *hardware* e o *software*, pois, de acordo com os requisitos, são impostas determinadas restrições de utilização de recursos, de tempo de resposta, entre outras.

Entre o *software* e o *hardware* existe uma categoria de *software* que inclui os sistemas que comandam sua execução, os *sistemas operacionais*. Os sistemas embarcados de forma geral e os sistemas operacionais simplificados interagem diretamente no *hardware*, como representado na Figura 5.2.

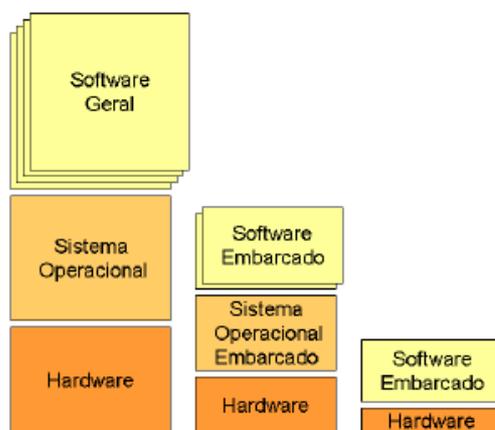


Figura 5.2 – Relação entre *software* embarcado e *software* geral

Fonte: GOMES (2010).

Essa categoria de *software* tem, usualmente, a função de monitorar, analisar ou controlar eventos do mundo real e reage com relação a esses eventos por meio de componentes específicos para coleta de dados do ambiente, análise de dados, transformação da informação, componentes de controles e monitoramento, entre outros, de acordo com os requisitos do sistema. A maioria dos *softwares* embarcados deve também considerar requisitos de tempo real.

As aplicações de sistemas de tempo real são bastante vastas e, na maioria dos casos, os sistemas de tempo real são embarcados (LABROSSE, 2009 apud AVELINO, 2009).

5.2 Hardware embarcado

Hardware embarcado é a parte física do computador, ou seja, é o conjunto de componentes eletrônicos, circuitos integrados e placas que se comunicam por meio de *hardware* e *software*, adaptados para uma tarefa particular, sendo geralmente parte de um sistema maior (KUMAR et al., 1996 apud SPINOLA, 1998).

Assim como no caso do *software*, o desenvolvimento do *hardware* também iniciou-se quase que artesanalmente, com sistematização de processo baixa e com grande dependência da habilidade do profissional (SAULO, 2010).

Muitas vezes, o *hardware* de um sistema embarcado é padronizado e o projeto do sistema compreende a configuração do *hardware* (com seus componentes definidos) e do *software*, que deve ser adaptado de acordo com as características específicas do *hardware*.

5.3 Desenvolvimento de sistemas embarcados

Em um sistema embarcado composto por *hardware* e *software*, o *software* é o componente que opera sobre o *hardware* de forma integrada, permitindo o processamento de informação. Segundo Keen (1996) apud Santos (2005), “o *software* é o que torna o *hardware* útil”.

Apesar de os sistemas embarcados, muitas vezes, serem associados a dispositivos limitados com relação à capacidade de processamento, memória e armazenamento de código, as abordagens mais recentes direcionam soluções para dispositivos capazes de desempenhar tarefas extremamente complexas, tanto nos aspectos relativos aos projetos de *hardware* quanto naqueles referentes aos projetos de *software*. Dessa forma, faz-se observar a importância dos testes no desenvolvimento de *software* embarcado.

O processo de especificação e modelagem de um sistema estende-se para o subsistema ou para o componente, e em geral se inicia com a descrição de uma especificação e termina com a descrição de um modelo. Ele deve contemplar as seguintes etapas:

- Uma especificação funcional, com requisitos explícitos e/ou implícitos com entradas, saídas, e estados internos.
- Propriedades a serem satisfeitas que possam ser validadas com a especificação funcional.
- Desempenho (medidas) de qualidade do projeto, como custo, confiabilidade, velocidade, tamanho etc., e dados como um conjunto de equações que envolva, entre outras coisas, entradas e saídas.
- Restrições sobre os índices de desempenho (EDWARDS et al., 1997).

5.3.1 Desenvolvimento de *software*

O desenvolvimento do *software* para sistemas embarcados tem como missão implementar suas funcionalidades, respeitando as restrições com as quais eles devem operar, tornando essa atividade um verdadeiro desafio.

O *software* embarcado se caracteriza pelos seguintes aspectos:

- Em geral, não utiliza periféricos de comunicação convencionais, pois é executado internamente em dispositivo sem ação direta de usuário.
- Reage aos eventos de acordo com o momento em que eles acontecem.
- É desenvolvido para operar de forma ininterrupta e autônoma por longos períodos.
- Pode causar catástrofes em caso de falha.
- Em geral, é desenvolvido utilizando algoritmos complexos e otimizações de *performance* para minimizar o consumo de recursos de CPUs, cumprindo assim suas restrições (DOUGLASS, 2000 apud NETO, 2002).
- É projetado e desenvolvido em paralelo com o *hardware* em que o sistema será executado.

- Tem dificuldade ou até impossibilidade de produção de protótipos, devido à interação do sistema com *hardware* (KOTONYA; SOMMERVILLE, 1997 apud NETO, 2002).
- Dispõe de tempo reduzido para seu desenvolvimento (STANKOVIC et al., 1996 apud NETO, 2002).
- Pode ser executado em várias plataformas.
- Considera ou tem como restrição o uso de *hardware* e *software* de mercado.

Há algum tempo, o *hardware* vem sendo produzido com uma das abordagens que visam à criação de componentes para reutilização. No caso do *software*, utilizando os conceitos da metodologia orientada a objetos, estão sendo conduzidos estudos e a implementação de componentes de *software* nos quais, analogamente, os programas passarão a ser tratados como uma coleção de componentes padrões conectados (SICKLE, 1997 apud SPINOLA, 1998).

Não é possível definir um processo único para o desenvolvimento dos inúmeros sistemas embarcados. As diferenças entre os fatores listados na Tabela 5.1 determinam grandes variações nos processos a serem estruturados para atingir os objetivos propostos.

Tabela 5.1 – Fatores a serem considerados em sistemas embarcados

SISTEMAS EMBUTIDOS: FATORES DIFERENCIADORES		
1	Componentes de hardware e software	Quantos e quais componentes de hardware e software são pré-definidos? Qual o grau de reuso requerido?
2	Distribuição	O sistema é distribuído? Qual o grau de distribuição do sistema?
3	Concorrência	Qual o grau de concorrência do sistema?
4	Confiabilidade	Qual o grau de confiabilidade requerido?
5	Interface homem-máquina	Qual o grau de exigência da interface homem-máquina requerido?
6	Segurança	Qual o grau de segurança operacional requerido?
7	Desempenho	Qual o desempenho requerido?
8	Forma	Quais as exigências relativas a tamanho e peso?
9	Consumo	Quais as restrições referentes a consumo e dissipação de energia?
10	Condições ambientais	Quais as exigências referentes a condições ambientais de operação?
11	Manutenção	Qual a natureza e qual a frequência de alterações requeridas? Qual a facilidade requerida para alteração do sistema?
12	Custo	Quais as restrições de custo requeridas?

Fonte: SPINOLA (1998).

O desenvolvimento de sistemas embarcados é, na realidade, um processo que necessita de duas abordagens diferentes, pois o desenvolvimento de *hardware* utiliza-se de métodos analíticos e ferramentas de apoio, enquanto o desenvolvimento de *software* embarcado não apresenta uma estrutura padrão ou básica que ofereça a mesma eficiência e estabilidade. Poucas ferramentas auxiliam o projeto de *software* embarcado, e existe uma deficiência no suporte às interfaces de *hardware*. Em geral, isso se deve ao fato de que as plataformas de *hardware* utilizadas por sistemas embarcados têm arquiteturas bastante distintas (MIRACHI; CARVALHO, 2009).

5.3.2 Ciclos de vida do software embarcado

Em geral, para o processo de desenvolvimento do *software* embarcado, não é praticado um paradigma de ciclo específico. Aplica-se o paradigma estruturado (adaptado), distribuído em cascatas de fases, como mostra a Figura 5.3. O final de uma fase implica o início de outra, reforçando a importância da execução das atividades de desenvolvimento com base em um modelo rígido para o desenvolvimento de *software* embarcado, uma vez que uma fase começa após a conclusão e a validação da outra, ou ocorre de forma sequencial (MIRACHI; CARVALHO, 2009).

Segundo ECSS (2009),

a adaptação para as restrições de desenvolvimento de *software* leva em conta as características especiais do *software* a ser desenvolvido e do ambiente de desenvolvimento [...] O tipo de desenvolvimento de *software* [...] e o sistema de destino, por exemplo, um processador embarcado.

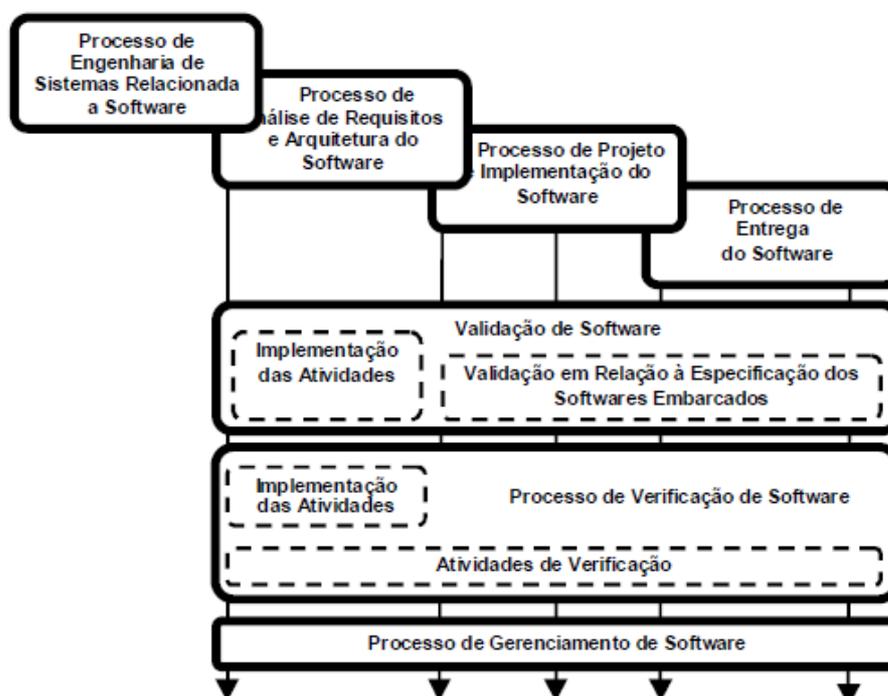


Figura 5.3 – Processo do ciclo de vida do *software* embarcado

Fonte: MIRACHI e CARVALHO (2009).

A capacidade de manter os projetos de *software* existentes foi ameaçada por projetos pobres com recursos inadequados, que geravam grande quantidade de defeitos e, por consequência, a insatisfação dos usuários. Muitos desses defeitos aumentavam a possibilidade de acidentes ou catástrofes, como no caso do *software* dos sistemas computacionais utilizados nas Olimpíadas de Atlanta, Estados Unidos, em 1996, que teve de ser retirado de operação. Outro exemplo foi o *software* implantado no Aeroporto de Denver, Estados Unidos, para manipulação de bagagens. Os mais de cem computadores interligados atrasaram a inauguração do aeroporto: a obra devia ter sido concluída em 1994, mas acabou sendo terminada em 1995 (GUERRA, 2009).

5.4 Disponibilidade

Um *software* é considerado como sendo de alta disponibilidade quando sua função necessita estar no ar o máximo de tempo possível, portanto, ele deve ser resistente a diferentes categorias de falhas, *hardware*, *software*, energia, mau uso, entre outros. Geralmente, um *software* com essa classificação é um dos componentes de um sistema que está diretamente envolvido ou

fundido com outros componentes e sua disponibilidade depende da resistência a falhas pelo conjunto (SAOTOME, 2009).

O mundo depende cada vez mais de serviços fornecidos por *softwares* com essa tecnologia, por isso é imprescindível a utilização de uma estratégia que garanta maior disponibilidade para seus *softwares*.

De acordo com a criticidade da necessidade do funcionamento desses componentes de *software* e da disponibilidade de recursos computacionais, haverá redundâncias para minimizar as falhas. Quanto mais redundâncias, maior será o custo de desenvolvimento e manutenção desses *softwares*.

A tolerância a falhas consiste basicamente em ter componentes redundantes que entrem em funcionamento automaticamente após a detecção de uma falha do componente principal. Em geral, esses componentes têm características de *hardware* (LAMAISON, 2007).

Os sistemas embarcados geralmente residem em máquinas que necessitam operar de forma contínua por muito tempo, de acordo com suas especificações e incluindo autocorreção de problemas, mau uso etc.

5.5 Acidentes

Os projetos de *software* estão ficando cada vez mais complexos. Com isso, a quantidade de defeitos e a conseqüente insatisfação dos usuários são crescentes em todas as categorias de *software*. Em especial pelo número expressivo de produtos com *softwares* embarcados, eles se tornam mais visíveis.

Diversos são os acidentes em que se apontam o *software* como causa, porém o relatório do acidente com o Ariane 5 é uma amostra mais fiel da necessidade dos sistemas embarcados críticos.

Uma missão espacial e seus sistemas são conceitualmente definidos como altamente críticos, desenvolvem recursos tecnológicos de última geração, capacitam recursos humanos, utilizam

somas vultosas de recursos financeiros e multiplicam exponencialmente a responsabilidade de projetos com esse porte.

Entendendo a grandiosidade e a criticidade dos sistemas espaciais, além da necessária confiança no funcionamento dos *softwares* utilizados, devemos sempre levar em consideração o sistema como um todo, respeitar e cumprir todas as normas determinadas em cada fase da missão, atendo-nos às particularidades do *software*. Um exemplo disso é o caso do Ariane 5, em que foi aproveitado um pacote de *software* de navegação do Ariane 4 que não tinha erros. No módulo Sistema de Referência Inercial (SRI), uma conversão de valores de 64-bits para 16-bits causou um operando inválido que o interrompeu, fazendo que o computador de bordo modificasse a trajetória do foguete e provocando a ruptura de juntas, levando à ativação da autodestruição.

Na conclusão do acidente com o Ariane 5 foi emitida a seguinte sentença:

Retornando ao erro de *software* – *software* é a expressão de um projeto detalhadamente elaborado e não admite falhas, como sistemas mecânicos. Além disso, linguagens de *software* são altamente flexíveis e expressivas, dessa forma podem solicitar requisitos que, por sua vez, levem a implementações complexas e de difícil validação [...]. A anomalia ocorrida não foi causada por um erro aleatório, e sim por um erro de projeto. Foi detectada e tratada inadequadamente, pois o *software* deveria ser considerado correto até demonstrado o contrário. A comissão tem razões para acreditar que tal ponto de vista é utilizado também em outras áreas de projeto de *software* do Ariane 5. A comissão é de opinião contrária – o *software* será passível a falhas enquanto não for avaliado com os métodos mais poderosos para demonstrar sua confiabilidade (SOMMERVILLE, 2011).

O *software*, por si só, não provoca acidentes. Contudo, um erro no projeto de *software* embarcado em um sistema crítico pode provocar um desastre de grandes proporções (ALONSO, 1998). Isso quer dizer que uma falha provocada por um *software* crítico pode pôr em risco a missão. Seu desenvolvimento deve ser submetido a práticas criteriosas, seguindo processos específicos com estratégia de coleta de requisitos, desenvolvimento, verificação e implantação confiáveis que levem em consideração as falhas típicas dos componentes de *software*.

6 TRABALHOS CORRELATOS

6.1 Diretrizes para o desenvolvimento de *software* de sistemas embarcados

Tese de doutorado: 1998 – Prof. Dr. Mauro de Mesquita Spinola – Poli-USP (SPINOLA, 1998). Identificação na Tabela 6.3 de correlatos (a).

6.1.1 Conceito geral da proposta

O foco adotado por (SPINOLA, 1998) foi caracterizar sistemas embarcados (produto e processo) e propor diretrizes para o desenvolvimento de *software* para esses sistemas e a interação entre seus componentes, conforme apresentado na Figura 6.1.

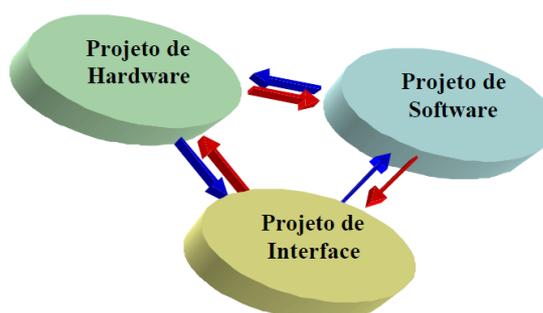


Figura 6.1 – Interação entre as atividades de projeto

Fonte: SPINOLA (1998).

6.1.2 Atividades gerais do processo

- **Análise de soluções candidatas para projeto detalhado:** dá suporte para a avaliação e a decisão em cada uma das atividades de detalhamento do projeto. Pode utilizar protótipos lógicos e físicos e é realizada com base no detalhamento do modelo do sistema. Considera aspectos metodológicos e aspectos de gerência de processo com ênfase em *software*.
- **Projeto detalhado de *hardware*:** envolve o detalhamento do projeto de *hardware* e a geração de lista de componentes, em que são considerados os diversos aspectos que podem ser utilizados, desde o desenvolvimento completo até o reuso dos componentes.

- **Projeto detalhado de *software*:** envolve o detalhamento da parte de *software* do projeto que desenvolve a estrutura de informação e seu tratamento.
- **Projeto detalhado de interface:** esta subtarefa envolve as atividades de detalhamento do *hardware* e do *software* voltadas a consolidar a conectividade do sistema.
- **Recomendações para adaptação:** são apresentadas as recomendações para adaptação desta tarefa de acordo com as características específicas do sistema, determinadas pela análise dos fatores diferenciadores.

Tabela 6.1 – Tarefas do desenvolvimento de sistemas embarcados

SISTEMAS EMBUTIDOS: TAREFAS DO DESENVOLVIMENTO
1. Requisitos do sistema
2. Decomposição do sistema
3. Arquitetura do sistema
4. Projetos de hardware e software
5. Implementação
6. Integração de sistema
7. Verificação e validação

Fonte: SPINOLA (1998).

Para cada uma das etapas do desenvolvimento de sistemas embarcados foram apresentados na Tabela 6.1 os elementos de processo requeridos, assim como as subtarefas que compõem cada tarefa. Como o foco do trabalho foi o desenvolvimento do *software*, não houve detalhamento dos requisitos do desenvolvimento dos componentes físicos (*hardware* e, eventualmente, mecânica).

6.1.3 Atividades de gerenciamento de sistema embarcado

Foram apresentadas as diretrizes para as atividades de gerência de processo de sistemas embarcados, com a identificação dos processos de apoio ao desenvolvimento e das práticas para definição e melhoria do conjunto de processos com a identificação das principais áreas de processos a serem estruturadas por uma organização que desenvolve sistemas embarcados;

foram utilizados como base os modelos SW-CMM (que focaliza *software*) e SE-CMM (que focaliza sistemas de engenharia), apresentados na Tabela 6.2.

Tabela 6.2 – Gerência do desenvolvimento de sistema embarcado

ÁREAS DE PROCESSO GERENCIAIS	
Áreas de processo de projeto	
Fornecem a infra-estrutura de gerência técnica necessária para o desenvolvimento dos sistemas. (BATE et al., 1995)	
1	Garantia da qualidade: avaliar a qualidade dos sistemas e do processo de desenvolvimento utilizado
2	Gerência da configuração: estabelecer e manter a integridade dos produtos desenvolvidos
3	Gerência de riscos: identificar, avaliar, monitorar e mitigar riscos
4	Planejamento e gerência de projeto: estabelecer planos razoáveis para os projetos de sistemas
Áreas de processo organizacionais	
Fornecem a infra-estrutura organizacional necessária para o desenvolvimento dos sistemas. (BATE et al., 1995)	
5	Coordenação intergrupos: integrar e estabelecer relações ativas e eficazes entre os grupos e indivíduos envolvidos no desenvolvimento
6	Gerência da evolução da linha de produtos: determinar a evolução dos produtos e gerenciar os recursos técnicos e serviços necessários para efetivar sua evolução
7	Gerência da tecnologia: fornecer o ambiente necessário ao desenvolvimento do sistema e à realização do processo
8	Treinamento: desenvolver as habilidades e o conhecimento necessários
9	Gerência de subcontratos: selecionar subcontratados e gerenciá-los

Fonte: SPINOLA (1998).

Os processos gerenciais foram divididos em duas categorias: áreas de processos de projeto e áreas de processos, baseadas na estrutura de processos do modelo SE-CMM. O foco do trabalho foi a apresentação da utilização dos diagramas da UML para o desenvolvimento de *software* embarcado.

6.2 Desenvolvimento de sistemas embarcados para aplicações espaciais

Artigo – Samoel Mirachi, Francisco das Chagas Carvalho “*Brazilian Symposium on Aerospace Eng. & Applications*” e “*3rd CTA-DLR Workshop on Data Analysis & Flight Control*” by AAB September 14-16, 2009, S. J. Campos, SP, Brazil (MIRACHI e CARVALHO, 2009). Identificação na Tabela 6.3 de correlatos (b).

6.2.1 Conceito geral da proposta

Proporcionar um ambiente para avaliação de qualidade de componentes de *software*, para ser utilizado no desenvolvimento de sistemas embarcados.

Foi apresentado o desenvolvimento de *software* de tempo real para sistemas embarcados em aplicações espaciais, utilizando diagramas modelados para uma aplicação específica de uma plataforma suborbital de microgravidade, tendo como principais funções o cálculo da lei de controle, o gerenciamento de bordo de dados de serviço e de carga útil, e a telemetria/telecomando com o *Electrical Ground Support Equipment (EGSE)*.

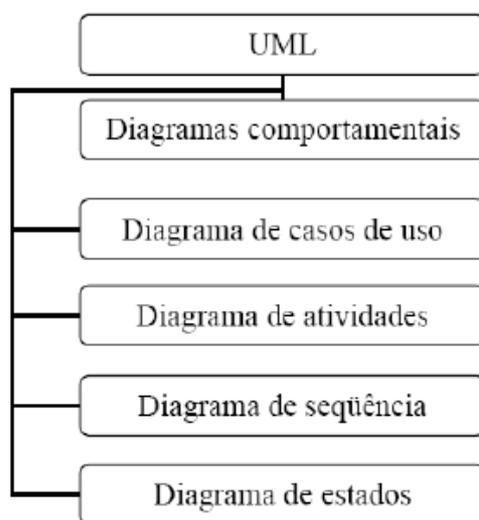


Figura 6.2 – Fluxo de funcionamento

Fonte: MIRACHI e CARVALHO (2009).

Para obter maior flexibilidade no desenvolvimento de *software* foi utilizado o modelo em cascata revisto, que prevê a possibilidade de, a qualquer tarefa do ciclo, regressar a uma tarefa anterior, de forma a contemplar alterações funcionais ou técnicas, caso seja necessário. Para cada fase foram definidas atividades e atribuídos diagramas da UML, como padrão de modelagem, conforme mostrado na Figura 6.2.

6.3 Model-Based Codesign

Artigo – Stephan Schulz Jerzy W. Rozenblit University of Arizona Michael Mrva Klaus Buchenrieder Siemens AG – IEEE (SCHULZ et al., 1997). Identificação na Tabela 6.3 de correlatos (c).

6.3.1 Conceito geral da proposta

Este trabalho apresenta uma abordagem *codesign* que permite aos desenvolvedores criar modelos de representação para um sistema formal cuja representação é a implementação do *hardware* e do *software*. Os desenvolvedores utilizam a simulação de modelagem, baseada em protótipos virtuais e seguida de mapeamentos interativos de especificação em uma arquitetura de *software* e *hardware*.

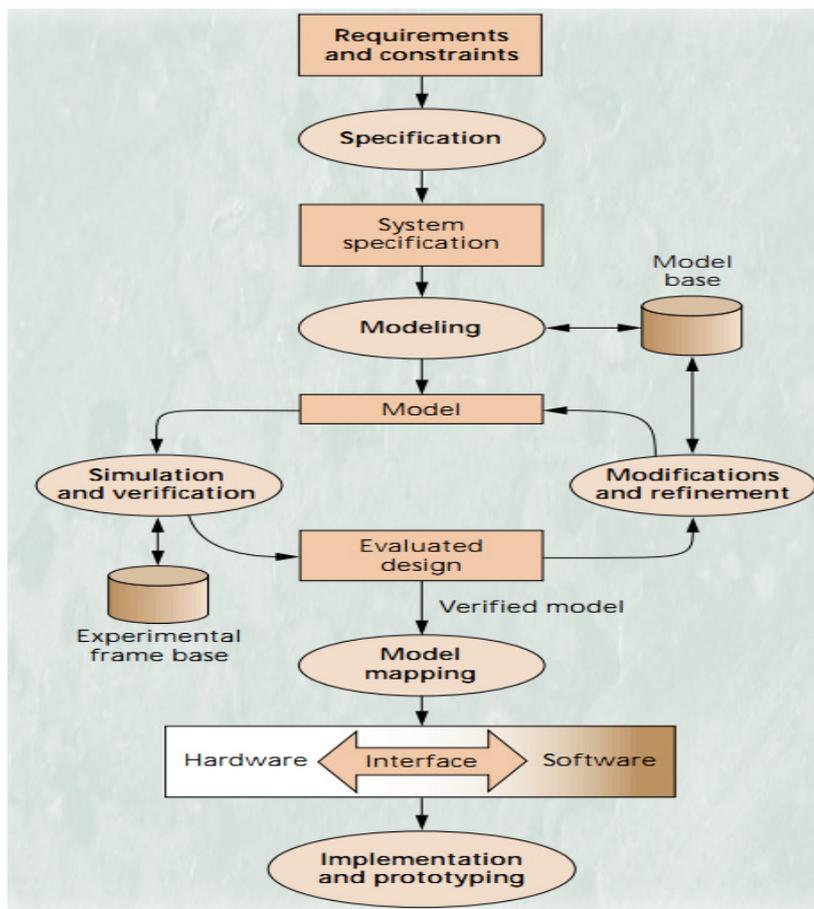


Figura 6.3 – Fases do modelo baseado em codesign

Fonte: SCHULZ et al (1997).

O objetivo de cada fase apresentada na Figura 6.3 é fornecer aos desenvolvedores uma maneira independente de avaliar por meio da simulação de sistema baseada em modelo, em que os modelos são mapeados para uma arquitetura de *software* e *hardware*. Essa abordagem enfatiza uma integração estruturada.

6.4 Estudo de requisitos do *software* embarcado no segmento da telemedicina

Artigo – Márcia Kondo, José R. Silva, Adilson Hira, Marcelo Zuffo (Kondo et al., 2011). Identificação na Tabela 6.3 de correlatos (d).

6.4.1 Conceito geral da proposta

Neste trabalho foi apresentado um estudo comparativo da análise de requisitos de sistemas embarcado. Utilizou-se uma metodologia baseada nas técnicas de análise de requisitos da engenharia e de design do *software*, levando em consideração as restrições funcionais, como o método Volere, apresentado na Figura 6.4.

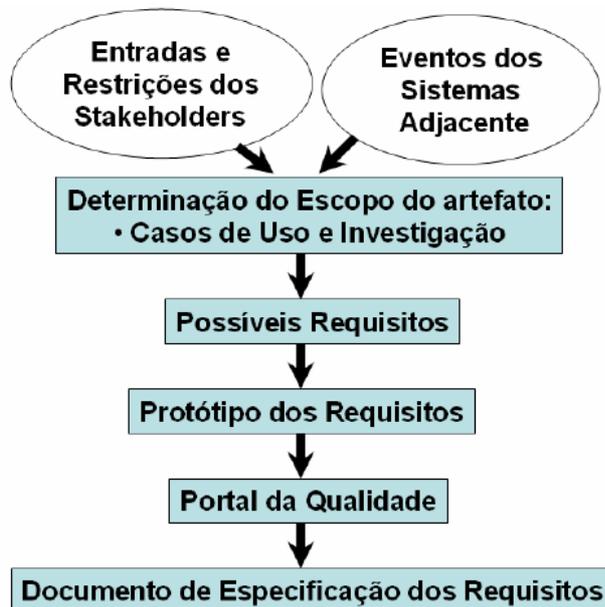


Figura 6.4 – Esquema do método Volere

Fonte: KONDO et al (2011).

6.4.2 Atividades gerais do processo

Os autores apresentaram um processo de criação de *software* segundo a perspectiva da engenharia de sistemas, em que foram apontadas atividades que devem ser consideradas nos componentes de desenvolvimento de *software*: métodos, procedimentos e ferramentas.

Como critério foi escolhido um método que fosse bem documentado e adaptável ao domínio da telemedicina, técnica específica desse domínio que tem uma sequência criteriosa e bem definida. Dentre os métodos estudados, selecionou-se o Volere, por ter obtido a maior aderência aos critérios de seleção.

Segundo Kondo et al., 2011), a abordagem do método conduziu à obtenção dos requisitos de forma estruturada, dividindo eficientemente o sistema em partes menores, utilizando a representação dos casos de uso. Os requisitos obtidos focam o ator, fazendo que o sistema permita maior interação com os diversos atores. O processo de obtenção de requisitos seguiu os passos previstos pelo método Volere.

6.5 Definição de um processo de engenharia de requisitos para software embarcado na indústria automotiva baseada em uma arquitetura de processos de *software*

Artigo – Luiz Carlos M. Ribeiro, Cristiane Soares Ramos, Maylon Felix Brito, Rejane M. C. Figueiredo (RIBEIRO et al., 2011). Identificação na tabela 6.3 de correlatos (e).

6.5.1 Conceito geral da proposta

Neste trabalho os autores apresentaram um processo padrão para a engenharia de requisitos de sistemas embarcados, focado na indústria automotiva. Esse processo foi elaborado a partir de uma arquitetura de *software* baseada no metamodelo *Software* e definido a partir da reutilização de componentes. O processo resultante está em conformidade com os modelos MR-MPS e *Automotive* SPICE, como apresentado na Figura 6.5.

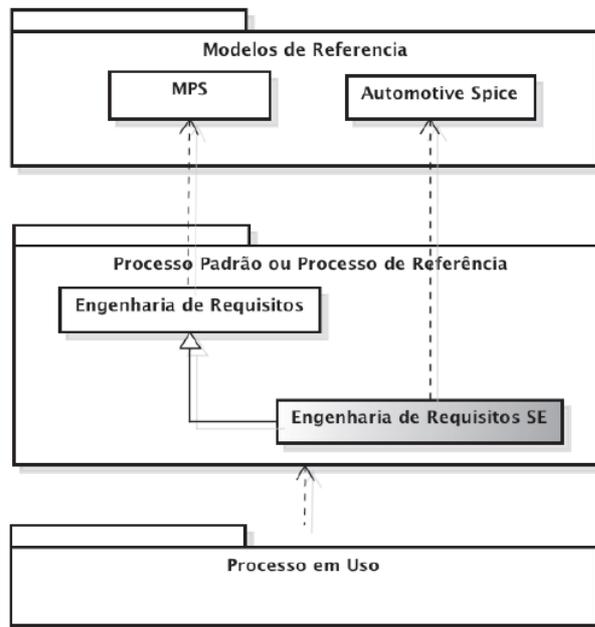


Figura 6.5 – Arquitetura de processos

Fonte: RIBEIRO et al (2011).

6.5.2 Atividades gerais do processo

O trabalho apresentou um estudo de comparação entre o modelo MR.MPS e o modelo SPICE, além de uma proposta de processo padrão de engenharia de requisitos com aderência ao MR-MPS. A partir desse processo padrão, foi gerado um processo em uso para a engenharia de requisitos de sistemas embarcados no contexto da indústria automotiva. Empregou-se a ferramenta *Eclipse Process Framework* (EPF) para a implementação dos conceitos do metamodelo, e utilizaram-se os conceitos de reutilização e os tipos de variabilidade de processo. As atividades e tarefas específicas para sistemas embarcados foram baseadas no modelo *Automotive SPICE*.

O processo padrão construído também foi baseado no reúso de um processo preexistente, com o objetivo de aproveitar conhecimentos já embutidos no processo de engenharia de requisitos de uso geral. Foi mantida a aderência ao modelo MR-MPS e apresentada a redução do esforço empregado em relação ao desenvolvimento sem o reúso. Não houve avaliação formal do processo.

6.6 Metodologia de desenvolvimento ágil em sistemas embarcados

Utilização de SCRUM para o desenvolvimento de sistema embarcado (INTELLIGES, 2011). Identificação na Tabela 6.3 de correlatos (f).

A Empresa Intelliges utiliza o processo V-Model (Figura 6.6), que é um modelo linear e sequencial, para buscar a melhoria contínua por meio da constante verificação e validação dos requisitos de projeto ao longo do desenvolvimento.

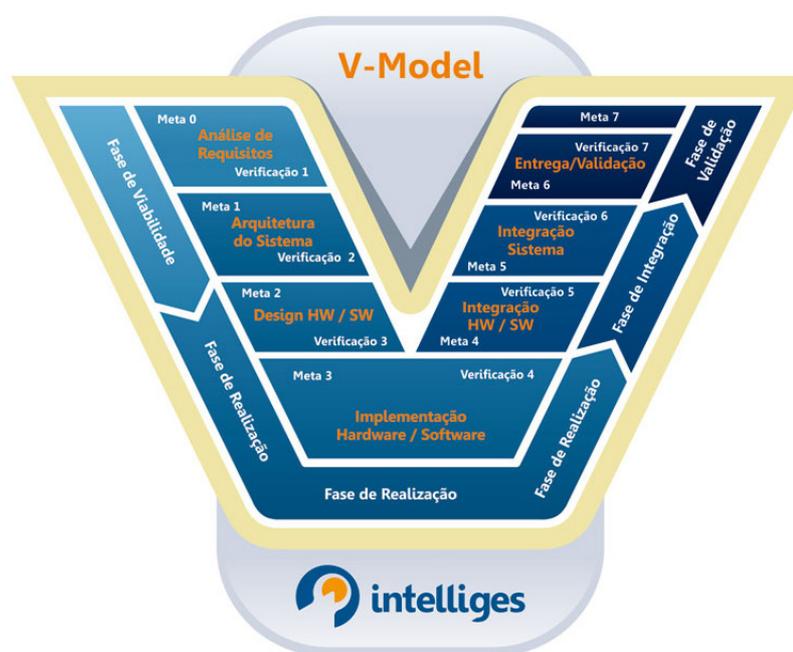


Figura 6.6 – V-Model

Fonte: INTELLIGES (2011).

Em projetos de inovação nos quais é difícil prever a maioria dos requisitos no momento inicial do desenvolvimento, é utilizado o processo SCRUM (Figura 6.7), adaptado a projetos de *hardware* e *software* de sistemas embarcados. Trata-se de um processo ágil, iterativo e incremental, e, nessa adaptação, as iterações relativas ao desenvolvimento de *hardware* têm períodos de duração mais longos dos que as iterações relativas ao *software*.

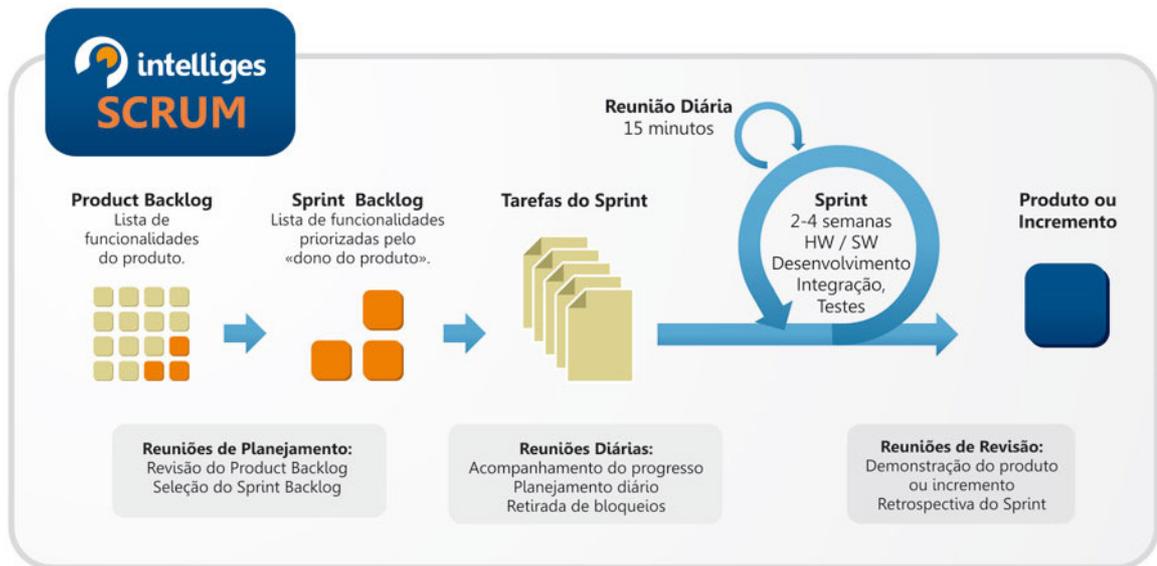


Figura 6.7 – Organização SCRUM para sistema embarcado

Fonte: INTELLIGES (2011).

6.7 Metodologia de projeto de *software* embarcada voltada ao teste

Dissertação – Humberto Varga Gomes, Universidade Federal do Rio Grande do Sul (GOMES, 2010). Identificação na Tabela 6.3 de correlatos (g).

6.7.1 Conceito geral da proposta

Este trabalho apresenta uma metodologia de desenvolvimento e teste que aponta os principais detalhes de execução, de organização e estruturação do código com relação à manutenção da compatibilidade do *software* e seu uso em dispositivos de *hardware*. Foi desenvolvido um *software* de acordo com a metodologia apresentada na Figura 6.8.

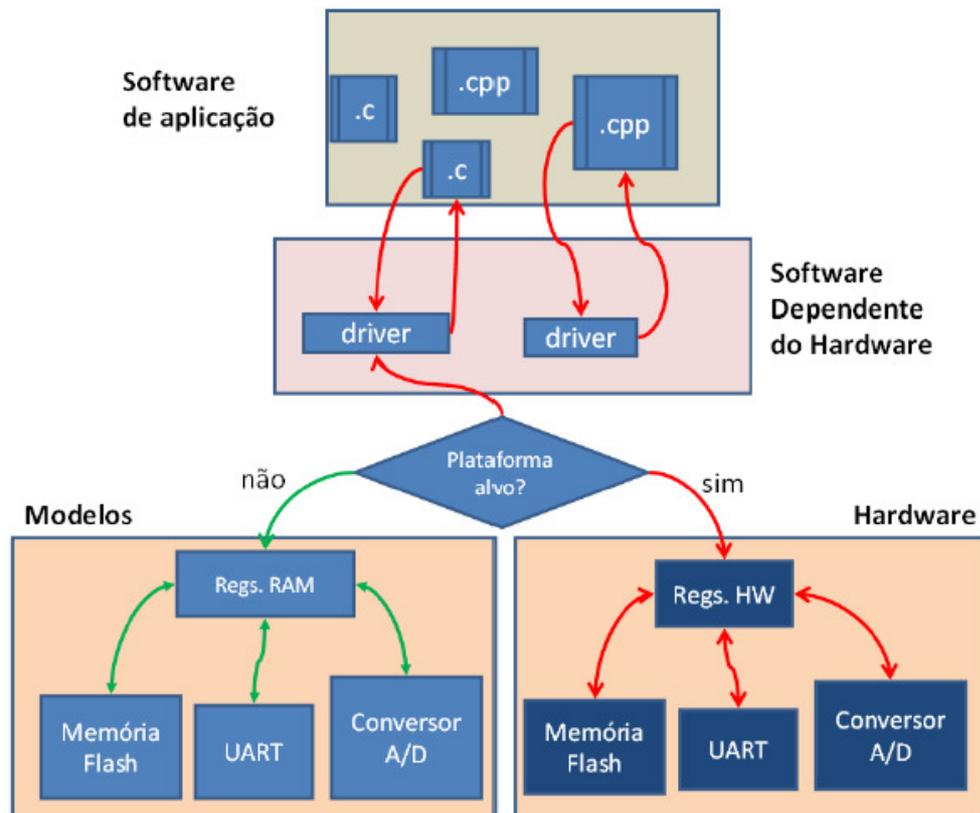


Figura 6.8 – Visão geral do *software* produzido

Fonte: GOMES (2010).

6.7.2 Atividades gerais do processo

Modelo de desenvolvimento

Neste trabalho foram apresentados todos os conceitos e características referentes à forma de organização de *softwares* embarcados. Além disso, mostrou-se, de forma integradora, uma abordagem de construção de *softwares* dos modelos dos dispositivos de *hardware*. Também foram detalhadas questões referentes às condições de execução dos testes de *hardware* e de *software*.

Foi utilizada uma estrutura generalizada para sistematizar a construção de componentes com modelagem de aspectos externos ao *hardware* e foram usados modelos para a execução e a

validação do componente. Empregaram-se, com total compatibilidade, modelos de dispositivos de *hardware* em diferentes projetos de *software* embarcado.

Também foi apresentada uma proposta de desenvolvimento de *software* com detalhamento das atividades de desenvolvimento especialmente direcionadas para os requisitos de testes de *software*, porém atendendo a todas as atividades necessárias para o desenvolvimento de *software*.

6.8 Diretrizes para desenvolvimento de *software* para sistemas embarcados

Monografia – Leonardo de Andrade Prates – 2012 (PRATES, 2012). Identificação na Tabela 6.3 de correlatos (h).

6.8.1 Conceito geral da proposta

O modelo foi baseado principalmente nas práticas das metodologias ágeis XP, buscando soluções mais rápidas para atender às oportunidades do mercado. Segundo Prates (2012), uma abordagem tradicional não traria os resultados esperados em tempo hábil, devido à série de artefatos formais que devem ser gerados antes de qualquer geração de *software* útil.

O modelo de processo para o desenvolvimento de sistemas embarcados proposto é apresentado na Figura 6.9, que representa uma mistura do processo evolucionário com a metodologia ágil XP, com as atividades do processo e as práticas da metodologia XP que foram consideradas importantes.

integração, teste de regressão, teste de aceitação) e procedimentos para controle de versões de *software/sistema*.

6.9 *Model-Driven Engineering of Complex Embedded Systems: Concepts and Tools*

Artigo – Flávio Rech Wagner, Francisco A. M. Nascimento, Marcio F. S. Oliveira. (WAGNER; NASCIMENTO; OLIVEIRA, 2011). Identificação na Tabela 6.3 de correlatos (i).

6.9.1 Conceito geral da proposta

Os objetivos deste trabalho são resolver a dificuldade de abstração e melhorar o nível de automação do projeto, desde a especificação inicial até o sistema final, utilizando os métodos de modelagem, formalismos e abstrações adequadas para especificar, analisar, verificar e sintetizar sistemas embarcados de forma rápida e precisa.

O esforço no uso deste modelo originou-se com o uso de ferramentas *Computer Aided Software Engineering (CASE)*, que, por meio de representações gráficas e conceitos de programação, geram automaticamente a implementação do código para, assim, reduzir o esforço da codificação e da depuração manual.

Este trabalho é uma nova abordagem para o *Model-Driven Engineering (MDE)*, que foi uma proposta para melhorar a gestão da complexidade e da capacidade de reutilização dos artefatos. O método de MDE representado na Figura 6.10 eleva o nível de abstração de design e fornece mecanismos para melhorar a portabilidade, a interoperabilidade, a facilidade de manutenção e a reutilização de modelos.

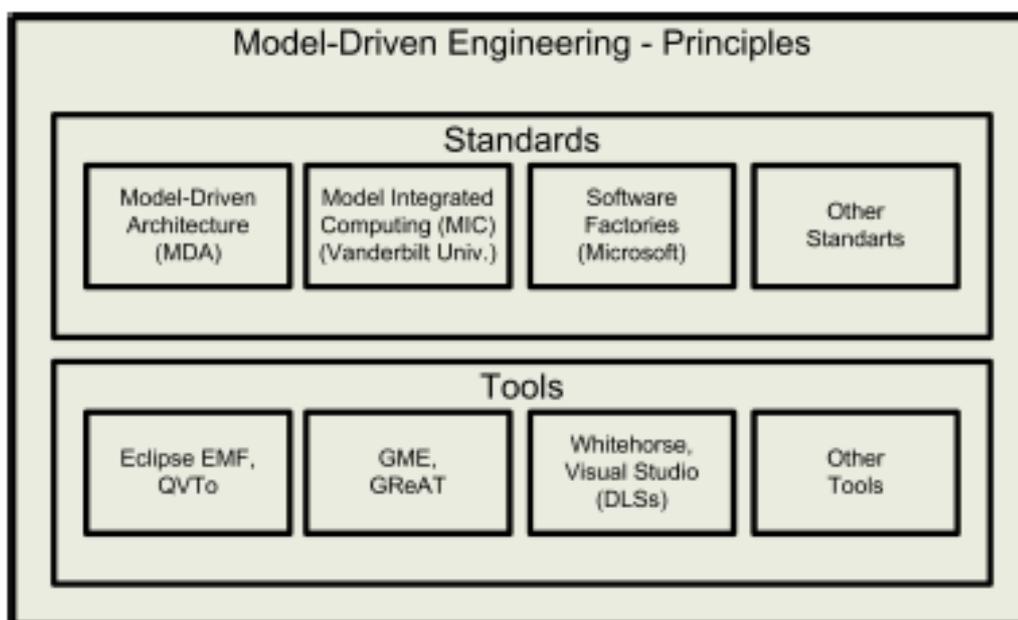


Figura 6.10 – *Model Driven*

Fonte: WAGNER, NASCIMENTO e OLIVEIRA (2011).

6.9.2 Atividades gerais do processo

Na abordagem deste trabalho foi utilizado o conceito de metafacilidade de objeto (MOF), uma representação padrão para os metamodelos e os modelos propostos pela OMG. Para definir a representação interna foram determinados modelos internos em conformidade com os metamodelos, representando as aplicações, capturando funcionalidades, indicando plataformas e recursos de *hardware/software* para realizar o projeto e as validações.

O método proposto nessa abordagem MDE tem construção automática de design realizado pelo produto, com uma abstração que permite a representação comum para várias atividades de design.

Neste trabalho foi utilizado o conceito inicial e fundamental de MDE, segundo o qual “Tudo é um modelo”, isto é, modelos desempenham um papel central no processo de desenvolvimento. Os conceitos básicos para apoiar o princípio MDE são o sistema, o modelo, o metamodelo e as relações entre eles, de modo que um modelo represente um sistema e esteja em conformidade com um metamodelo.

O modelo tem estruturas tecnológicas de ferramentas que dão suporte a diferentes operações e tarefas comuns para MDE, independentemente do domínio do aplicativo, de acordo com determinadas normas. Para definir as relações entre os modelos e refiná-los, foi utilizado um tipo de ferramenta que adota uma estrutura para MDE que permite aplicativos específicos para o domínio dos ambientes de desenvolvimento usados. Ferramentas como essa dependem de uma estrutura de tecnologia MDE para projetar não apenas o *software*, mas sistemas inteiros, como *hardware*, elétrica e mecânica. No artigo foram apresentadas algumas dessas ferramentas para desenvolvimento de sistema embutido.

O artigo apresentou uma introdução ao MDE aplicada para o desenvolvimento de sistemas embarcados complexos. Nessa abordagem foram realizadas transformações entre os modelos utilizados, a partir da modelagem UML de um aplicativo, que consiste em classes e diagramas de sequência, de forma que sua representação interna possa ser usada por ferramentas de verificação formais.

O modelo obtido capta aspectos estruturais de uma aplicação padrão utilizando uma hierarquia de módulos e processos, bem como aspectos comportamentais.

Foi definido um projeto de metamodelo que pode ser manipulado por transformação usando regras para implementar restrições de projeto e gerar o projeto candidato. Foram utilizados os modelos UML/MARTE para gerar regras de transformação, poupando o tempo que seria gasto com avaliações desnecessárias.

6.10 MDEReq: uma abordagem para engenharia de requisitos no domínio de *software* embarcado

Artigo – Milena Rota Sena Marques, Eliane Siegert e Lisane de Brisolara (MARQUES; SIEGERT; BRISOLARA, 2013). Identificação na Tabela 6.3 de correlatos (j).

6.10.1 Conceito geral da proposta

Os autores apresentaram uma proposta utilizando a engenharia de requisitos orientada a modelos no desenvolvimento de *software* embarcado para buscar consistência em seus modelos. A engenharia de requisitos foi focada para realizar a rastreabilidade por meio do uso

de notações da SysML e modelos construídos usando UML e MARTE integrados, utilizando ferramentas para a modelagem do projeto, conforme apresentado na Figura 6.11.

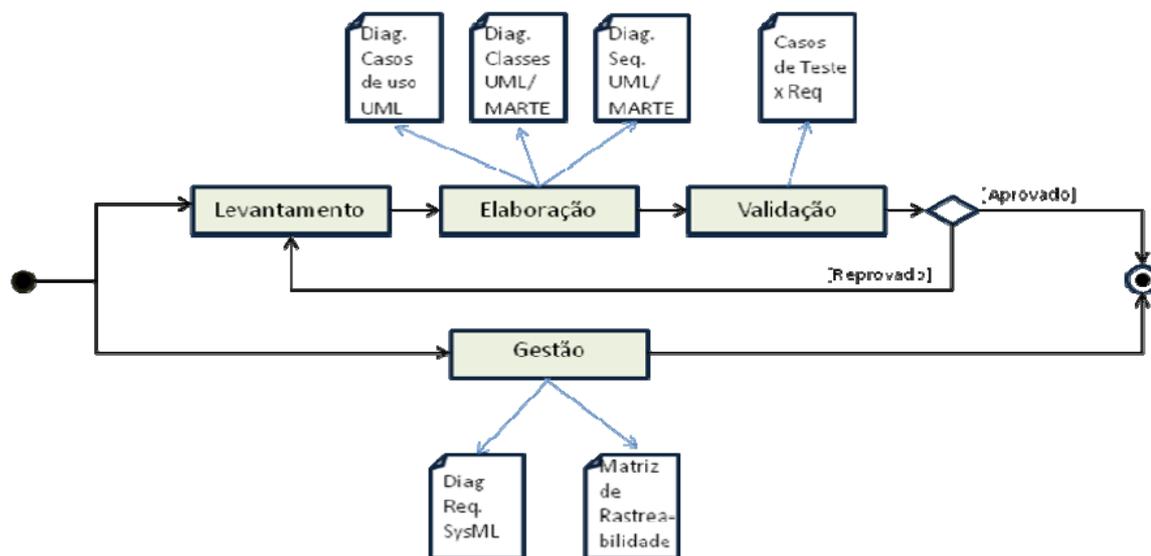


Figura 6.11 – Abordagem MDEReq

Fonte: MARQUES, SIEGERT e BRISOLARA (2013).

6.10.2 Atividades gerais do processo

A proposta deste trabalho é uma abordagem para a engenharia de requisitos de sistemas embarcados, a qual dá suporte às principais etapas da engenharia de requisitos (levantamento, elaboração, validação e gestão), por meio da construção de um modelo integrado que faz uso das linguagens UML, MARTE e SysML para evitar redefinições que não permitiriam o uso de ferramentas de modelagem já disponíveis.

Foi utilizada uma modelagem orientada a objetos para abstrair a complexidade dos sistemas embarcados. Além disso, trabalhou-se com rastreabilidade de requisitos para identificar inconsistências entre modelos e possibilitar uma visão mais completa do sistema para auxiliar a verificação e a validação.

A importância dos componentes de sistemas embarcados também pode ser facilmente identificada nos estudos dos trabalhos correlatos. Muitos deles são desenvolvidos buscando ferramentas de implementação e verificação ou desenvolvendo componentes com a detecção

de erros, entre outros recursos. Os requisitos de sistema também são uma linha de pesquisa bastante explorada, uma vez que constituem a base de todo o planejamento realizado, desde a concepção até o controle e a construção.

6.11 Tabelas de correlação

A Tabela 6.3 apresenta a relação dos trabalhos apresentados nas seções deste capítulo com as atividades esperadas para o desenvolvimento de sistemas embarcados, a saber:

- A coluna **Seq.** organiza sequencialmente as atividades observadas na tabela.
- A coluna **Atividades** descreve as fases e as atividades de desenvolvimento de sistemas que foram observadas e apresentadas na tabela, bem como o direcionamento do trabalho correlato.
- A coluna **Abordagem** apresenta o contexto das abordagens utilizadas em cada fase dos trabalhos correlatos:
 - **Diretrizes:** trabalhos que apresentam diretrizes para o desenvolvimento de sistemas embarcados.
 - **Especificação:** trabalhos que especificam modelo, métodos, ações e padrões para o desenvolvimento de sistemas embarcados.
 - **Ferramentas/técnicas:** trabalhos que indicam ferramentas e técnicas de desenvolvimento de sistemas embarcados.

Tabela 6.3 – Tabela de Correlação⁵

Seq	Atividades	Abordagem	*	a	b	c	d	e	f	g	h	i	j
1	Ciclo de Vida Definido	Diretrizes	x						x		x		
		Especificação	x		x				x				
		Ferramenta/Técnica							x				
2	Atividades Comuns	Diretrizes	x								x		
		Especificação	x	x				x					
		Ferramenta/Técnica											
3	Planejamento	Diretrizes	x	x		x				x	x		
		Especificação	x		x							x	
		Ferramenta/Técnica											x
4	Gerencia de Requisitos	Diretrizes	x	x		x	x			x	x		
		Especificação	x			x	x					x	x
		Ferramenta/Técnica			x	x	x						
5	Modelagem (Análise e Design)	Diretrizes	x	x						x	x		
		Especificação	x					x				x	
		Ferramenta/Técnica								x			x
6	Implementação	Diretrizes		x						x	x		
		Especificação	x										x
		Ferramenta/Técnica	x							x			x
7	Integração	Diretrizes	x							x	x		
		Especificação	x										
		Ferramenta/Técnica									x		
8	Verificação e Validação	Diretrizes	x	x	x					x	x		
		Especificação	x										
		Ferramenta/Técnica									x		
9	Avaliação do Processo	Diretrizes	x			x							
		Especificação	x										
		Ferramenta/Técnica											
10	Uso de Ferramenta Específica	Diretrizes	x				x						
		Especificação	x										
		Ferramenta/Técnica					x						x
11	Uso de Modelo de Qualidade	Diretrizes	x			x							
		Especificação	x			x							
		Ferramenta/Técnica											
12	Permite uso de técnicas	Diretrizes	x							x	x		
		Especificação	x		x	x							x
		Ferramenta/Técnica											
13	Reutilização de Componente	Diretrizes	x										
		Especificação	x				x	x					
		Ferramenta/Técnica											
14	Individualização entre HW e SW	Diretrizes	x	x						x			
		Especificação	x										
		Ferramenta/Técnica											
15	Foco somente no sistema	Diretrizes									x		
		Especificação			x	x	x	x	x	x		x	x
		Ferramenta/Técnica		x									

⁵ As colunas de a até j correspondem aos trabalhos correlatos, conforme referência indicada no início deste capítulo

Avaliação dos trabalhos apresentados

Avaliando a Tabela 6.3, que apresenta o direcionamento dos dez trabalhos correlatos apresentados nesta proposta, pode-se observar que eles buscam resolver problemas em atividades específicas da engenharia de *software*.

- **Ciclo de vida:** foi abordado em três trabalhos, e em apenas um deles apontou especificação, diretrizes e ferramentas.
- **Atividades comuns:** foram abordadas em dois trabalhos, com propostas únicas para diretrizes e especificação.
- **Planejamento:** foi abordado em seis dos trabalhos, tendo maior ênfase na definição de diretrizes e especificações das atividades.
- **Gerência de requisitos:** foi abordada em sete trabalhos; dois deles apresentaram propostas para diretrizes, especificação e uso de ferramentas.
- **Modelagem (análise e design):** foi abordada em seis trabalhos, com foco maior na especificação do modelo.
- **Implementação:** foi abordada em quatro trabalhos, com maior foco na utilização de ferramentas.
- **Integração:** foi abordada em dois trabalhos, com foco nas diretrizes a serem seguidas.
- **Verificação e validação:** foram abordadas em quatro trabalhos aqui apresentados e, de forma geral, apresentaram as diretrizes, a especificação e as ferramentas utilizadas.
- **Avaliação do processo:** foi abordada em um trabalho que apresentou diretrizes para as atividades de avaliação do processo e busca de melhoria contínua.
- **Uso de ferramenta específica:** foi abordado em dois dos trabalhos apresentados, que direcionaram suas pesquisas para a criação e o uso de ferramentas para o processo.

- **Uso de modelo de qualidade:** foi abordado em um dos trabalhos, que incluiu em seus estudos diretrizes e especificações para o uso de um modelo de qualidade.
- **Uso de técnicas:** foi abordada em cinco trabalhos que apresentaram técnicas específicas para serem utilizadas, com maior foco na especificação das técnicas.
- **Reutilização de componente:** foi abordada por dois trabalhos, focando na especificação dos componentes para a reutilização.
- **Individualização entre HW e SW:** foi abordada por dois trabalhos como parte do contexto de seus estudos sobre o tratamento individualizado das partes *software* e *hardware* do componente embarcado.
- **Foco somente no sistema:** foi abordado nos dez trabalhos apresentados, que utilizaram o conceito de *sistema* em suas abordagens.

Pode-se observar que os trabalhos concentram seus estudos em pontos específicos do desenvolvimento de sistemas embarcados, buscando melhorar a qualidade em determinados pontos.

A maioria dos trabalhos apresentados direciona os recursos para o desenvolvimento do sistema embarcado e apenas dois especificam a parte *hardware* e a parte *software* individualmente.

As atividades que receberam maior atenção nos trabalhos apontados foram as áreas de Verificação e Validação, Gerência de Requisitos e Modelagem, respectivamente, em ordem de maior para menor abordagem.

7 ARQUITETURA DO PROCESSO

O sistema embarcado tem características específicas e deve ser desenvolvido utilizando processos de *software* específicos, seguindo um padrão bem estabelecido e consolidado, como acontece nas outras engenharias.

Na literatura, as referências a sistemas embarcados são vastas, mas poucas direcionam seus estudos ou atividades observando de forma independente os componentes exclusivamente de *software*. Por essas razões, faz-se necessário o emprego de uma abordagem de desenvolvimento de *software* que leve em consideração as características e peculiaridades desse contexto.

Com o aumento considerável na quantidade de memória disponível e na capacidade dos processadores utilizados por produtos eletrônicos (KUCINSKIS, 2011), será possível o avanço no uso e na autonomia dos *softwares* embarcados, o que já acontece em missões espaciais, exigindo o desenvolvimento de práticas e atividades que observem separadamente os requisitos das partes *software* e *hardware* de um componente de sistema embarcado.

Atualmente, a maioria dos produtos eletrônicos tem algum componente computacional. Em geral, tal recurso é utilizado com o objetivo de diminuir custos e/ou torná-lo flexível. Essa utilização ocorre em todas as categorias de produto, especialmente naqueles considerados críticos. O aumento do uso desses sistemas, sua diversidade e o número de funções que estão sendo incorporadas a um único sistema embarcado multiplica seu grau de criticidade (WEHRMEISTER; BECKER; PEREIRA, 2007). Um produto de *software* é considerado crítico quando é capaz de levar o produto em que está instalado a uma situação de risco, isto é, quando pode provocar acidentes graves (PÁSCOA, 2002).

Em geral, os problemas não estão no *software* em si, mas na forma como eles são desenvolvidos. Por isso, é preciso aplicar na indústria de *software*, mais efetivamente e com a mesma disciplina os conceitos de qualidade aplicada em outras engenharias, procedimentos, modelos e ciclos de vida independentes.

Quando a engenharia de *software* passou a tratar as preocupações e problemas relativos à qualidade levando em consideração as peculiaridades do *software*, foi necessária a criação de um guia para a aplicação da norma ISO 9001 à produção de *software* e, conseqüentemente, surgiram modelos específicos. É possível destacar os esforços da *International Organization for Standardization (ISO)* com o modelo *Software Process Improvement and Capability dEtermination (SPICE)*, o modelo Modelo de Maturidade de Processo de *Software (CMM/SEI)* da *Universidade de Carnegie Mellon*, o modelo Trillium, desenvolvido pela Bell Canadá para sistemas de telecomunicações, e o *Goal Quality Metric (GQM)*, desenvolvido em um projeto conjunto com um laboratório da NASA (MARWEDEL, 2003). Atualmente, foi desenvolvido o modelo *Capability Maturity Model Integration (CMMI)*.

Com a finalidade de construir *softwares* dentro do prazo estimado, com custo adequado e atendendo às expectativas de funcionamento do solicitante, foram desenvolvidas algumas normas e modelos, dentre eles o CMMI. Existem, porém, grandes dificuldades relacionadas ao uso de tais modelos e seus respectivos produtos pelas organizações, e isso acontece porque as organizações e seus desenvolvedores interpretam esse uso como documentação a ser produzida, o que, em caso de atraso ou diminuição de prazos e custos, é a primeira atividade a ser negligenciada, inviabilizando seu uso e refletindo diretamente na qualidade de seu resultado. Isso acontece principalmente no desenvolvimento de pequenos produtos e/ou em organizações pequenas.

Estudando os modelos de qualidade, podemos observar que a utilização de sua estrutura e organização, de forma adaptada para a característica do produto a ser desenvolvido, pode contribuir para a qualidade do produto a ser disponibilizado no mercado.

O ciclo de vida proposto foi idealizado segundo a estrutura do TSP, em que, para cada ciclo de produção, devem ser realizadas as seguintes ações:

- **Lançamento (declaração das necessidades):** estabelece as metas e os objetivos do desenvolvimento, as responsabilidades.
- **Estratégia:** define a ordem em que as funções do produto serão definidas, desenhadas, implementadas e testadas, além dos padrões e dos documentos que serão produzidos.

- **Planejamento:** lista os produtos a serem desenvolvidos, a estimativa, o plano de qualidade, a lista de tarefas, entre outros.
- **Definição do ambiente total:** envolve o estudo e o entendimento da solicitação, a engenharia e o projeto do produto, e a parametrização do ambiente de desenvolvimento.
- **Fases:** detalhamento das estratégias gerais a serem desenvolvidas em cada fase de desenvolvimento (requisitos, design, implementação e teste).
- **Post mortem:** análise das atividades desenvolvidas no ciclo e propostas de mudanças no processo para o próximo projeto ou ciclo.

7.1 Processo proposto

O projeto de um sistema embarcado deve seguir um ciclo de vida especializado, uma vez que um sistema embarcado desenvolve atividades especializadas.

É real a necessidade do aumento da qualidade dos componentes de sistemas embarcados. Foram estudados os modelos de qualidade, em especial o CMMI-DEV. Observou-se que a aplicação de suas práticas na elaboração de um processo que atenda às características do produto de sistema embarcado pode contribuir para o aumento da qualidade do produto a ser concebido. Para isso, tais práticas devem ser aplicadas de forma adequada para garantir a evolução de seu desenvolvimento, pois as atividades requeridas pelo nível devem ser definidas e cumpridas.

Neste trabalho foi apresentada a elaboração de um processo de desenvolvimento para componentes de sistemas embarcados que atenda a cada área de processo do CMMI-DEV, de acordo com as características e as necessidades do componente a ser desenvolvido, buscando proporcionar ao processo determinadas capacidades, sem se preocupar com a conquista de maturidade no modelo.

As fases e as atividades do processo proposto atende a todas as áreas de processo do nível 2 de maturidade. Isso ocorre porque as atividades requeridas por essa fase são essenciais para um processo de desenvolvimento, tendo como objetivo básico proporcionar o gerenciamento dos

requisitos e do projeto para que eles possam ser planejados, executados, medidos e controlados, e as práticas utilizadas possam ser mantidas mesmo em momentos de estresse.

Conforme apresentado no Capítulo 3, o modelo CMMI-DEV tem duas representações – contínua e por estágio – e o processo proposto pode ser implementado utilizando ambas, uma vez que a de forma estagiada cobre por completo o nível 2 e a de forma contínua atende a grande parte das áreas de processo dos níveis 3, 4 e 5.

A garantia de que o processo proposto pode conquistar o nível 2 do CMMI-DEV não se deu por escolha, e sim por consequência. A partir do estudo e da análise das exigências de cobertura de cada nível, relacionando-as com as características do processo proposto para o desenvolvimento de componentes para sistemas embarcados, percebeu-se que tais necessidades são inteiramente requeridas no nível 2 do CMMI-DEV.

Os componentes embarcados, em geral, têm características exclusivas que necessitam de controles específicos, portanto, um processo para atendê-los deve permitir a adaptação das atividades a serem realizadas no desenvolvimento de cada componente, o que torna tal nível completamente pertinente.

O nível 2 do CMMI-DEV, representado na Figura 7.1, é composto por um conjunto de caixas pretas, cada qual representando as fases de seu ciclo de vida e suas atividades internas. Para isso deve:

- Oferecer infraestrutura adequada para apoiar o processo;
- Ser planejado e executado de acordo com uma política;
- Oferecer recursos adequados para produzir saídas controladas;
- Ser monitorado, controlado e revisado;
- Avaliar sua aderência em relação à descrição do processo;
- Não exigir a especificação das atividades de cada fase.

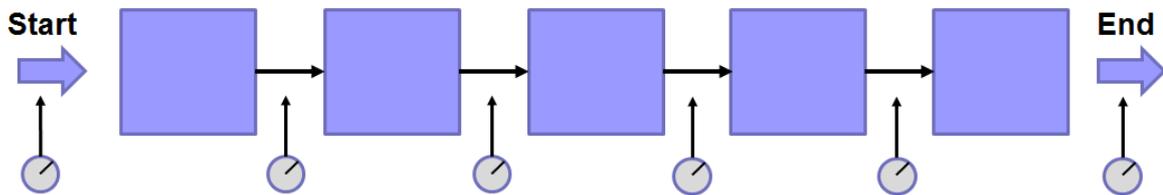


Figura 7.1 – Nível 2 CMMI gerenciado

Fonte: adaptada COUTO (2007).

A adoção do nível 2 do CMMI-DEV para o processo de desenvolvimento é adequado, uma vez que as atividades internas de cada fase não exigem definição formalizada, o que flexibiliza a adaptação dessas atividades de acordo com a característica do componente a ser desenvolvido.

O nível 3 de maturidade é caracterizado como um processo definido, pois, a partir do conjunto de fases determinado no nível 2, ele é descrito de forma rigorosa, estabelecendo claramente os objetivos e os critérios para cumpri-lo. Esse nível de maturidade exige especificação de todas as entradas, atividades, papéis, medidas, etapas de verificação e saídas; então, deve ser realizado segundo essa especificação, com o mínimo de ajustes ou adaptações para não descaracterizar o processo definido.

Para o processo proposto, a adoção desse nível inflexibiliza a adaptação de acordo com as características do componente em desenvolvimento, pois no nível 3, após sua especificação, essas ações devem ser rigorosamente cumpridas, com flexibilidade e adaptações mínimas, já que uma grande adaptação em um processo com esse nível de maturidade em geral descaracteriza seus objetivos e metas e a forma de cumpri-los.

O processo para o desenvolvimento de componentes para sistemas embarcados deve incluir atividades, recursos e saídas que deverão auxiliar na identificação, elaboração e execução de modelos para validação de comportamentos do componente antes de sua implementação em fases iniciais, eliminando erros de concepção e minimizando possíveis problemas, prejuízos ou catástrofes.

Apesar de o nível 2 do CMMI-DEV não requerer especificação detalhada das atividades de cada fase nem exigir a definição de artefatos de saída, esse processo oferece atividades e

documentos a serem realizados, os quais podem e devem ser adaptados de acordo com as características e os requisitos do componente a ser desenvolvido, sem deixar de atender os requisitos de maturidade do processo. Dessa forma, o processo vai atender às exigências do nível 2 do CMMI-DEV na totalidade e nas partes das exigências do nível 3.

Neste estudo foi elaborado um processo cíclico organizado em fases, com atividades independentes e compartilhadas entre o *software* e o *hardware* do componente embarcado. A realização de cada ciclo completo oferece como resultado um componente embarcado, conforme apresentado no ciclo de vida na Figura 7.2.

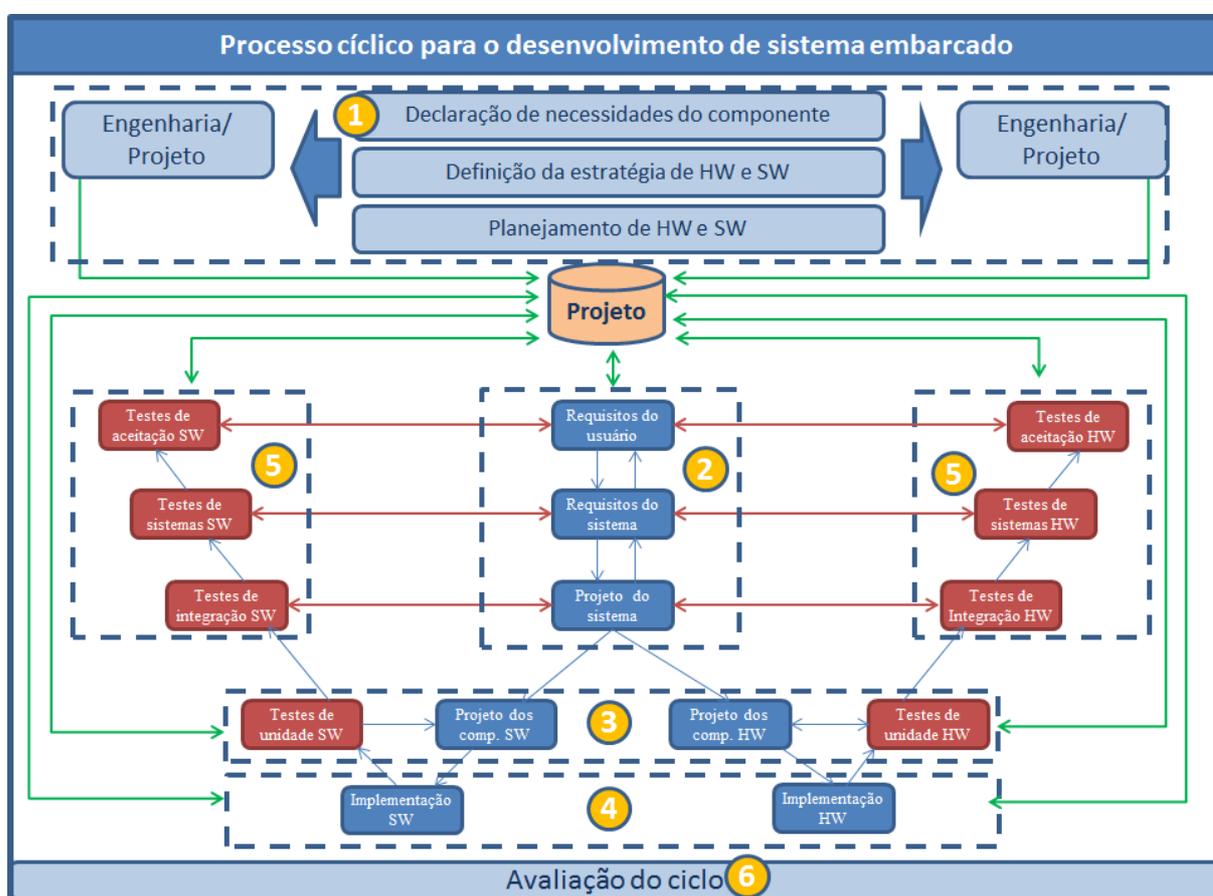


Figura 7.2 – Processo proposto para sistemas embarcados.

O processo é denominado *cíclico*, pois cada componente do sistema embarcado a ser desenvolvido segundo suas especificações deve realizar todas as suas fases e atividades, formando assim um ciclo de desenvolvimento de componentes de sistema embarcado, conforme apresentado na figura 7.3.

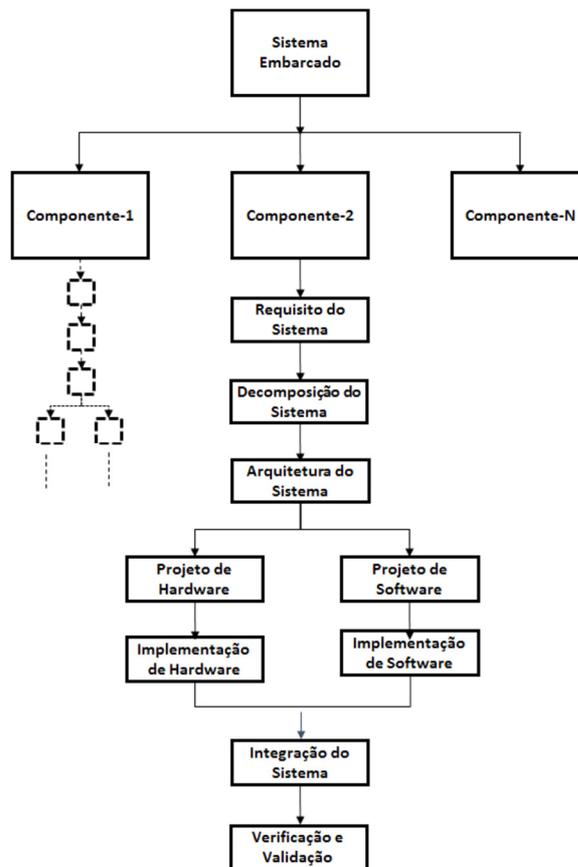


Figura 7.3 – Funcionamento do Processo Cíclico.

Nessa abordagem está sendo apresentado um processo para o desenvolvimento de sistemas embarcados com foco na gestão do projeto e nas boas práticas de desenvolvimento de sistema, passando por um ciclo de desenvolvimento completo, isto é, todas as fases do ciclo de vida apresentadas na Figura 7.2. Vale lembrar que se entende por *componente embarcado* um conjunto composto por *hardware* e *software* que, juntos, vão realizar uma atividade.

7.1.1 Ciclo de desenvolvimento

Ciclo de desenvolvimento é a realização sequencial de todas as fases e atividades do ciclo de vida. Cada ciclo de desenvolvimento terá como resultado um componente de um sistema embarcado e todos os documentos produzidos durante seu desenvolvimento. Quando um sistema embarcado for composto por diversos componentes, deverá ter diferentes ciclos de desenvolvimento, sendo um para cada componente de sistemas embarcado.

Cada ciclo deve estabelecer as relações de trabalho, definir e distribuir os papéis aos membros da equipe, bem como definir metas, estratégias e planos de trabalho para todas as fases do processo. Quando oportuno, deve realizar modificações no processo, de acordo com o resultado da avaliação do ciclo.

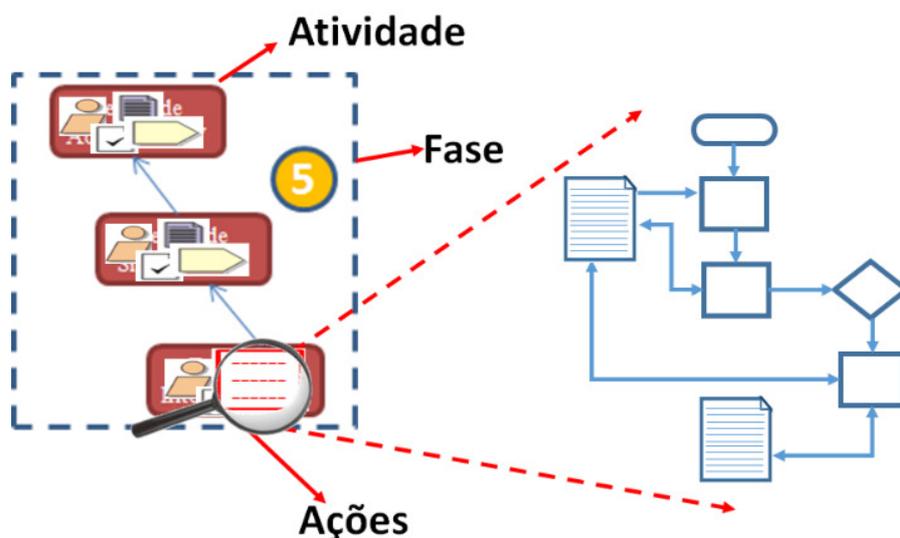


Figura 7.4 – Componentes do processo cíclico.

Na estrutura apresentada na Figura 7.4, as fases são o agrupamento de atividades no ciclo de vida; as atividades são procedimentos que devem ser realizados em cada fase; e as ações são detalhamentos da realização de cada atividade especificando a sequência a ser seguida, os desvios quando pertinentes, as saídas esperadas e os artefatos produzidos.

O processo é organizado em fases, compostas por atividades e realizadas por meio de ações governadas por procedimentos e documentos (modelos) que resultam em artefatos comuns, os quais compõem o produto de sistema embarcado (programas e sua documentação). Todas as fases e atividades são identificadas em todos os fluxos pela numeração sequencial utilizada na descrição das atividades em suas respectivas fases e referenciadas em fluxos (ver Apêndice B).

7.1.2 Atividades comuns (FC): são as atividades de controle realizadas em todas as fases do processo, sempre iniciadas na fase de engenharia. O item 1 da Figura 7.2 apresenta seu posicionamento na estrutura do processo. As atividades comuns nascem

na fase de engenharia e evoluem nas demais fases, e sua representação gráfica no ciclo de vida do processo está atrelada à atividade de engenharia.

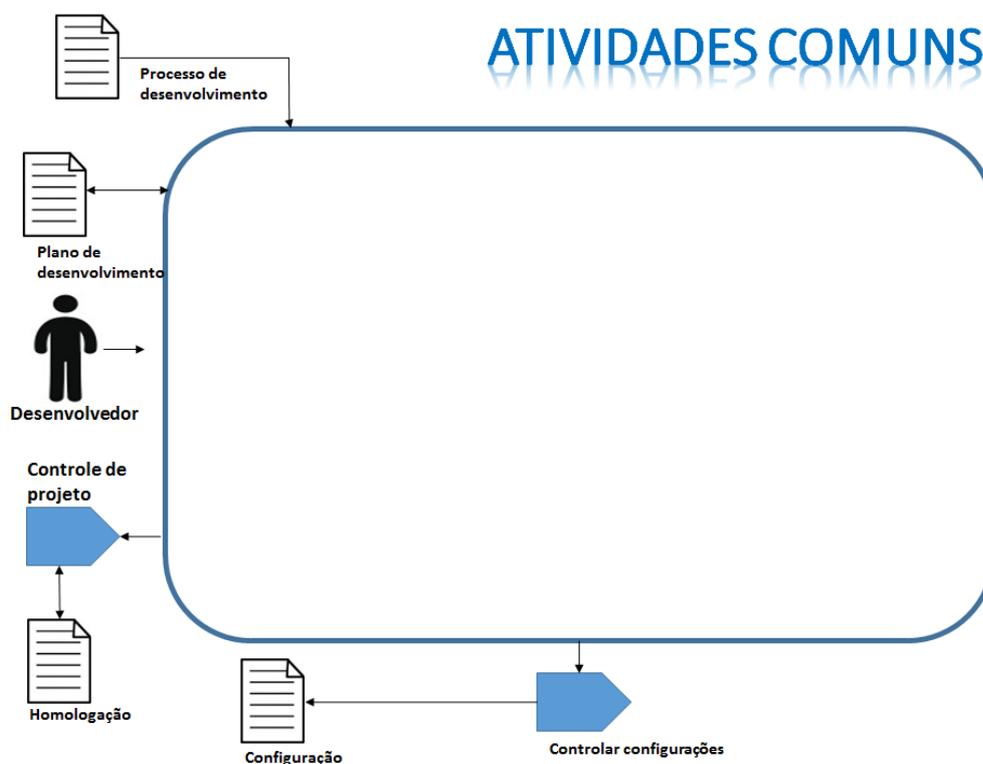


Figura 7.5 – Atividades comuns – Fase.

As atividades comuns ao processo são aquelas que devem ser realizadas em todas as fases do desenvolvimento dos componentes. Consistem em atividades de controle do projeto e de qualidade, devendo se iniciar no mesmo momento em que nasce o projeto. Tais atividades devem ser descritas de maneira formal e cumpridas de acordo com o que foi definido, conforme mostrado na Figura 7.5.

- **Controle do projeto – 1:** nesta atividade devem-se realizar ações para o monitoramento do projeto, o acompanhamento das ações realizadas, a avaliação do desempenho do projeto de acordo com o planejado, a realização de mudanças necessárias de acordo com o contexto do projeto.

- **Controlar configurações – 2:** nesta atividade deve-se definir a estrutura e a configuração necessárias para o desenvolvimento do projeto, identificar e descrever todos os itens que devem ser controlados, todas as mudanças que os artefatos produzidos podem sofrer, bem como todos os responsáveis por elas.

A Figura 7.6 apresenta o posicionamento individualizado desta fase no ciclo de desenvolvimento.

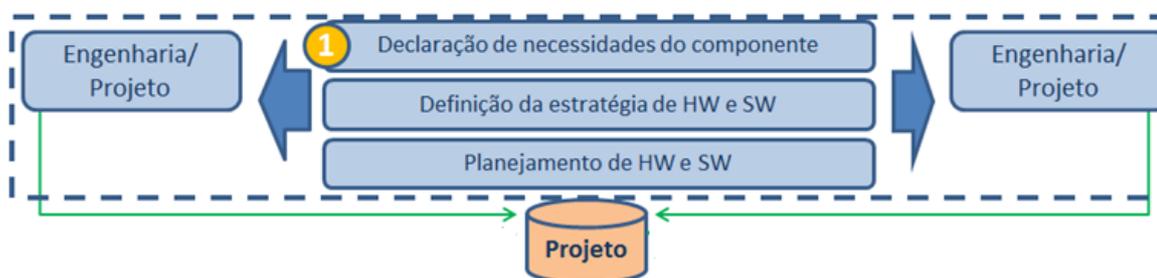


Figura 7.6 – Engenharia – Ciclo.

7.1.3 **Engenharia (F1):** realização de atividades para o entendimento do produto completo, o estudo e o planejamento do projeto, a viabilidade e a preparação do ambiente. O item 1 da Figura 7.2 e individualizado na Figura 7.6 apresenta seu posicionamento na estrutura do processo.

Na fase de engenharia, o componente é descrito de forma suficiente para permitir o planejamento de todas as atividades que serão realizadas. É definido o ambiente a ser utilizado para todo o desenvolvimento, além dos mecanismos para medição e controle do projeto (Figura 7.7).

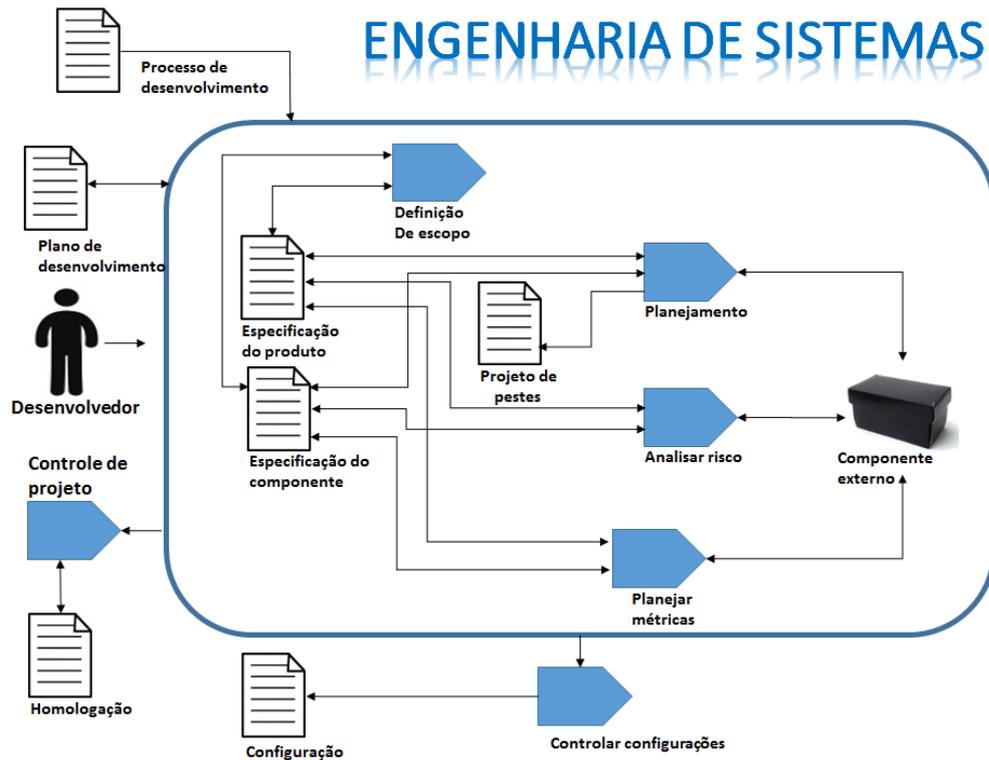


Figura 7.7 – Engenharia – Fase.

- **Definição de escopo – 3:** atividade realizada para delimitar a fronteira das funcionalidades a serem realizadas pelos componentes e sua inserção no contexto. Podem ser apresentadas também as funcionalidades do produto completo (sistema). O documento de especificação do produto é opcional, uma vez que o processo é aplicado a cada componente.
- **Planejamento – 4:** atividade de adaptação, preparação, organização e estruturação do processo para o desenvolvimento do componente, de acordo com as diretrizes do processo. Oferece parâmetros para acompanhamento, medição e tomadas de decisão.
- **Analisar risco – 5:** verificação dos pontos críticos que possam vir a apresentar imperfeição ou falha durante a realização do projeto, a qual pode comprometer o desenvolvimento do componente. É o estudo de cada fase de desenvolvimento feito para identificar possíveis riscos e evitá-los. Entre suas ações estão incluídos: identificação, análise, classificação, planejamento, monitoramento, rastreamento e controle de riscos.

- **Planejar métricas – 6:** atividade realizada para a identificação e a compreensão das medidas que devem ser contempladas como mecanismos de controle e avaliação. Nela devem ser definidas as métricas de qualidade do processo de desenvolvimento do componente, porém de forma especial, para os componentes de sistemas embarcados. Deve ser planejada com muita precisão a medição da qualidade do produto a ser desenvolvido. Nesta atividade devem-se identificar todos os requisitos do componente a ser desenvolvido e medido ao longo de sua construção.

7.1.4 **Análise de requisitos (F2):** detalhamento dos requisitos do produto e dos componentes a serem desenvolvidos. O item 2 da Figura 7.2 apresenta seu posicionamento na estrutura do processo.

Na fase de requisitos o componente é descrito de maneira completa e com especificações de regras detalhadas, em que se devem destacar, de forma clara, os requisitos do projeto, suas restrições, entre outras informações que não podem ser omitidas, pois estabelecem critérios importantes durante todas as etapas futuras (Figura 7.8).

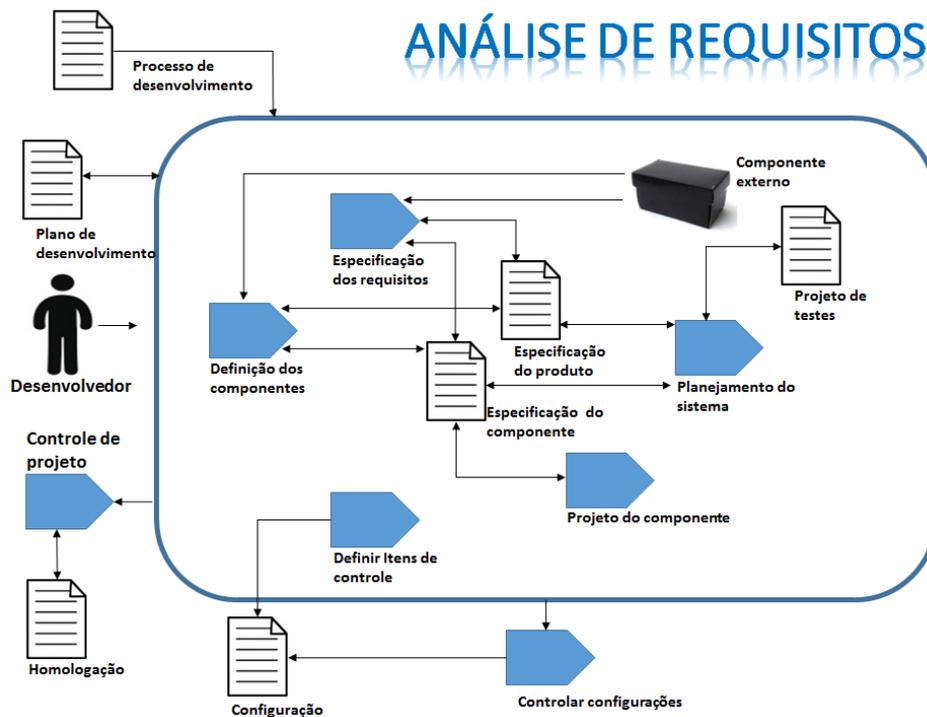


Figura 7.8 – Requisitos – Fase.

- **Especificação dos requisitos – 7:** definição detalhada dos requisitos do produto a ser construído.
- **Definição dos componentes – 8:** detalhamento do componente que será desenvolvido.
- **Planejamento do sistema – 9:** identificação, individualização e especificação das tarefas atribuídas ao *software* e ao *hardware* do componente a ser construído.
- **Projeto do componente – 10:** elaboração do projeto do componente (*hardware* e *software*) a ser construído.
- **Definir itens de controle – 11:** de acordo com os requisitos, devem-se identificar todos os itens do produto/componente e itens de estrutura do ambiente que deverão ser controlados pela gerência da configuração, além de indicar a forma de controle e as ferramentas a serem utilizadas.

A Figura 7.9 apresenta o posicionamento individualizado desta fase no ciclo de desenvolvimento.



Figura 7.9 – Requisitos do produto – ciclo.

7.1.5 Análise e projeto (F3): definição dos componentes e divisão das funcionalidades a serem implementadas nas partes de hardware e de software do componente, estudo de reutilização de componentes ou utilização de componentes externos, elaboração e

validação de modelos e protótipos. O item 3 da Figura 7.2 apresenta seu posicionamento na estrutura do processo.

É nessa fase que se identificam as funcionalidades do sistema que serão implementadas em *hardware* e as que serão implementadas em *software*. Nela também se avalia a utilização de componentes externos ao projeto (Figura 7.10).

Ainda nessa fase são produzidos os modelos individuais de *hardware* e de *software*. É importante o uso de técnicas e ferramentas para prototipação e simulação de arquitetura de *software* e de *hardware*, pois, por meio dos modelos produzidos, poderão ser estudados os resultados obtidos. Se necessário, pode-se rever o critério de separação das funcionalidades, realizando simulações até que se produza um modelo que atenda aos requisitos estabelecidos.

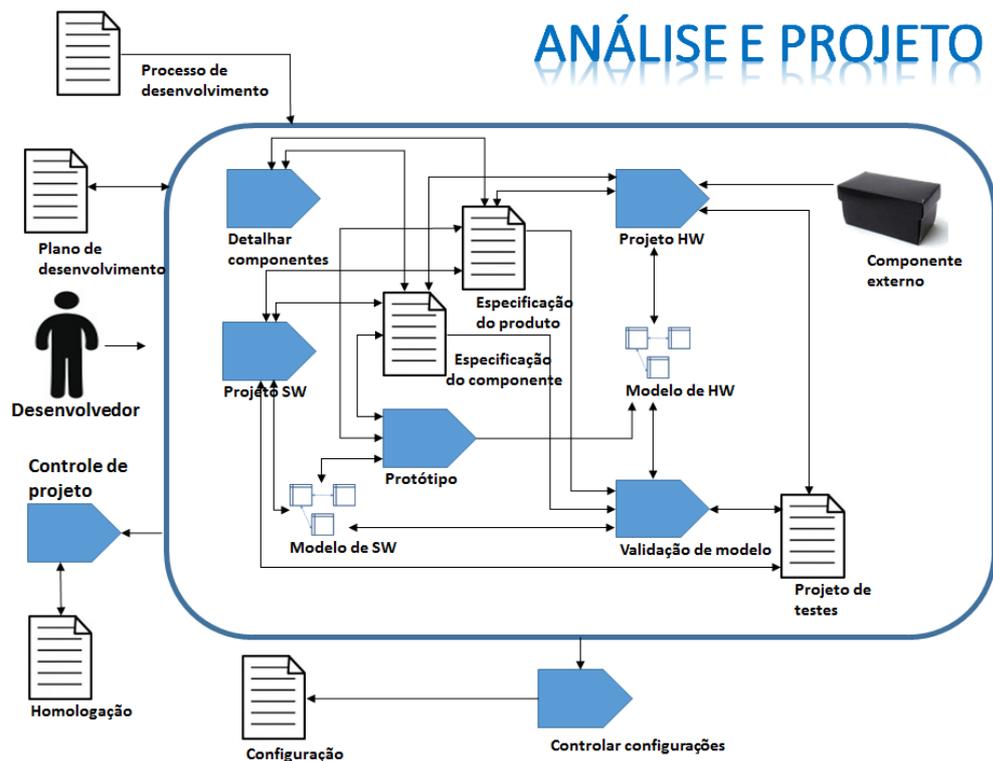


Figura 7.10 – Análise e projeto – fase.

- **Detalhar componentes – 12:** detalhamento dos itens que serão utilizados pelos

componentes de *hardware* e de *software* de acordo com a individualização das tarefas. Esta atividade vai alimentar os documentos de especificação do produto e definição dos componentes.

- **Projeto HW – 13:** desenho da arquitetura do *hardware* para a realização da prototipação e posterior construção de acordo com a arquitetura do *software*. Nesta atividade são feitas a análise e a definição das ferramentas de modelagem e produto e definição dos componentes.
- **Projeto SW – 14:** desenho da arquitetura do *software* para a realização da prototipação e posterior construção de acordo com a arquitetura do *hardware*. Nesta atividade são feitas a análise e a definição das ferramentas de modelagem e prototipação a serem utilizadas. Ela vai alimentar os documentos de especificação do produto e definição dos componentes.
- **Protótipo – 15:** construção do protótipo dos modelos de *hardware* e de *software* elaborados a serem validados.
- **Validação de modelo – 16:** validação dos modelos e protótipos construídos em atendimento aos requisitos de acordo com os planos.

A Figura 7.11 apresenta o posicionamento individualizado desta fase no ciclo de desenvolvimento.



Figura 7.11 – Análise e projeto – ciclo.

7.1.6 Implementação e integração (F4): nesta fase são realizadas atividades independentes de implementação do hardware e do software do componente. Em seguida, conclui-se sua integração e são realizados testes independentes e testes integrados. O item 4 da Figura 7.2 apresenta seu posicionamento na estrutura do processo.

A fase de implementação deve ser realizada especialmente quando já se atingiu(ram) o(s) modelo(s) ideal(is) que atenda(m) aos requisitos do componente. Nesta fase cada um dos componentes (*hardware* e *software*) deve ser produzido segundo os modelos individuais que foram validados nas fases de modelagem e design. Também nesse momento devem ocorrer testes preliminares das partes do componente. Após a validação das partes dos componentes, eles devem ser integrados para formar o componente especificado e validado (Figuras 7.12 e 7.13).

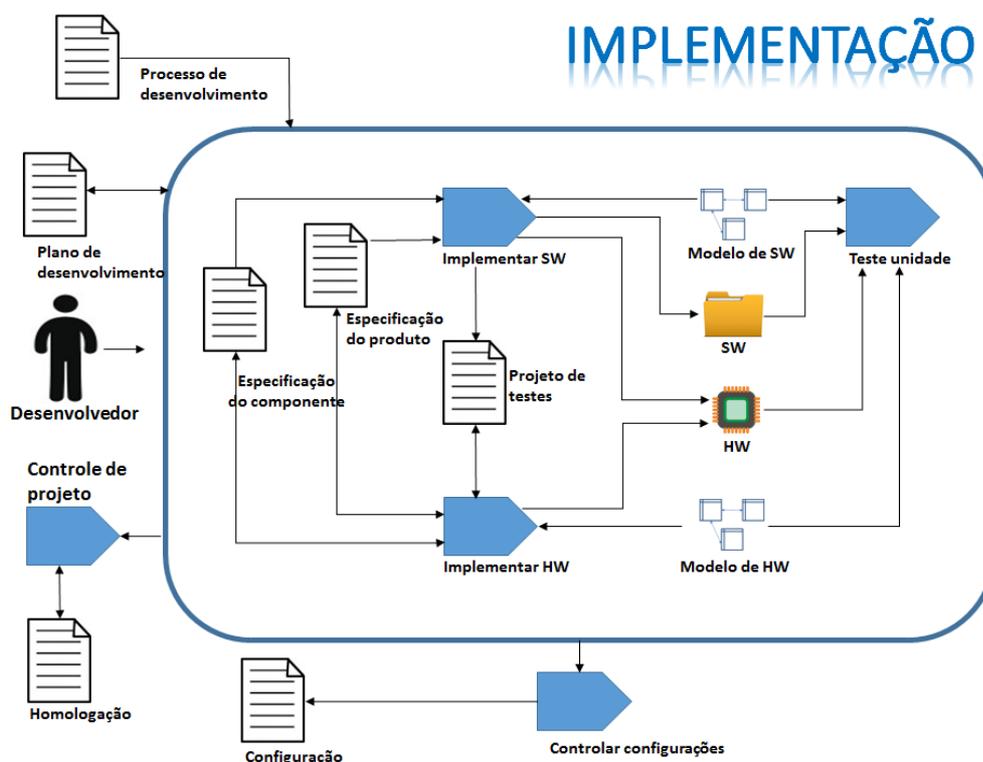


Figura 7.12 – Implementação – fase.

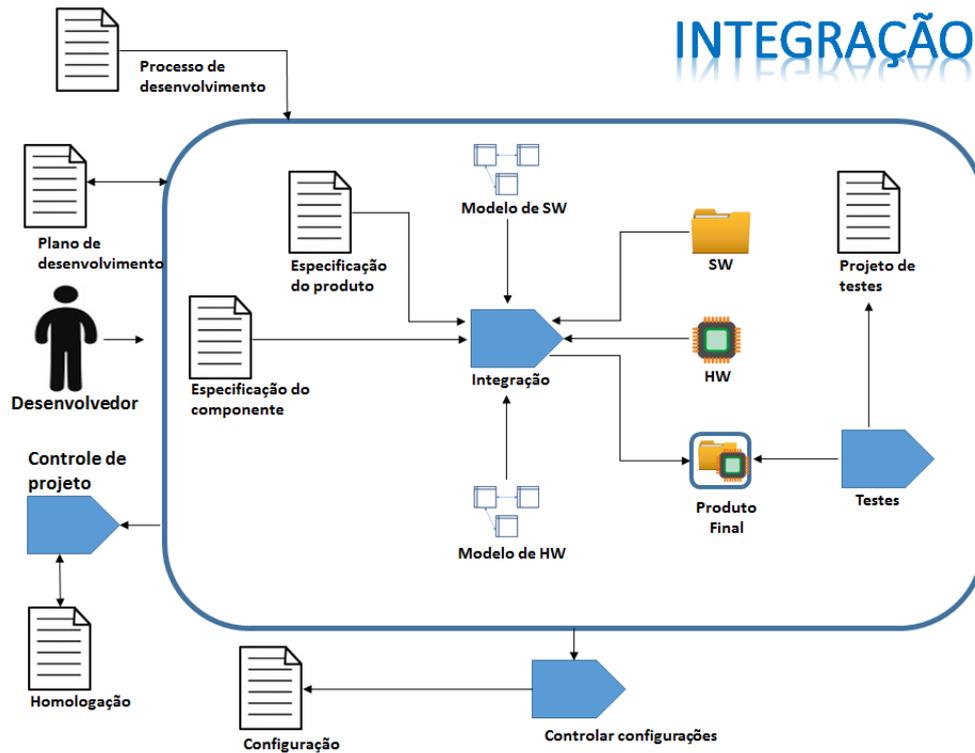


Figura 7.13 – Integração – fase.

- **Implementar SW – 17:** construção do *software* de acordo com a especificação e os modelos validados.
- **Implementar HW – 18:** produção ou aquisição do *hardware* que atenda às especificações validadas.
- **Teste de unidade – 19:** atividades de teste e conferência de cada parte do componente (*hardware* e *software*) de acordo com o plano de testes.
- **Integração-SW-HW – 20:** integração (conexão) das funcionalidades da parte *hardware* e da parte *software* do componente.
- **Teste – 21:** atividades de teste e conferência de cada parte do componente e do componente integrado, bem como simulação do seu ambiente de uso de acordo com o plano de testes.

A Figura 7.14 apresenta o posicionamento individualizado desta fase no ciclo de desenvolvimento.



Figura 7.14 – Implementação e integração – ciclo.

7.1.7 **Verificação e validação do sistema (F5):** nesta fase executam-se os testes de verificação e validação do componente, bem como a conferência, o acompanhamento e a homologação. A Figura 7.2, o item 5, apresenta seu posicionamento na estrutura do processo.

A verificação e validação devem buscar de forma exaustiva e completa todas as condições ao qual o componente será submetido, seguindo os planos de teste que foram produzidos ao longo do desenvolvimento. Devem ser validados os comportamentos do componente integrado, das partes do componente e do seu comportamento inserido no sistema ou no ambiente em que ele vai funcionar. É importante o uso de ferramentas e técnicas de testes, bem como de simulações (Figura 7.15).

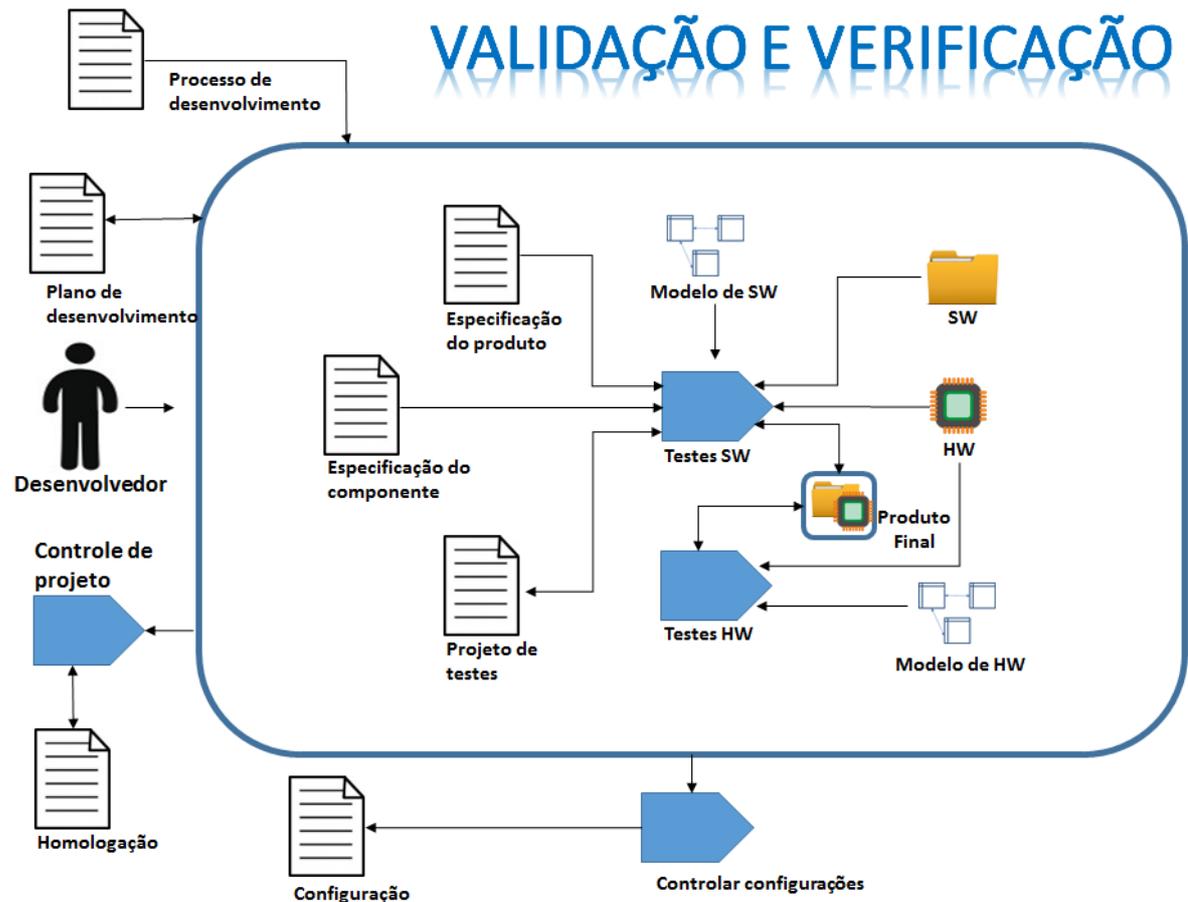


Figura 7.15 – Validação e verificação – Fase.

- **Testes SW – 22:** testes de verificação e validação do *software* do componente integrado e da sua integração e atendimento aos requisitos do *hardware*, de acordo com o plano de testes.
- **Testes HW – 23:** testes de verificação e validação do *hardware* do componente integrado e da sua integração e atendimento aos requisitos do *software*, de acordo com o plano de testes.

A Figura 7.16 apresenta o posicionamento individualizado desta fase no ciclo de desenvolvimento.

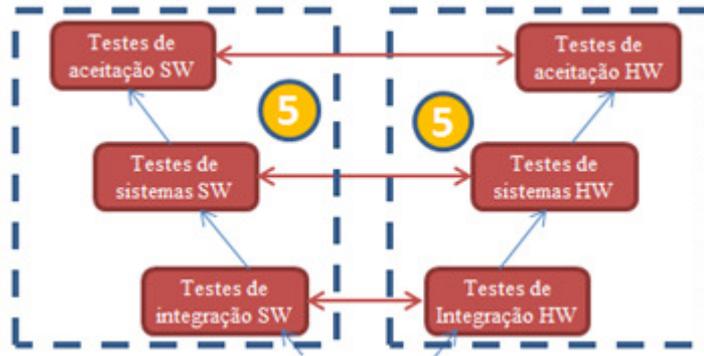


Figura 7.16 – Verificação e validação – ciclo.

7.1.8 **Avaliação do ciclo (F6):** análise das atividades desenvolvidas no ciclo, ajustes e adaptação para o próximo ciclo (item 6 da Figura 7.2).

Na fase de avaliação do ciclo deve-se congelar e integrar (reunir) todos os artefatos produzidos durante o processo de desenvolvimento, elaborando um livro (pasta) único com todos os documentos do projeto, além de documentos externos ao projeto que sejam relevantes. O produto a ser entregue incluirá não apenas o componente produzido, mas também toda a documentação resultante de sua produção. Deve-se realizar ainda um documento de avaliação do ciclo de desenvolvimento no qual sejam apresentadas todas as impressões deixadas no andamento do projeto, com possíveis sugestões de soluções (Figuras 7.17).

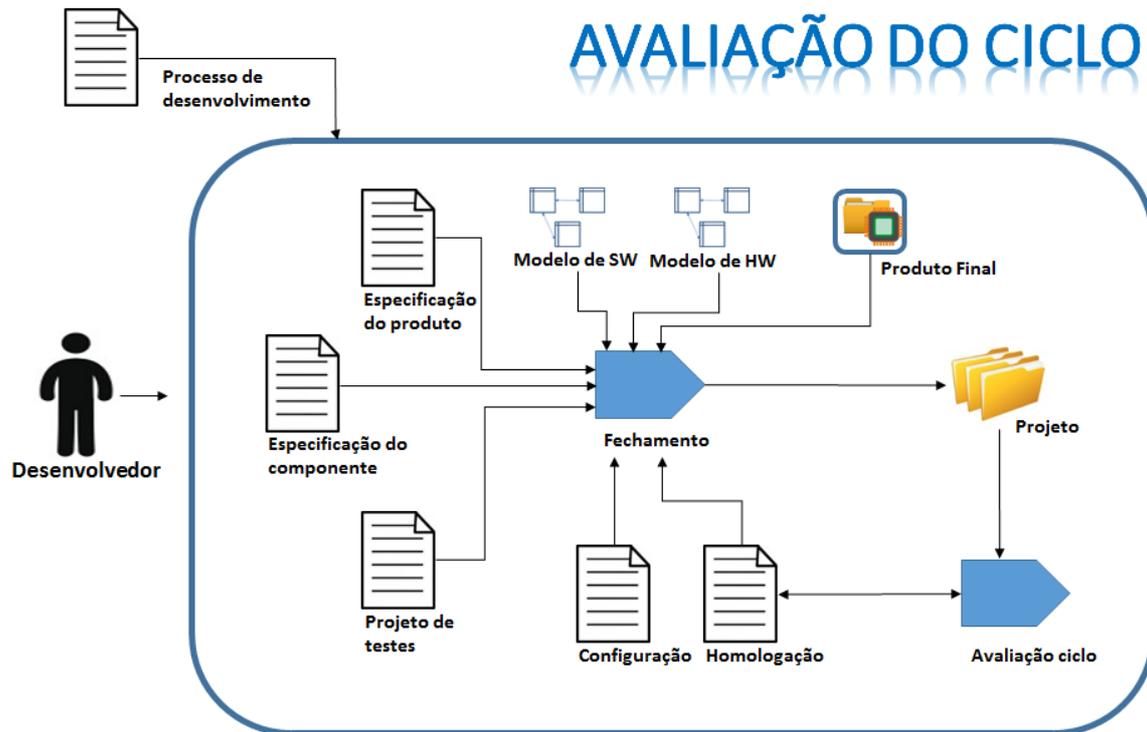


Figura 7.17 – Avaliação do ciclo – fase.

- **Fechamento – 24:** conclusão e fechamento de todas as atividades do ciclo de desenvolvimento do componente, bem como a elaboração do livro do projeto.
- **Avaliação do ciclo – 25:** análise e avaliação de todas as atividades realizadas durante o ciclo, produtividade e qualidade do processo e do produto, e sugestões de adaptações e melhoria para os próximos ciclos.

A Figura 7.18 apresenta o posicionamento individualizado desta fase no ciclo de desenvolvimento.

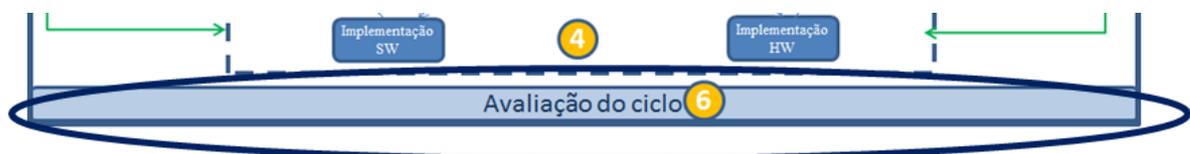


Figura 7.18 – Avaliação do ciclo – ciclo.

7.1.9 Descrição dos documentos a serem produzidos durante o processo.

- **Processo de desenvolvimento:** documento contendo um conjunto de atividades, organizadas e formalizadas que produzam saídas (artefatos) predefinidas com o objetivo de auxiliar na construção de componentes com qualidade prevista (ver Apêndice B).
- **Homologação:** documento resultante das atividades de acompanhamento e validação de cada fase do projeto, que registra a checagem de todos os artefatos produzidos em cada fase, de acordo com as especificações e as saídas esperadas, bem como o comprometimento e os resultados dos envolvidos. Esse documento é produzido na fase de engenharia e alimentado nas demais fases, podendo ser desconsiderado quando suas informações forem inseridas no plano de desenvolvimento. Esse documento é adaptado do artefato do rup rup_pacpln (ver Apêndice G).
- **Configuração:** documento resultante do planejamento, do controle e dos registros dos itens de configuração, seja para o ambiente de desenvolvimento e uso, linguagens, seja para todos os artefatos produzidos durante o processo, incluindo o próprio documento de configuração. Esse documento é produzido na fase de engenharia e alimentado nas demais fases, de acordo com as mudanças. Esse documento é adaptado do artefato do rup rup_cmpln (ver Apêndice H).
- **Plano de desenvolvimento:** documento contendo o resultado do conjunto de atividades organizadas e formalizadas com atribuição de sequências de tempo, recursos e dependências entre as tarefas para o desenvolvimento do *hardware* e do *software* do componente embarcado. Esse documento é adaptado do artefato do rup rup_sdpln (ver Apêndice C).
- **Especificação do produto:** o documento é a saída de diferentes atividades realizadas ao longo do processo, relacionadas à descrição e à definição do produto. Esse documento reúne o conjunto de informações, especificações e necessidades do produto completo a ser construído (sistema embarcado). É de grande importância, mas opcional, pois o processo cíclico direciona o desenvolvimento de componente de forma individual. Entende-se que a especificação do produto é um documento existente em

instância superior de desenvolvimento. Esse documento é adaptado do artefato do rup rup_vision (ver Apêndice D).

- **Especificação do componente:** esse documento é a saída de diferentes atividades realizadas ao longo do processo, atividades relacionadas à descrição, à definição e ao detalhamento do componente (*hardware* e *software*). Reúne o conjunto de informações, especificações e necessidades do componente a ser produzido e inserido no sistema embarcado. Esse documento é adaptado do artefato do rup rup_cmpln (ver Apêndice E).
- **Projeto de testes:** documento resultante da elaboração de procedimentos sistemáticos para realizar os testes do componente (*hardware* e *software*). Planejado detalhadamente de acordo com o fluxo de trabalho e com a evolução do produto em todas as fases pertinentes. Esse documento é adaptado do artefato do rup rup_tstpln (ver Apêndice F).
- **Componentes externos:** documentação completa dos componentes externos ao projeto que poderão ser utilizados. São considerados externos os componentes construídos em outros projetos ou adquiridos do mercado, como documentos institucionais de outros projetos, especificações do sistema e documentação de componentes externos.
- **Modelo de SW:** representação gráfica resultante da atividade de construção do modelo de *software* que explique as características e o comportamento da parte *software* do componente a ser desenvolvido, bem como sua integração com o *hardware* e sua inserção no sistema embarcado. O conteúdo é inserido no documento de especificação de produto e de componente.
- **Modelo de HW:** representação gráfica resultante da atividade de construção do modelo de *hardware* que explique as características e o comportamento da parte *hardware* do componente a ser desenvolvido, bem como sua integração com o *software* e sua inserção no sistema embarcado. O conteúdo é inserido no documento de especificação de produto e de componente.
- **Software:** programa ou conjunto dos programas construídos de acordo com as especificações e os documentos resultantes das atividades anteriores.

- **Hardware:** conjunto de componentes eletrônicos construídos de acordo com as especificações e os documentos resultantes das atividades anteriores.
- **Produto final:** componente embarcado produzido por meio da integração do *hardware* e do *software*.

7.2 Representação da arquitetura

Na arquitetura apresentada, para cada componente (conjunto *hardware* e *software*) a ser desenvolvido será realizada uma iteração, isto é, para cada componente vai existir um projeto que passará por todas as fases e atividades do processo.

Essa arquitetura é sequencial, conforme apresentado na Figura 7.19, em que uma fase é iniciada após o encerramento da fase anterior. Seus controles de planejamento e mudanças são iniciados na fase de engenharia e realizados em todo o fluxo sequencial do processo.

O controle de projeto e a configuração são atividades de suporte que devem ser realizadas em todas as fases, assim como a verificação e a validação.

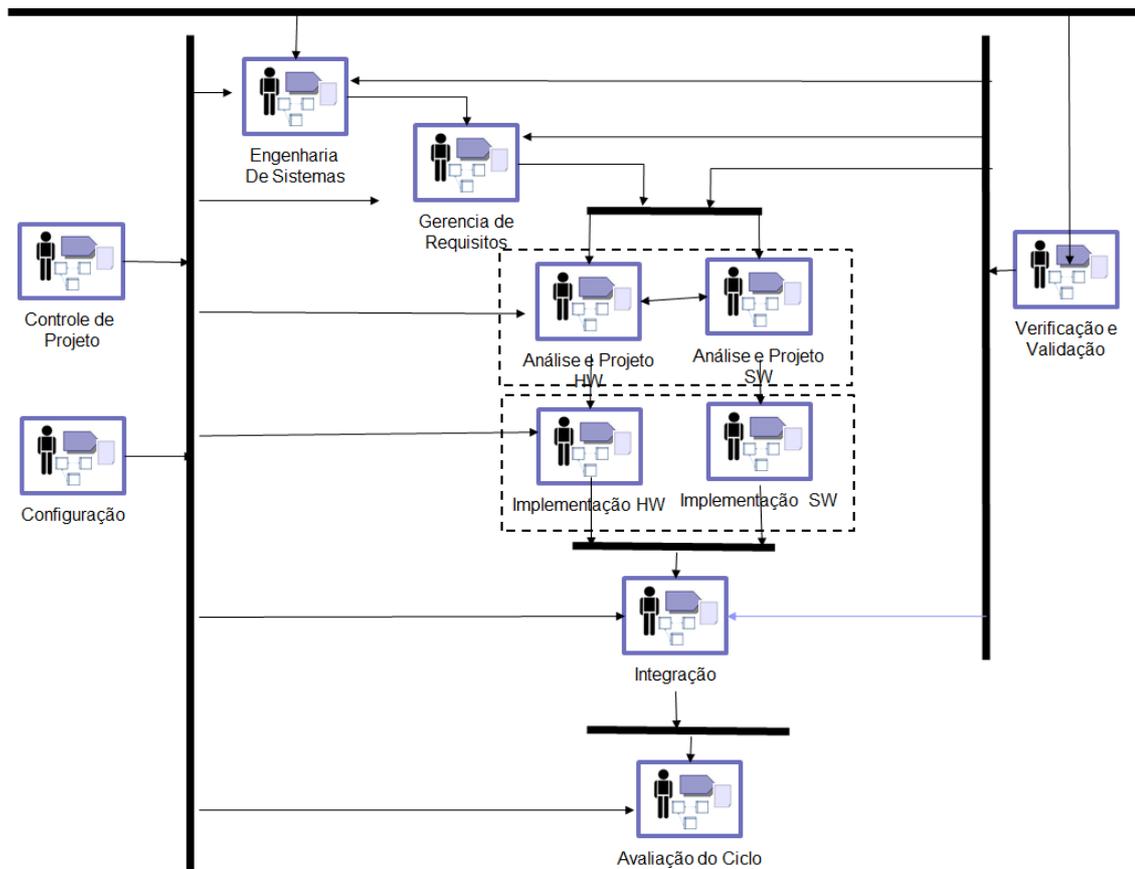


Figura 7.19 – Funcionamento da arquitetura.

Todas as fases produzem ou alimentam um artefato do processo. A Figura 7.20 mostra a linha do tempo de cada artefato produzido: a linha azul representa o momento em que o artefato é criado ou alimentado, e a linha preta mostra os momentos em que o artefato é utilizado sem modificações ou alimentações.

	Engenharia	Requisitos	Análise	Implementação	Integração	V&V	Avaliação
Processo de Desenvolvimento	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Plano de Desenvolvimento	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Configuração	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Plano de Homologação	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Especificação do Produto	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Definição do Componente	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Projeto de Testes	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Componentes externos	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Modelo de SW	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Modelo de HW	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Software	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Hardware	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização
Produto-Final	Criação	Utilização	Utilização	Utilização	Utilização	Utilização	Utilização

Legenda: Criação
 Utilização

Figura 7.20 – Linha do tempo dos artefatos.

7.3 Fluxograma do processo

A Figura 7.21 apresenta a notação utilizada para a elaboração do fluxograma geral do processo, representado nas Figuras 7.22 e 7.23. É preciso considerar que:

- O fluxo sequencial abaixo é uma sugestão para a realização do processo de desenvolvimento de componente embarcado.
- As ações descritas à direita da barra de sincronismo são realizadas sequencialmente.
- A ação descrita à esquerda da barra de sincronismo deve ser realizada em todas as atividades.
- As siglas F1, F2, F3, F4, F5, F6 e FC representam os códigos das fases em que as atividades são realizadas.
- A notação utilizada é:

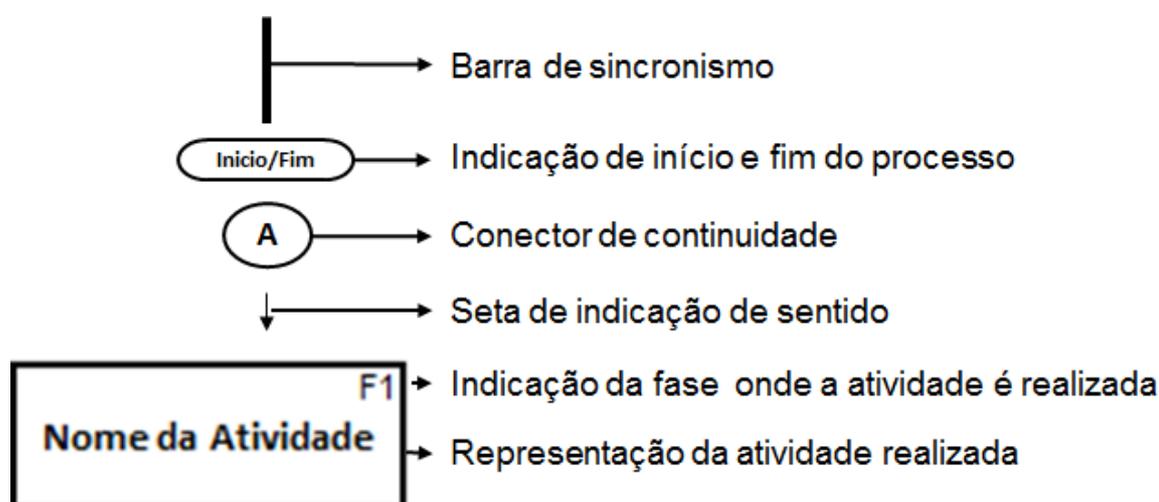


Figura 7.21 – Representação gráfica do fluxograma.

7.3.1 Fluxograma geral do processo

Este fluxograma sugere, de forma geral, as atividades principais que devem ser realizadas contemplando todas as fases do processo (parte 1 e parte 2).



Figura 7.22 – Fluxograma geral de realização do processo – parte 1.

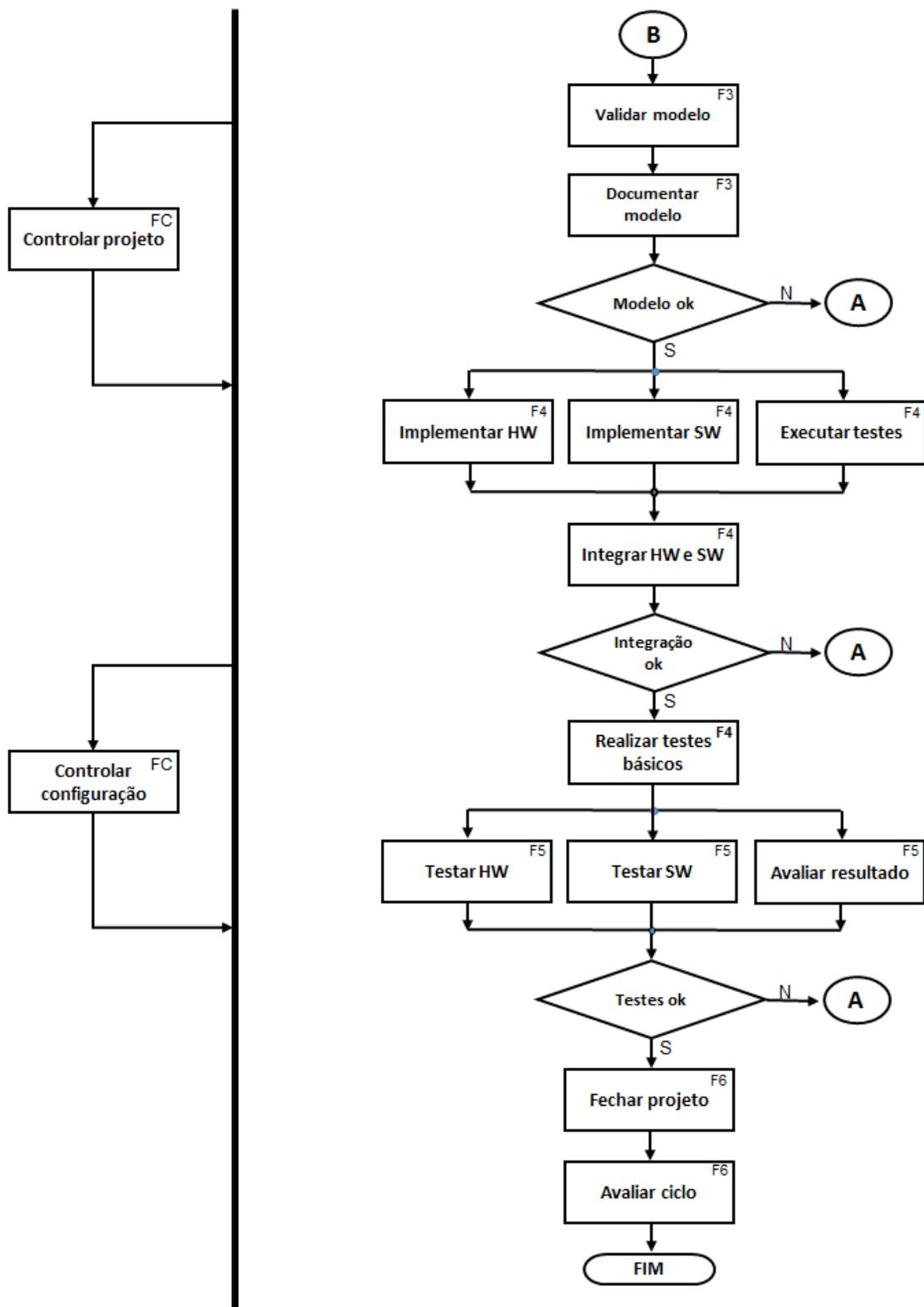


Figura 7.23 – Fluxograma geral de realização do processo – parte 2.

7.3.2 Fluxograma de cada fase do processo

A Figura 7.24 apresenta a notação utilizada para a elaboração dos fluxos de fase, os quais seguem a numeração de 7.25 a 7.30. É preciso considerar os seguintes aspectos:

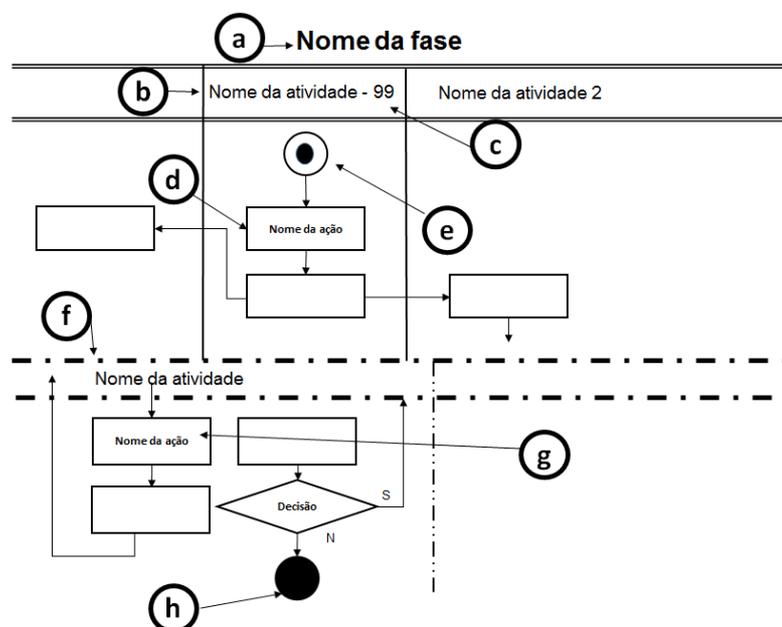


Figura 7.24 – Notação para o fluxograma das fases.

- a) Nome da fase a que o fluxo se refere.
- b) Raia onde são realizadas as ações em cada atividade.
- c) Número sequencial da atividade no processo.
- d) Ação a ser realizada.
- e) As atividades localizadas nessas raia são consequência de todas as ações das atividades das raia acima.
- f) Início do fluxo.
- g) Ação a ser realizada em consequência de todas as outras ações das atividades das raia acima.
- h) Fim do fluxo

Todas as atividades de cada fase do processo têm um fluxograma individual sugerido (ver item 4 do Apêndice B).

7.3.2.1 Fluxograma da fase de engenharia – F1

Este fluxograma sugere, de forma geral, as principais atividades da fase.

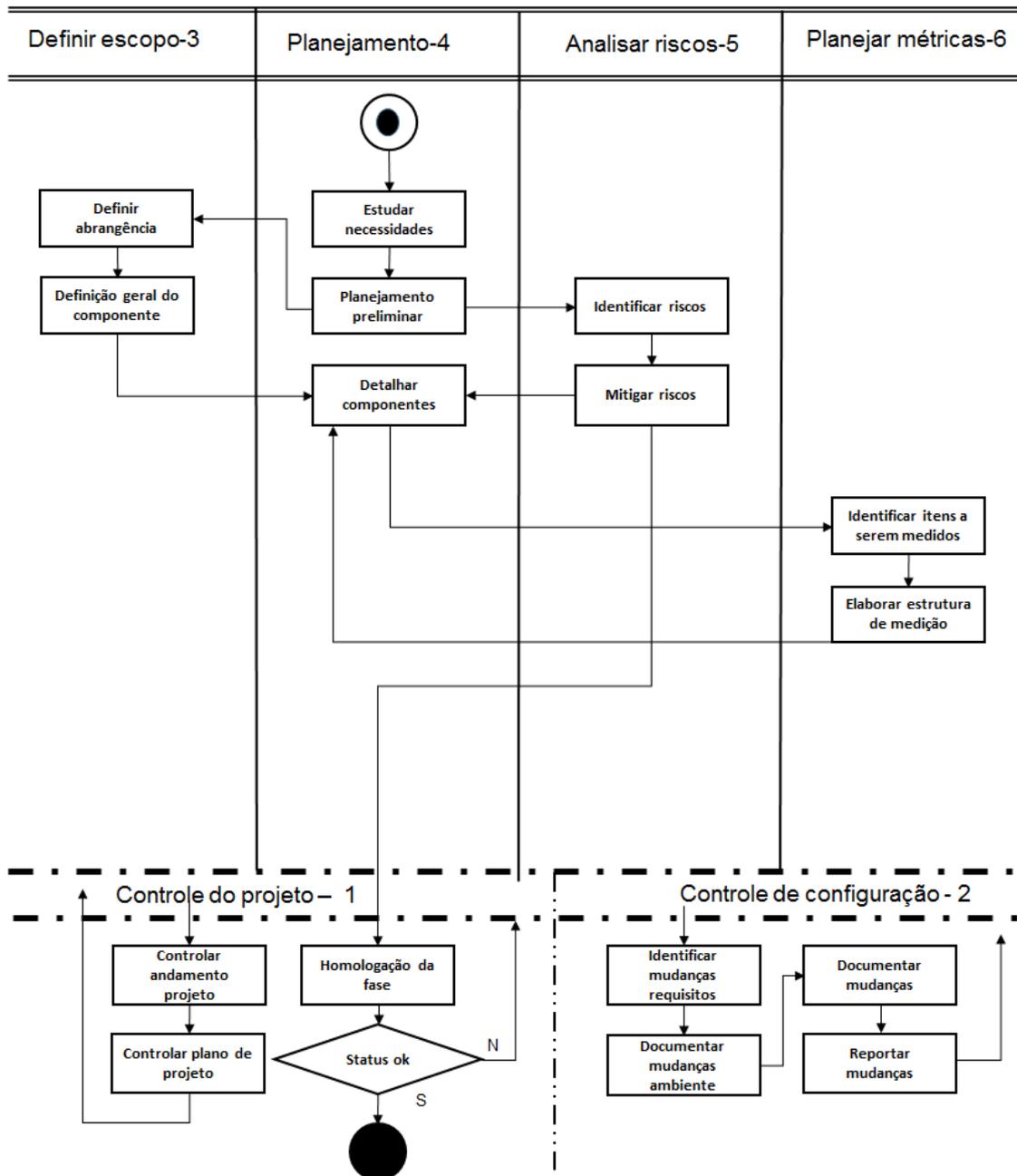


Figura 7.25 – Fluxograma da fase de engenharia.

7.3.2.2 Fluxograma da fase de análise de requisito – F2

Este fluxograma sugere, de forma geral, as principais atividades da fase.

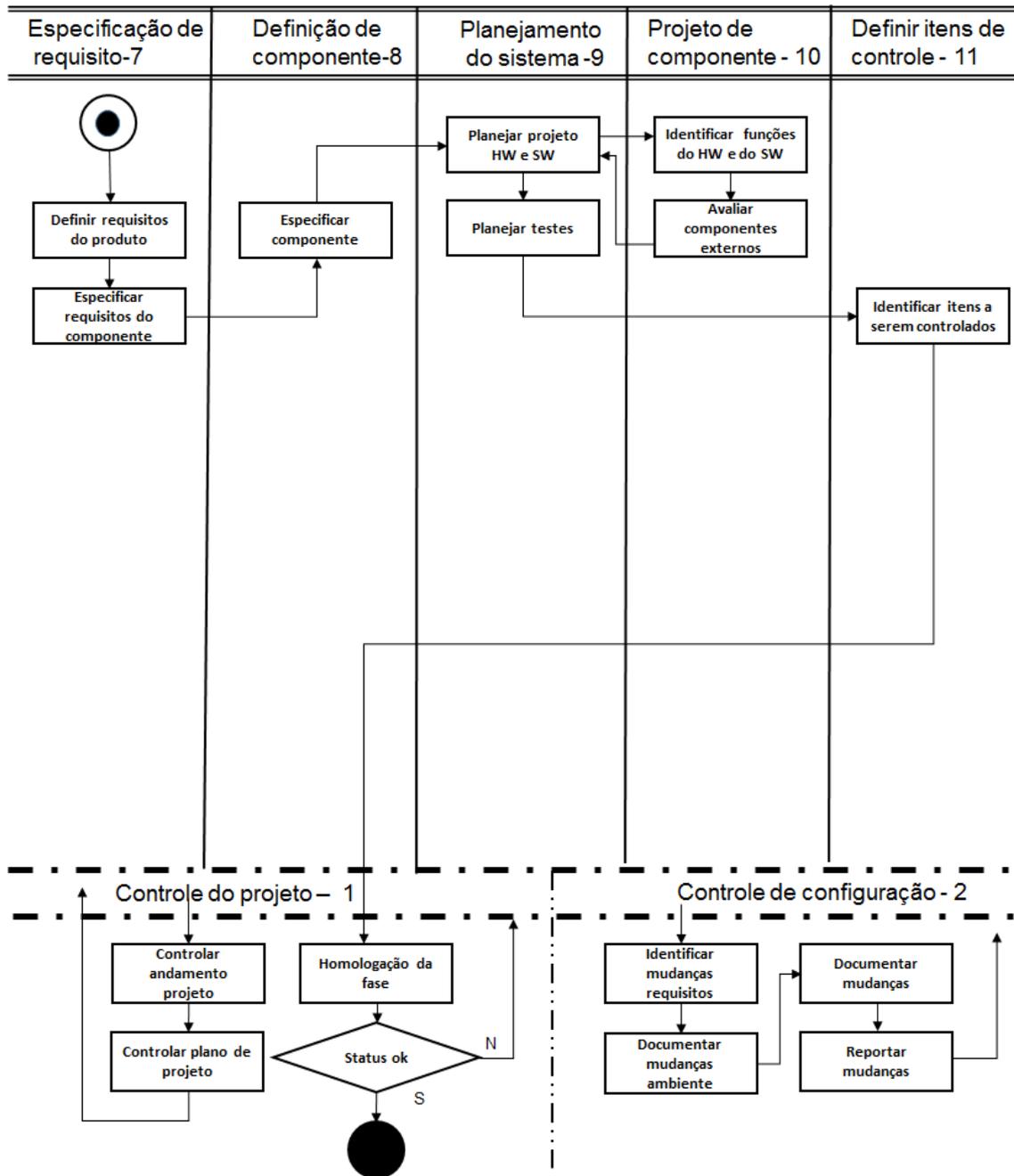


Figura 7.26 – Fluxograma da fase de análise de requisitos.

7.3.2.3 Fluxograma da fase da análise e projeto – F3

Este fluxograma sugere, de forma geral, as principais atividades da fase.

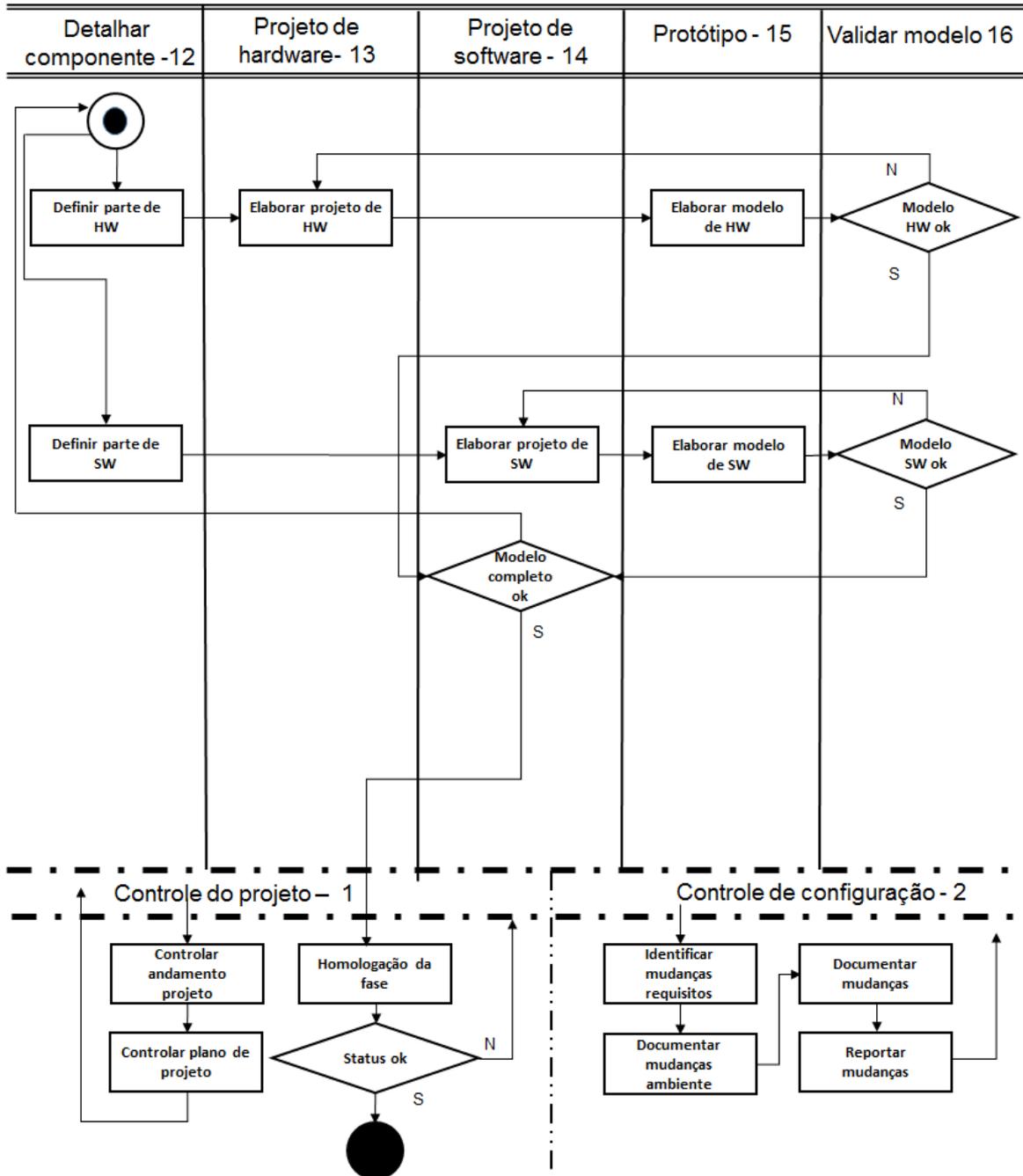


Figura 7.27 – Fluxograma da fase de análise e projeto.

7.3.2.4 Fluxograma da fase de implementação e integração – F4

Este fluxograma sugere, de forma geral, as principais atividades da fase.

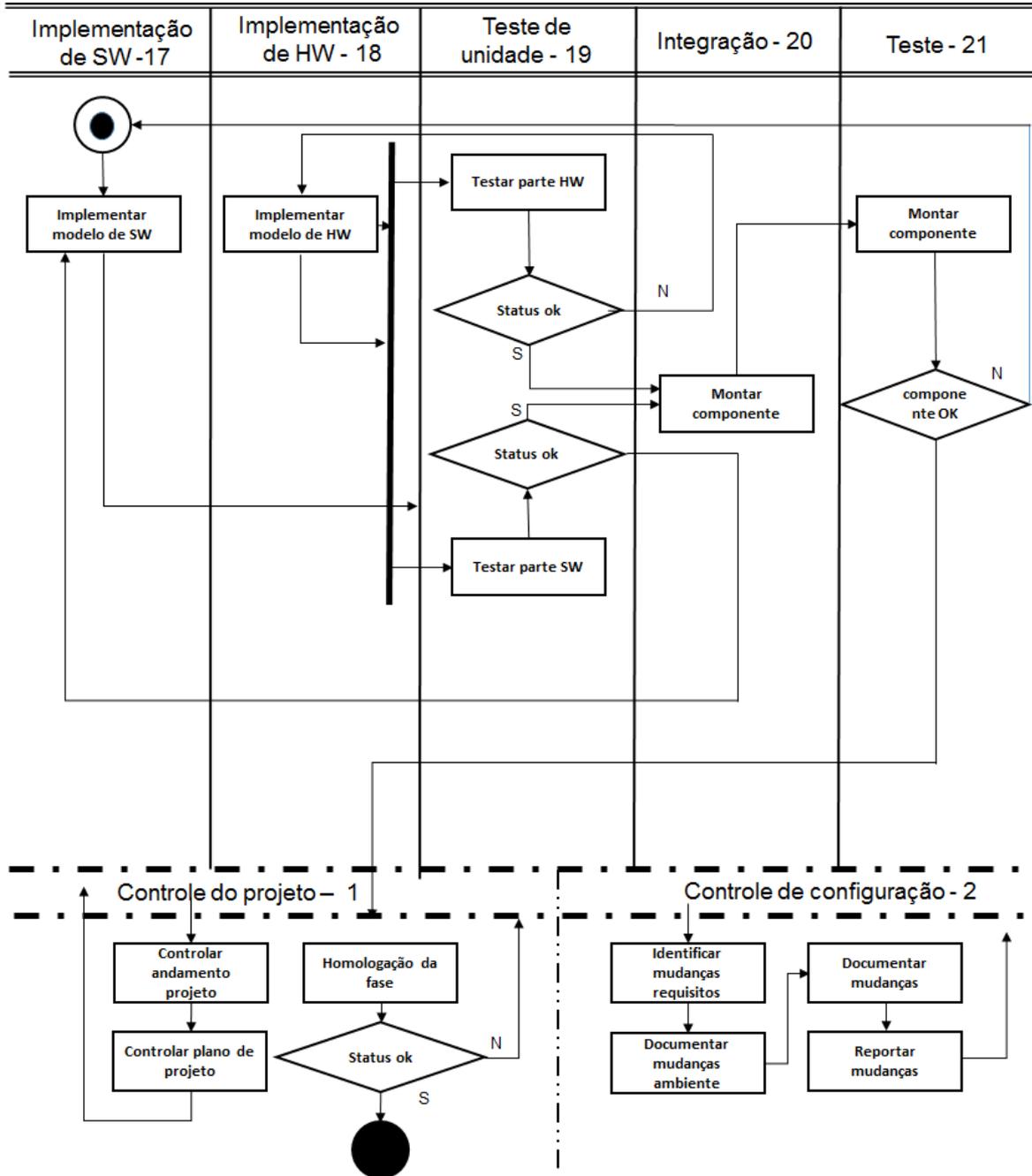


Figura 7.28 – Implementação e integração.

7.3.2.5 Fluxograma da fase da fase de verificação e validação – F5

Este fluxograma sugere, de forma geral, as principais atividades da fase.

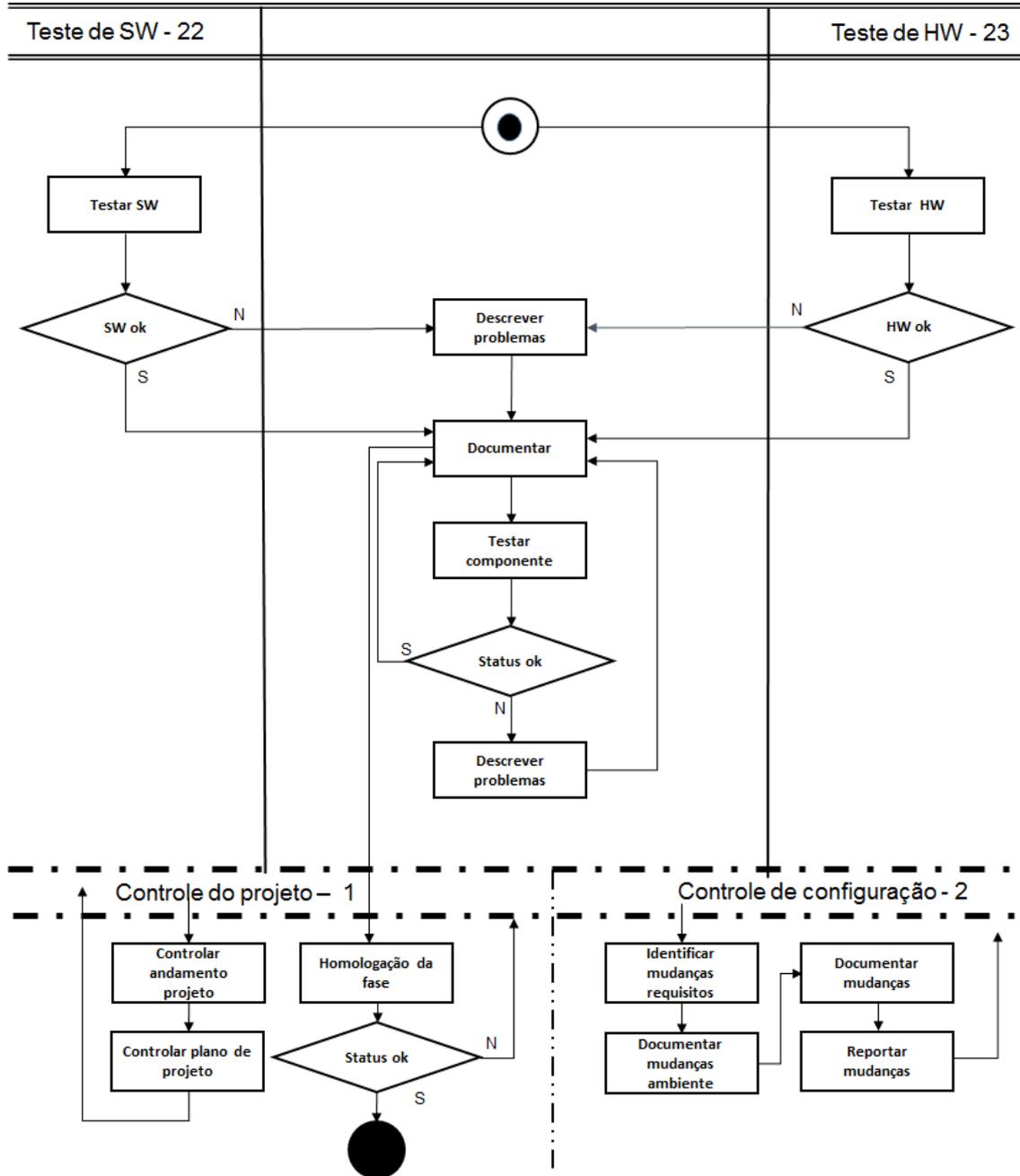


Figura 7.29 – Fluxograma da fase de verificação e validação.

7.3.2.6 Fluxograma da fase da fase de avaliação do ciclo – F6

Este fluxograma sugere, de forma geral, as principais atividades da fase.

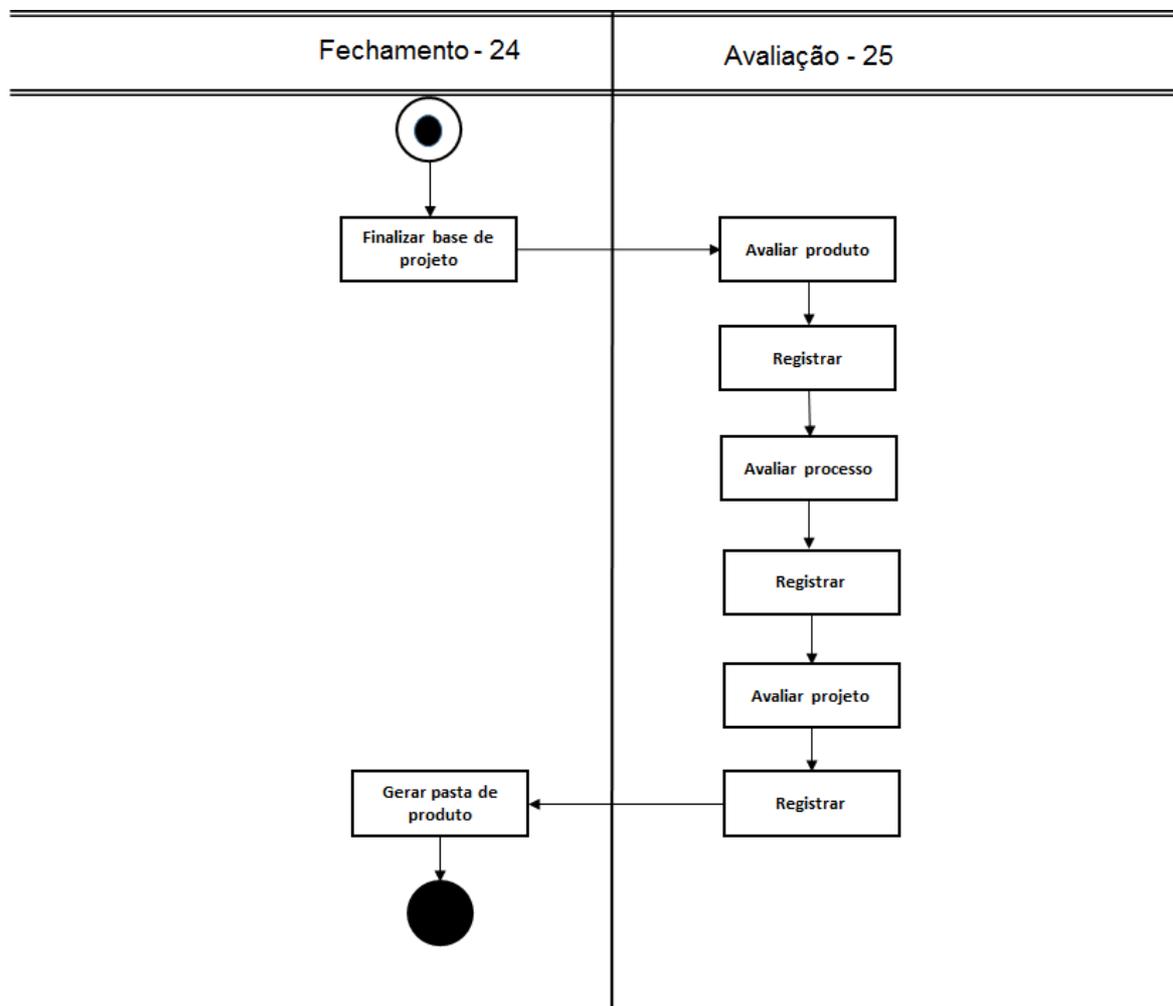


Figura 7.30 – Fluxograma da fase de avaliação do ciclo.

7.4 Maturidade do processo proposto

A documentação de um processo não deve ser vista como mais uma das atividades a serem realizadas no desenvolvimento do produto. Os documentos gerados em um bom processo são resultado de boas práticas realizadas em todas as suas fases, de forma natural e com a credibilidade daqueles que os utilizam.

Para que um processo seja classificado no nível 2 de maturidade, os projetos da organização devem ser planejados e executados de acordo com uma política. O *status* deve ser facilmente identificado e as pessoas envolvidas devem ser capacitadas de acordo com as necessidades identificadas para produzir as saídas esperadas. Todas as partes interessadas devem ser envolvidas, e os controles e as revisões devem ser realizados por meio de monitoramento e validações. Todas essas atividades devem estar de acordo com a descrição do processo.

7.4.1 Atendimento às práticas do CMMI

Indicação da cobertura das práticas do CMMI no processo.

7.4.1.1 Genéricas

A prática genérica auxilia na implementação efetiva do processo. Deve seguir uma estrutura de ação em um contexto de empresa, em que seja descrita e realizada. Em geral, esse tipo de prática é direcionada para a captação do apoio dos grandes gestores para sua realização.

No processo as práticas genéricas são planejadas e oficializadas, conforme mostrado na Tabela 7.1.

Tabela 7.1 – Práticas genéricas

Código	Descrição	Atividade	Documento	Fase
GP 2.2	Planejar o processo.	Planejamento.	Plano de desenvolvimento - Apêndice C, processo de desenvolvimento - Apêndice B, documentos externos.	Engenharia, gerência de requisitos.
GP 2.3	Fornecer recursos.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
GP2.4	Designar responsabilidades.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
GP2.5	Treinar pessoas.	Planejamento.	Plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
GP2.6	Controlar produtos de trabalho.	Controle do projeto.	Plano de Desenvolvimento - Apêndice C, homologação - Apêndice H.	Comum.
GP2.7	Identificar e envolver as partes interessadas relevantes.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
GP2.8	Monitorar e controlar o processo.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
GP2.9	Avaliar objetivamente a adesão.	Controle do projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Comum.
GP2.10	Revisar status com a gerência de nível superior.	Controle do projeto.	Plano de desenvolvimento - Apêndice C, documentos externos.	Comum.

- **GP2.2:** a implementação dessa prática genérica é registrada no plano de desenvolvimento durante o planejamento. Deve ser legitimada conforme um plano institucional e/ou plano de desenvolvimento do produto que ofereça garantia de que ela sempre será realizada.
- **GP2.3:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com a elaboração do plano de especificação de equipe, e aprovada e viabilizada pelos níveis superiores.
- **GP2.4:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com a especificação dos papéis e responsabilidades, e aprovada e viabilizada pelos níveis superiores.
- **GP2.5:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com o planejamento de aquisição e treinamento da equipe segundo um plano institucional da organização, e aprovada e viabilizada pelos níveis superiores.
- **GP2.6:** a implementação dessa prática genérica é registrada na homologação, com o planejamento de aceitação do produto, e também no plano de gerenciamento da configuração, segundo critérios de controle e homologação institucional, e no plano de desenvolvimento do produto, sendo aprovada e viabilizada pelos níveis superiores.

- **GP2.7:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com a definição de todos os responsáveis pelo projeto, tanto por seu controle e sua execução quanto por sua viabilização, esta última com autoridade para tomadas de decisões. Essa atividade deve ser previamente aprovada e viabilizada pelos níveis superiores de forma institucional.
- **GP2.8:** a implementação dessa prática genérica é registrada no plano de desenvolvimento com o acompanhamento do projeto, e deve ser aprovada e viabilizada pelos níveis superiores de acordo com diretrizes institucionais.
- **GP2.9:** a implementação dessa prática genérica é registrada no plano de homologação e realizada de acordo com o plano de desenvolvimento, com avaliação e aprovação dos artefatos produzidos em cada fase do projeto. Deve ser aprovada e viabilizada pelos níveis superiores de forma institucional.
- **GP2.10:** a implementação dessa prática genérica é registrada no plano de homologação e no plano de desenvolvimento, ao final de cada fase, e realizada de acordo com o plano de desenvolvimento. Deve ser aprovada e viabilizada pelos níveis superiores de forma institucional.

7.4.2 Específicas de nível 2

7.4.2.1 Gerência de requisitos

A atividade de coleta de requisitos acontece em sua respectiva fase, mas as atividades de controle e alinhamento são realizadas em todas as fases. Serve de base estrutural para as demais atividades, conforme mostra a Tabela 7.2.

Tabela 7.2 – Gerência de requisitos

Código	Descrição	Atividade	Documento/Apêndice	Fase
SP1.1	Entender requisitos.	Definição de escopo, definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice F, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP1.2	Obter comprometimento com os requisitos.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.3	Gerenciar mudanças nos requisitos.	Controlar configuração, definir itens de controle.	Configuração - Apêndice G.	Comum.
SP1.4	Manter rastreabilidade bidirecional dos requisitos.	Controlar configuração.	Configuração - Apêndice G (matrix de rastreabilidade).	Comum, gerência de requisitos.
SP1.5	Garantir alinhamento entre produtos de trabalho, planos de projeto e requisitos.	Controle de projeto.	Homologação - Apêndice H.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com o entendimento dos requisitos do componente, bem como seu posicionamento no produto final. Deve ser revisada e validada com os responsáveis, de acordo com o planejado.
- **SP1.2:** a implementação dessa prática específica se dá com a identificação das necessidades de mudança dos requisitos e a análise de suas consequências internas e externas ao componente, bem como com a identificação de mudanças de requisito externas ocorridas e seu impacto no componente.
- **SP1.3:** a implementação dessa prática específica se dá com o controle rígido das mudanças de requisitos que acontecerem, independentemente do estímulo a essa mudança, de acordo com o plano de desenvolvimento e o controle de configuração.
- **SP1.4:** a implementação dessa prática específica se dá com o mapeamento e o controle dos requisitos e seus desdobramentos até o último nível, em que de um ponto do requisito consegue-se atingir seus pares, bem como planos de testes, configuração, documentação, entre outros. O nível de controle deve seguir o que foi definido no plano de desenvolvimento.
- **SP1.5:** a implementação dessa prática específica se dá com a realização das atividades de acompanhamento e a homologação dos artefatos produzidos em cada fase do processo, de acordo com o plano de desenvolvimento.

7.4.2.2 Medição e análise

As atividades da área de processo de medição e análise nascem na fase de engenharia e repetem-se em todas as fases, por meio das atividades comuns. As distribuições da cobertura de suas práticas ocorrem conforme disposto na Tabela 7.3.

Tabela 7.3 – Medição e Análise

Código	Descrição	Atividade	Documento	Fase
SP1.1	Estabelecer objetivos de medição.	Planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Especificar medidas.	Planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.3	Especificar procedimentos de coleta e armazenamento de dados.	Planejamento, planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.4	Especificar procedimento de análise.	Planejamento, planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Obter dados resultantes de medição.	Planejamento, planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Analisar dados resultantes de medição.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP2.3	Armazenar dados resultantes.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP2.4	Comunicar resultados.	Avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Avaliação do ciclo.

- **SP1.1:** a implementação dessa prática específica se dá com a realização das atividades de identificação da necessidade de medição dos itens a serem utilizados no projeto e no planejamento.
- **SP1.2:** a implementação dessa prática específica se dá com a realização das atividades de definição das medidas a serem feitas para o projeto, de acordo com a necessidade.
- **SP1.3:** a implementação dessa prática específica se dá com a realização das atividades de definição das medidas a serem feitas e a especificação de meios de armazenamento dessas medidas.
- **SP1.4:** a implementação dessa prática específica se dá com a definição da análise dos itens medidos e a comunicação aos níveis superiores e aos interessados, de acordo com o plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com a definição da estrutura de armazenamento e de análise dos dados, de acordo com o plano de desenvolvimento.
- **SP2.2:** a implementação dessa prática específica se dá com a análise dos dados armazenados, de acordo com o plano de desenvolvimento e homologado com os níveis superiores.
- **SP2.3:** a implementação dessa prática específica se dá com a documentação dos resultados da análise, de acordo com o plano de desenvolvimento.

- **SP2.4:** a implementação dessa prática específica se dá com a homologação dos resultados dos dados de medidos e analisados, de acordo com o plano de desenvolvimento ao final de cada fase e do projeto.

7.4.2.3 Acompanhamento e controle de projeto

As atividades da área de processo, de acompanhamento e controle de projeto, iniciam-se na fase de engenharia, conforme apresentado na Tabela 7.4, e estendem-se nas atividades comuns.

Tabela 7.4 – Acompanhamento e controle de projeto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Monitorar os parâmetros de planejamento do projeto.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.2	Monitorar compromissos.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.3	Monitorar riscos do projeto.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.4	Monitorar a gestão de dados.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.5	Monitorar o envolvimento das partes interessadas.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.6	Conduzir revisões de progresso.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP1.7	Conduzir revisões de macro.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP2.1	Analisar as questões.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP2.2	Implementar ações corretivas.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP2.3	Gerenciar ações corretivas.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.

- **SP1.1:** a implementação dessa prática específica se dá com o estudo e a documentação dos parâmetros controlados pelo projeto e seus resultados, de acordo com o plano de desenvolvimento.
- **SP1.2:** a implementação dessa prática específica se dá com a homologação do acompanhamento e o monitoramento das medições realizadas durante a realização do projeto em cada fase, de acordo com o plano de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com o acompanhamento dos riscos identificados e documentados no plano do projeto e revisados em cada fase do projeto, de acordo com o plano de desenvolvimento.
- **SP1.4:** a implementação dessa prática específica se dá com a realização da atividade de homologação feita em cada fase do projeto, de acordo com o plano de desenvolvimento.

- **SP1.5:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento.
- **SP1.6:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento, e também deve ser feita sempre que necessário.
- **SP1.7:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento, e também deve ser feita sempre que necessário.
- **SP2.2:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento, e também deve ser feita sempre que necessário, com base nas questões identificadas.
- **SP2.3:** a implementação dessa prática específica se dá com a realização de atividades identificadas para solucionar questões críticas, devendo ser incluída no plano de desenvolvimento e receber acompanhamento específico.

7.4.2.4 Gerência de acordo com fornecedores

As atividades da área de processo de gerência de acordo com fornecedores devem ter suas práticas voltadas para todos os tipos de fornecedores envolvidos no processo, entre eles fornecedores de componentes, de *softwares*, de ferramentas para o desenvolvimento, enfim, de todas as aquisições que acontecerão durante o projeto, conforme a Tabela 7.5.

Tabela 7.5 – Gerência de acordos com fornecedores

Código	Descrição	Atividade	Documento	Fase
SP1.1	Determinar o tipo de aquisição.	Planejamento, planejamento do sistema, projeto do sistema.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Engenharia, gerência de requisitos.
SP1.2	Selecionar fornecedores.	Planejamento.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Comum, gerência de requisitos.
SP1.3	Estabelecer contratos com fornecedores.	Planejamento.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Comum, gerência de requisitos.
SP2.1	Executar contrato com fornecedores.	Planejamento.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Comum, gerência de requisitos.
SP2.2	Aceitar o produto adquirido.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D, homologação - Apêndice H, projeto de teste - Apêndice F.	Comum.
SP2.3	Garantir a transição de produtos.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com a identificação dos componentes que poderão ser adquiridos no mercado ou reutilizados de outros projetos. Deve ser incluída no plano de desenvolvimento, e sua documentação deve ser anexada à documentação produzida pelo projeto, entrando no processo normal de desenvolvimento, de acordo com o plano.
- **SP1.2:** a implementação dessa prática específica se dá com a identificação dos fornecedores dos componentes que serão adquiridos no mercado ou reutilizados de outros projetos, devendo seguir os critérios de seleção de fornecedores definidos no plano de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com a oficialização do fornecimento por meio do contrato e sua inclusão na documentação do projeto.
- **SP2.1:** a implementação dessa prática específica se dá com o início e o acompanhamento das atividades desenvolvidas pelo fornecedor e avaliadas na homologação, ao final de cada fase do processo, sempre que for necessário.
- **SP2.2:** a implementação dessa prática específica se dá com o registro e o acompanhamento das atividades desenvolvidas pelo fornecedor e avaliadas na homologação, ao final de cada fase do processo, sempre que for necessário.
- **SP2.3:** a implementação dessa prática específica se dá com o acompanhamento dos produtos desenvolvidos pelo fornecedor e avaliados na homologação, ao final de cada fase do processo, sempre que for necessário.

7.4.2.5 Planejamento de projeto.

As atividades da área de processo de planejamento de projeto nascem na fase de engenharia e estendem-se por todo o ciclo de vida. Elas devem ser realizadas com muita propriedade, pois servem de base para a realização das demais atividades. O processo proposto cobre as práticas dessa área de processo, conforme mostrado na Tabela 7.6.

Tabela 7.6 – Planejamento de projeto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Estimar o escopo do projeto.	Definição de escopo.	Especificação do produto - Apêndice F, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia
SP1.2	Estabelecer estimativas para atributos de produtos de trabalho e de tarefas.	Especificação dos requisitos.	Especificação do produto - Apêndice F, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Gerência dos requisitos.
SP1.3	Definir fases do ciclo de vida do projeto.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.4	Estimativas de esforço e custo.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Estabelecer orçamento e cronograma.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Identificar riscos do projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.3	Plano de gerenciamento de dados.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.4	Planejar recursos do projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.5	Planejar habilidades e conhecimentos necessários.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.6	Planejar o envolvimento das partes interessadas.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.7	Estabelecer o plano do projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.1	Revisar planos que afetam o projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia, comum.
SP3.2	Conciliar carga de trabalho e recursos.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.3	Obter comprometimento com o plano.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.

- **SP1.1:** a implementação dessa prática específica se dá com a identificação e a definição da estrutura do componente a ser desenvolvido para sua estimativa.
- **SP1.2:** a implementação dessa prática específica se dá com a identificação dos atributos do componente a ser desenvolvido, que serão considerados para a estimativa do projeto.
- **SP1.3:** a implementação dessa prática específica se dá com a organização das fases e atividades que serão realizadas durante o projeto.
- **SP1.4:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, identificando a estimativa de custo e tempo levando em consideração as estimativas indicadas pelo produto.

- **SP2.1:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, de acordo com o plano de desenvolvimento.
- **SP2.2:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, avaliando o risco do projeto de acordo com o plano de desenvolvimento. É realizada com foco nos riscos do produto.
- **SP2.3:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, identificando os dados que devem ser medidos e acompanhados durante o projeto, de acordo com o plano de desenvolvimento. Esses dados são identificados de acordo com a criticidade do componente e atendem ao produto e ao processo.
- **SP2.4:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, identificando e alocando os recursos humanos necessários para seu desenvolvimento, bem como a especificação e a providência do ambiente necessário para o desenvolvimento do projeto.
- **SP2.5:** a implementação dessa prática específica se dá com a identificação das características técnicas e admirativas para a execução do projeto.
- **SP2.6:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, com a alocação dos recursos humanos para a execução das atividades.
- **SP2.7:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, com a indicação da estimativa de tempo para a execução de cada atividade.
- **SP3.1:** a implementação dessa prática específica se dá com o acompanhamento do plano de projeto e homologado ao final de cada fase.
- **SP3.2:** a implementação dessa prática específica se dá com a realização das atividades de acompanhamento e o replanejamento do projeto para atender a alguma demanda ou planejamento não cumprido.

7.4.2.6 Gerência da configuração

As atividades da área de processo de gerência da configuração têm seu escopo definido na fase de engenharia; nesta fase são controladas todas as configurações e todas as mudanças que acontecem em todas as atividades de todas as fases e os controles não se resumem aos requisitos e versões de programas ou componentes, mas sim englobam todos os elementos que fazem parte do ambiente de desenvolvimento e implantação do projeto. A distribuição de suas atividades estão representadas na Tabela 7.7.

Tabela 7.7 – Gerência de configuração

Código	Descrição	Atividade	Documento	Fase
SP1.1	Identificar itens de configuração.	Definir itens de controle.	Configuração - Apêndice G.	Gerência dos requisitos.
SP1.2	Estabelecer um sistema de gestão de configuração.	Planejamento.	Plano de desenvolvimento - Apêndice C, configuração - Apêndice G, homologação - Apêndice H.	Engenharia.
SP1.3	Criar ou liberar baselines.	Controlar configuração.	Configuração - Apêndice G.	Comum.
SP2.1	Acompanhar solicitações de mudanças.	Controlar configuração, controle de projeto.	Configuração - Apêndice G, plano de desenvolvimento - Apêndice C.	Comum.
SP2.2	Controlar itens de configurações.	Controlar configuração.	Configuração - Apêndice G.	Comum.
SP3.1	Estabelecer registros de gestão de configuração.	Controlar configuração.	Configuração - Apêndice G.	Comum.
SP3.2	Realizar auditorias de configuração.	Controle de projeto.	Homologação - Apêndice H.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com a identificação e o planejamento dos itens que deverão ser controlados durante o projeto de acordo com o plano de desenvolvimento.
- **SP1.2:** a implementação dessa prática específica se dá com o estabelecimento do mecanismo de controle de mudanças e configuração.
- **SP1.3:** a implementação dessa prática específica se dá com o registro das mudanças de acordo com o plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com a análise da solicitação da mudança e o registro do impacto no componente e no plano de desenvolvimento.
- **SP2.2:** a implementação dessa prática específica se dá com o registro das mudanças e o controle da configuração utilizada.
- **SP3.1:** a implementação dessa prática específica se dá com o registro das mudanças, o reporte e o controle da configuração utilizada.

- **SP3.2:** a implementação dessa prática específica se dá com a manutenção dos registros e avaliação dos dados da configuração.

7.4.2.7 Garantia da qualidade

No processo proposto, as atividades da área de processo de garantia da qualidade têm a responsabilidade de consolidar a execução e o resultado de todas as atividades do processo e a qualidade do produto construído. Suas atividades são apresentadas na Tabela 7.8.

Tabela 7.8 – Garantia da qualidade de processo e produto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Avaliar objetivamente os processos.	Avaliação do ciclo.	Projeto, homologação - Apêndice H.	Avaliação do ciclo.
SP1.2	Avaliar objetivamente produtos de trabalho.	Teste SW, teste HW.	Plano de desenvolvimento - Apêndice C, plano de teste - Apêndice F, projeto, homologação - Apêndice H.	Verificação e validação.
SP2.1	Comunicar e resolver problemas de descumprimento.	Fechamento, avaliação do ciclo.	Projeto, Homologação - Apêndice H	Avaliação do ciclo.
SP2.2	Estabelecer registros.	Avaliação do ciclo.	Homologação - Apêndice H	Avaliação do ciclo.

- **SP1.1:** a implementação dessa prática específica se dá com a realização das atividades de avaliação do andamento do projeto do componente, de acordo com o plano do projeto e com a descrição, bem como de avaliação dos artefatos gerados com relação à especificação do produto.
- **SP1.2:** a implementação dessa prática específica se dá com a realização das atividades dos artefatos produzidos com relação à especificação do produto.
- **SP2.1:** a implementação dessa prática específica se dá com a realização das atividades homologação em cada fase do projeto, registrando e comunicando as não conformidades.
- **SP2.2:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto com o registro das avaliações realizadas.

7.4.3 Específicas de nível 3

7.4.3.1 Desenvolvimento de requisitos

No processo proposto esta área de processo é atendida com o detalhamento do requisito, sua validação nos cenários necessários, e suas atividades são apresentadas na Tabela 7.9.

Tabela 7.9 – Desenvolvimento de requisitos

Código	Descrição	Atividades	Documento	Fase
SP1.1	Levantar necessidades.	Definição de escopo, definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP1.2	Transformar as necessidades dos envolvidos em requisitos do cliente.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Estabelecer requisitos de produto e de componente de produto.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP2.2	Alocar requisitos de componente de produto.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.3	Identificar requisitos de interface.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP3.1	Estabelecer conceitos operacionais e cenários.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP3.2	Estabelecer uma definição da funcionalidade requerida e atributos de qualidade.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP3.3	Analisar requisitos.	Definição dos requisitos, definição dos componentes, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, homologação - Apêndice H.	Engenharia, gerência de requisitos, comum.
SP3.4	Analisar requisitos visando ao balanceamento.	Definição dos requisitos, definição dos componentes, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos, comum.
SP3.5	Validar requisitos.	Controle de projeto.	Homologação - Apêndice H.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com o levantamento dos dados preliminares e detalhados do produto/componente a ser desenvolvido nas fases de engenharia e gerência de requisitos, respectivamente.
- **SP1.2:** a implementação dessa prática específica acontece com o detalhamento dos requisitos do usuário na especificação geral do componente.
- **SP2.1:** a implementação dessa prática específica é resultado do acompanhamento do projeto e homologação, quando, ao longo do processo, é avaliada a aderência dos componentes ao produto.
- **SP2.2:** a implementação dessa prática específica se dá com a atividade de distribuição das funcionalidades e da modelagem do componente.
- **SP2.3:** a implementação dessa prática específica se dá com a atividade de distribuição das funcionalidades e da modelagem do componente.

- **SP3.1:** a implementação dessa prática específica se dá com as atividades de identificação e construção do ambiente operacional e de desenvolvimento do componente.
- **SP3.2:** a implementação dessa prática específica se dá com a identificação, definição e especificação dos requisitos especiais do componente.
- **SP3.3:** a implementação dessa prática específica se dá com a atividade de homologação e verificação dos requisitos de acordo com o plano, ao longo do projeto.
- **SP3.4:** a implementação dessa prática específica se dá com a identificação, definição e especificação dos requisitos especiais e restrições do componente.
- **SP3.5:** a implementação dessa prática específica se dá com as atividades de homologação do componente em cada fase do projeto, de acordo com o planejamento.

7.4.3.2 Solução técnica

O atendimento dessa área de processo se faz com o estudo e a proposta de soluções para o desenvolvimento ou a aquisição do componente. Suas atividades são apresentadas na Tabela 7.10.

Tabela 7.10 – Solução técnica

Código	Descrição	Atividades	Documento	Fase
SP1.1	Desenvolver soluções alternativas e critérios de seleção.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, Gerencia de Requisitos
SP1.2	Selecionar soluções de componentes de produto.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, Gerencia de Requisitos
SP2.1	Desenvolver o design do produto ou dos componentes de produto.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP2.2	Estabelecer um pacote de dados técnicos.	Controlar configuração.	Configuração.	Comum.
SP2.3	Projetar interfaces utilizando critérios.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP2.4	Analisar alternativas: desenvolver, comprar ou reusar.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP3.1	Implementar design.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP3.2	Elaborar a documentação de suporte ao produto.	Não.		

- **SP1.1:** a implementação dessa prática específica se dá com o estudo de soluções e a proposta de soluções, de acordo com os critérios que foram definidos no plano de desenvolvimento.
- **SP1.2:** a implementação dessa prática específica se dá com o estudo de soluções e a proposta de soluções, de acordo com os critérios que foram definidos no plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com a realização da atividade de modelagem do componente.
- **SP2.2:** a implementação dessa prática específica se dá com a definição, o controle e a armazenagem da configuração de todos os artefatos produzidos durante o projeto.
- **SP2.3:** a implementação dessa prática específica se dá com a realização da atividade de modelagem do componente.
- **SP2.4:** a implementação dessa prática específica se dá com o estudo de soluções e a proposta de soluções, de acordo com os critérios que foram definidos no plano de desenvolvimento.
- **SP3.1:** a implementação dessa prática específica se dá com a implementação do componente.
- **SP3.2:** a implementação dessa prática específica não pode ser considerada atendida, pois direciona para a confecção da documentação para o usuário final (entende-se por *usuário final* a pessoa que vai usar o produto). Esse processo foca o desenvolvimento de um componente que não é um usuário convencional, por isso é considerado não realizado, porém, segundo a visão de desenvolvimento, todos os documentos produzidos pelo projeto poderiam ser considerados uma documentação, mas não uma documentação para o usuário final.

7.4.3.3 Integração de produto

Essa área de processo é atendida com a realização das atividades de integração do processo proposto e realizada na fase de integração. Suas atividades são apresentadas na Tabela 7.11.

Tabela 7.11 – Integração de produto

Código	Descrição		Documento	Fase
SP1.1	Estabelecer uma estratégia de integração.	Planejamento Projeto do Componente	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP1.2	Estabelecer ambiente de integração do produto.	Controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Gerência de requisitos, comum.
SP1.3	Estabelecer procedimentos e critérios para integração do produto.	Planejamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Revisar descrições de interfaces para assegurar completude.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP2.2	Gerenciar interfaces.	Controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Gerência de requisitos, comum.
SP3.1	Confirmar se os componentes do produto estão prontos para serem integrados.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP3.2	Montar componentes do produto.	Integração	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Implementação e integração.
SP3.3	Avaliar componentes de produto montados.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP3.4	Empacotar e entregar produto ou componente de produto.	Fechamento.	Projeto.	Avaliação do ciclo.

- **SP1.1:** a implementação dessa prática específica se dá com a modelagem do componente e planejamento.
- **SP1.2:** a implementação dessa prática específica se dá com o controle da configuração, de acordo com o planejamento.
- **SP1.3:** a implementação dessa prática específica se dá com a modelagem do componente e planejamento.
- **SP2.1:** a implementação dessa prática específica se dá com a realização das atividades de homologação, de acordo com o planejamento.
- **SP2.2:** a implementação dessa prática específica se dá com realização das atividades de controle da configuração, de acordo com o planejamento.
- **SP3.1:** a implementação dessa prática específica se dá com realização das atividades de controle de projeto e homologação, de acordo com o planejamento.

- **SP3.2:** a implementação dessa prática específica se dá com a realização da atividade de integração do produto.
- **SP3.3:** a implementação dessa prática específica se dá com a realização da atividade de controle de projeto, de acordo com o planejamento.
- **SP3.4:** a implementação dessa prática específica se dá com realização da atividade de fechamento na fase de avaliação do ciclo, de acordo com o planejamento.

7.4.3.4 Verificação

As atividades de verificação são planejadas durante o planejamento do projeto e realizadas na fase de verificação e validação. Suas atividades são apresentadas na Tabela 7.12.

Tabela 7.12 – Verificação

Código	Descrição	Atividades	Documento	Fase
SP1.1	Selecionar produtos de trabalho para verificação.	Planejamento, testes SW, testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Engenharia, verificação e validação.
SP1.2	Estabelecer ambiente de verificação.	Planejamento e controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Engenharia, comum.
SP1.3	Estabelecer procedimentos e critérios de verificação.	Planejamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Preparar-se para revisão por pares.	Não.		
SP2.2	Conduzir revisão por pares.	Não.		
SP2.3	Analisar dados de revisão por pares.	Não.		
SP3.1	Realizar verificação.	Testes SW, testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.
SP3.2	Analisar resultados da verificação.	Testes SW, testes HW e fechamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação e avaliação do ciclo.

- **SP1.1:** a implementação dessa prática específica se inicia com a realização da atividade de planejamento e continua com as atividades de testes de *software* e *hardware*.
- **SP1.2:** a implementação dessa prática específica se dá com a realização da atividade, a definição e a preparação do ambiente de desenvolvimento.

- **SP1.3:** a implementação dessa prática específica se dá com a realização da atividade de planejamento.
- **SP2.1:** essa prática específica é considerada não atendida, pois esse processo também pode ser utilizado para desenvolvimento individual. Contudo, essa prática pode ser planejada para a atividade de planejamento do projeto.
- **SP2.2:** essa prática específica é considerada não atendida, pois esse processo também pode ser utilizado para desenvolvimento individual. Contudo, essa prática pode ser planejada para a atividade de planejamento do projeto.
- **SP2.3:** essa prática específica é considerada não atendida, pois esse processo também pode ser utilizado para desenvolvimento individual. Contudo, essa prática pode ser planejada para a atividade de planejamento do projeto.
- **SP3.1:** a implementação dessa prática se dá com as atividades de testes de *software* e testes de *hardware*, de acordo com o planejamento e o plano de testes.
- **SP3.2:** a implementação dessa prática se dá com as atividades de homologação e a realização da fase de avaliação do ciclo, de acordo com o planejamento.

7.4.3.5 Validação

As atividades das fases de verificação e validação atendem à área de processo ao longo do projeto, segundo seu planejamento e homologação de avaliação do ciclo. Suas atividades são apresentadas na Tabela 7.13.

Tabela 7.13 – Validação

Código	Descrição	Atividades	Documento	Fase
SP1.1	Selecionar produtos para validação.	Testes SW e testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.
SP1.2	Estabelecer ambiente de validação.	Controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Comum
SP1.3	Estabelecer procedimentos e critérios de validação.	Planejamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Realizar validação	Testes SW e testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.
SP2.2	Analisar resultados de validação.	Testes SW e testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.

- **SP1.1:** a implementação dessa prática específica se dá com a realização da atividade de planejamento.
- **SP1.2:** a implementação dessa prática específica se dá com a realização da atividade, a definição e a preparação do ambiente de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com a realização da atividade de planejamento.
- **SP2.1:** a implementação dessa prática se dá com as atividades de testes de *software* e testes de *hardware*, de acordo com o planejamento e o plano de testes.
- **SP2.2:** a implementação dessa prática se dá com as atividades de homologação e a realização da fase de avaliação do ciclo, de acordo com o planejamento.

7.4.3.6 Foco no processo da organização

Essa área de processo não é atendida diretamente pelo processo proposto, pois trata-se de atividades realizadas de forma organizacional. As atividades dessa área de processo são apresentadas na Tabela 7.14.

Tabela 7.14 – Foco no processo da organização

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer necessidades de processo da organização.	Não.		
SP1.2	Avaliar os processos da organização.	Não.		
SP1.3	Identificar melhorias para os processos da organização.	Não.		
SP2.1	Estabelecer planos de ação de processo.	Não.		
SP2.2	Implementar planos de ação de processo.	Não.		
SP3.1	Implantar ativos de processo da organização.	Não.		
SP3.2	Implantar processos padrão.	Não.		
SP3.3	Monitorar a implementação.	Não.		
SP3.4	Incorporar experiências em ativos de processos organizacionais.	Não.		

As práticas da área de processo com foco no processo da organização não são consideradas atendidas, pois essa área de processo está relacionada com a implementação de processos padrão na organização. Tais práticas são realizadas nesse processo, mas, para atender às suas necessidades individualmente, que requer idade e adaptação de acordo com o componente a ser produzido, o foco que essa área de processo exige é de nível organizacional, sendo assim, inviável de atender.

7.4.3.7 Definição do processo da organização.

Essa área de processo é atendida pelo processo proposto, definindo e segundo diretrizes pertinentes ao projeto de forma individual. Suas atividades são apresentadas na Tabela 7.15.

Tabela 7.15 – Definição do processo da organização

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer processos padrão.	Não.		
SP1.2	Estabelecer descrições de modelos de ciclo de vida.	Planejamento.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia.
SP1.3	Estabelecer critérios e diretrizes para adaptação.	Planejamento.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia.
SP1.4	Estabelecer o repositório de medições da organização.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP1.5	Estabelecer a biblioteca de ativos de processo da organização.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP1.6	Estabelecer padrões de ambiente de trabalho.	Controlar configuração, planejamento.	Configuração - Apêndice G, plano de desenvolvimento - Apêndice C.	Comum, gerência de requisitos.
SP1.7	Estabelecer regras e orientações para equipes.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.

- **SP1.1:** essa prática não é considerada atendida, pois o processo oferece flexibilidade para sua adaptação de acordo com a necessidade do componente a ser produzido, inviabilizando assim a cobertura da prática.

- **SP1.2:** essa prática é atendida na realização da atividade de planejamento do projeto, com a adaptação do projeto, de acordo com o componente a ser produzido.
- **SP1.3:** o atendimento dessa prática se dá com a realização da atividade de planejamento do projeto, com a adaptação do projeto de acordo com o componente a ser produzido.
- **SP1.4:** o atendimento dessa prática se dá com a realização da atividade de controle de projeto ao longo do ciclo de vida, de acordo com o planejado.
- **SP1.5:** o atendimento dessa prática se dá com a realização e definição dos itens a serem controlados pela atividade de controlar configuração.
- **SP1.6:** o atendimento dessa prática se dá com a definição do ambiente de configuração pelas atividades de controlar configuração e planejamento.
- **SP1.7:** o atendimento dessa prática se dá com a realização das atividades de planejamento.

7.4.3.8 Treinamento organizacional.

Essa área de processo é parcialmente atendida, uma vez que no processo proposto são definidas estratégias de capacitação e treinamento dos envolvidos, de acordo com o necessário para atender ao projeto, porém não define diretrizes organizacionais. Suas atividades são apresentadas na Tabela 7.16.

Tabela 7.16 – Treinamento organizacional

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer necessidades estratégicas de treinamento.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.2	Identificar as necessidades de treinamento sob responsabilidade da organização.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.3	Estabelecer um plano tático de treinamento na organização.	Não.		
SP1.4	Estabelecer a capacidade de treinamento.	Não.		
SP2.1	Fornecer treinamento.	Não.		
SP2.2	Estabelecer registros de treinamento.	Não.		
SP2.3	Avaliar a eficácia dos treinamentos.	Não.		

- **SP1.1 e SP1.2:** essas práticas são atendidas na realização da atividade de planejamento do projeto, com a alocação e a preparação da equipe.
- **SP1.3 a SP2.3:** as práticas na área de processo de treinamento organizacional devem ser realizadas de acordo com as diretrizes organizacionais, e o plano de treinamento indicado nesse processo se dá a partir das necessidades do projeto, de acordo com o componente a ser produzido, por isso tais práticas não são consideradas atendidas no contexto requerido pela área de processo.

7.4.3.9 Gerência integrada de projeto

Essa área é atendida parcialmente pelo processo, pois realiza atividades de integração e controle do projeto do componente individual, bem como de todo o sistema, ao longo do desenvolvimento do projeto. Assim, pode não atender como um todo, uma vez que não está articulada com processos da organização. Suas atividades são apresentadas na Tabela 7.17.

Tabela 7.17 – Gerência integrada de projeto

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer o processo definido para o projeto.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.2	Utilizar os ativos de processo organizacional para planejar as atividades do projeto.	Não.		
SP1.3	Estabelecer o ambiente de trabalho do projeto.	Não.		
SP1.4	Integrar planos.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.5	Gerenciar o projeto utilizando planos integrados.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.6	Estabelecer equipes.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.7	Contribuir para os ativos de processos organizacionais.	Controlar configuração.	Configuração - Apêndice G, plano de desenvolvimento - Apêndice C.	Comum.
SP2.1	Gerenciar o envolvimento das partes interessadas.	Planejamento, controle de projeto, homologação.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Gerenciar dependências.	Definição dos requisitos, definição dos componentes, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.3	Solucionar questões críticas de coordenação.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.

- **SP1.1:** o atendimento dessa prática se dá com a realização das atividades de planejamento do projeto, a partir do ciclo de vida definido para ele.

- **SP1.2:** essa prática não é interpretada como não atendida, pois no processo sugerido está relacionada ao projeto do componente, e não segue as diretrizes organizacionais.
- **SP1.3:** essa prática não é interpretada como não atendida, pois no processo sugerido está relacionada ao projeto do componente, e não segue as diretrizes organizacionais.
- **SP1.4:** essa prática é realizada com a atividade de planejamento do projeto.
- **SP1.5:** essa prática é realizada com a atividade de planejamento do projeto.
- **SP1.6:** essa prática é realizada com a atividade de planejamento do projeto.
- **SP1.7:** essa prática é realizada no controle da configuração, de acordo com o planejado.
- **SP2.1:** essa prática é realizada na atividade de homologação em controle do projeto.
- **SP2.2:** essa prática é realizada na atividade de homologação em controle do projeto.
- **SP2.3:** essa prática é realizada na atividade de homologação em controle do projeto.

7.4.3.10 Gerência de risco

Essa área é realizada pelo processo proposto, iniciando com seu planejamento e prosseguindo ao longo do projeto. Suas atividades são apresentadas na Tabela 7.18.

Tabela 7.18 – Gerência de risco

Código	Descrição	Atividades	Documento	Fase
SP1.1	Determinar fontes e categorias de riscos.	Planejamento Controle de Projeto	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Definir parâmetros para riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.3	Estabelecer uma estratégia para gestão de riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Identificar riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Avaliar, categorizar e priorizar riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.1	Elaborar planos de mitigação de riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.2	Implementar planos de mitigação de riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.

- **SP1.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP1.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP1.3:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP2.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP2.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP3.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP3.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
-

7.4.3.11 Análise de decisão e resolução

O processo proposto não oferece atividades específicas para a tomada de decisão, porém, de forma indireta, auxilia em sua execução. As atividades dessa área de processo estão apresentadas na Tabela 7.19.

Tabela 7.19 – Análise de decisão e resolução

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer diretrizes para análise e decisão.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.2	Estabelecer critérios de avaliação.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.3	Identificar soluções alternativas.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.4	Selecionar métodos de avaliação.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.5	Avaliar soluções alternativas.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.6	Selecionar soluções.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.

- **SP1.1, SP1.2, SP1.3, SP1.4, SP1.5 e SP1.6:** as práticas dessa área de processo são descritas no planejamento do projeto, realizadas durante o andamento do projeto na atividade de controle de projeto e finalizadas durante a atividade de fechamento.

7.4.4 Específicas de nível 4

7.4.4.1 Desempenho do processo organizacional

O processo proposto em seu planejamento, pela natureza da realização das atividades, atende parcialmente a essa área de processo, pois as boas práticas resultam melhor desempenho do processo. Contudo, seu foco é o desempenho do produto. Suas atividades são apresentadas na Tabela 7.20.

Tabela 7.20 – Desempenho do processo organizacional

Código	Descrição		Documento	Fase
SP1.1	Estabelecer objetivos para qualidade e para desempenho de processo.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Selecionar processos.	Não.		
SP1.3	Estabelecer medidas de desempenho de processo.	Não.		
SP1.4	Analisar a performance do processo e estabelecer baselines de desempenho de processo.	Não.		
SP1.5	Estabelecer modelos de desempenho de processo.	Não.		

- **SP1.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante o andamento do projeto, no controle de projeto.
- **SP1.2 a SP1.5:** essas práticas específicas não são consideradas atendidas, pois a elas se referem as atividades organizacionais e as atividades de avaliação de desempenho. Nesse processo, referem-se parcialmente ao desempenho do processo e, mais profundamente, ao desempenho do componente produzido.

7.4.4.2 Gerência quantitativa de projeto

O processo proposto atende parcialmente à área de processo de gerência quantitativa do projeto, estabelecendo e definindo as atividades de gestão. Suas atividades são apresentadas na Tabela 7.21.

Tabela 7.21 – Gerência quantitativa de projeto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Estabelecer os objetivos do projeto.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Compor o processo definido.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.3	Selecionar subprocessos e atributos.	Não.		
SP1.4	Selecionar medidas e técnicas analíticas.	Não.		
SP2.1	Monitorar o desempenho dos subprocessos selecionados.	Não.		
SP2.2	Gerenciar o desempenho do projeto.	Não.		
SP2.3	Realizar a análise de causa raiz.	Não.		

- **SP1.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante todo o andamento do projeto, no controle de projeto.

- **SP1.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante todo o andamento do projeto, no controle de projeto.
- **SP1.3 a SP2.3:** essas práticas não são consideradas atendidas pelo processo, pois são relacionadas à medição de sua *performance*. Para isso, o processo deve estar definido e ser rigorosamente seguido, o que não é viável para sua flexibilidade, de acordo com o componente a ser produzido.

7.4.5 Específicas de nível 5

7.4.5.1 Análise causal e resolução

O processo atende a essa área desde o planejamento do projeto, indicando pontos de análise e recebendo acompanhamento em todas as fases. A finalização das atividades envolvidas nesse processo é feita na avaliação do ciclo. Suas atividades são apresentadas na Tabela 7.22.

Tabela 7.22 – Análise causal e resolução

Código	Descrição	Atividade	Documento	Fase
SP1.1	Selecionar resultados para análise	Controle de projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Analisar causas.	Controle de projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Implementar propostas de ação.	Controle de projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Avaliar efeitos de ações implementadas.	Não.		
SP2.3	Registrar dados.	Fechamento.	Projeto.	Avaliação do ciclo.

- **SP1.1 a SP2.1:** essas práticas são realizadas durante o controle do projeto, de acordo com o planejamento da homologação.
- **SP2.2:** essa prática específica é considerada não realizada, pois não é uma ação explícita do processo. Contudo, ela pode ser incluída no planejamento do projeto a fim de ser realizada.
- **SP2.3:** essa prática específica é realizada na fase de avaliação do ciclo com a atividade de fechamento.

7.4.5.2 Gestão do processo organizacional

Na fase de avaliação do ciclo de vida do projeto, em especial na atividade de fechamento do processo, são realizadas as atividades de avaliação do desempenho desse ciclo, buscando

entender os resultados alcançados e reestruturar novos ciclos com base nos sucessos e fracassos. Essas atividades são apresentadas na Tabela 7.23.

Tabela 7.23 – Gestão do processo organizacional

Código	Descrição	Atividade	Documento	Fase
SP1.1	Manter objetivos de negócio.	Não.		
SP1.2	Analisar dados de desempenho do processo.	Fechamento.	Projeto.	Avaliação do ciclo.
SP1.3	Identificar áreas com potencial para melhorias.	Fechamento.	Projeto.	Avaliação do ciclo.
SP2.1	Identificar melhorias sugeridas.	Fechamento.	Projeto.	Avaliação do ciclo.
SP2.2	Analisar sugestões de melhorias.	Não.		
SP2.3	Validar melhorias.	Fechamento.	Projeto.	Avaliação do ciclo.
SP2.4	Selecionar e aplicar melhorias para implantação.	Fechamento.	Projeto.	Avaliação do ciclo.
SP3.1	Planejar implantação.	Fechamento.	Projeto.	Avaliação do ciclo.
SP3.2	Gerenciar implantação.	Fechamento.	Projeto.	Avaliação do ciclo.
SP3.3	Avaliar os efeitos de melhorias.	Não.		

- **P1.1, SP2.2 e SP3.3:** essas são consideradas não atendidas pelo processo, pois não são realizadas ações que as atendam diretamente. Contudo, no processo é possível realizá-las se forem incluídas no planejamento.
- **SP1.2, SP1.3, SP2.1, SP2.3, SP2.4, SP3.1 e SP3.2:** essas práticas específicas são realizadas na avaliação do ciclo, na atividade de fechamento.

7.5 Estatística da cobertura do processo ao modelo CMMI-DEV

A elaboração do processo de desenvolvimento teve como objetivos o entendimento das áreas de processo do CMMI-DEV, a identificação dos objetivos de suas áreas de processo e o atendimento de suas práticas, de acordo com as principais necessidades de desenvolvimento de sistemas embarcados. Seu resultado é apresentado nas Figuras 7.31 e 7.32, que mostram, respectivamente, a cobertura por áreas de processo e a cobertura por nível.

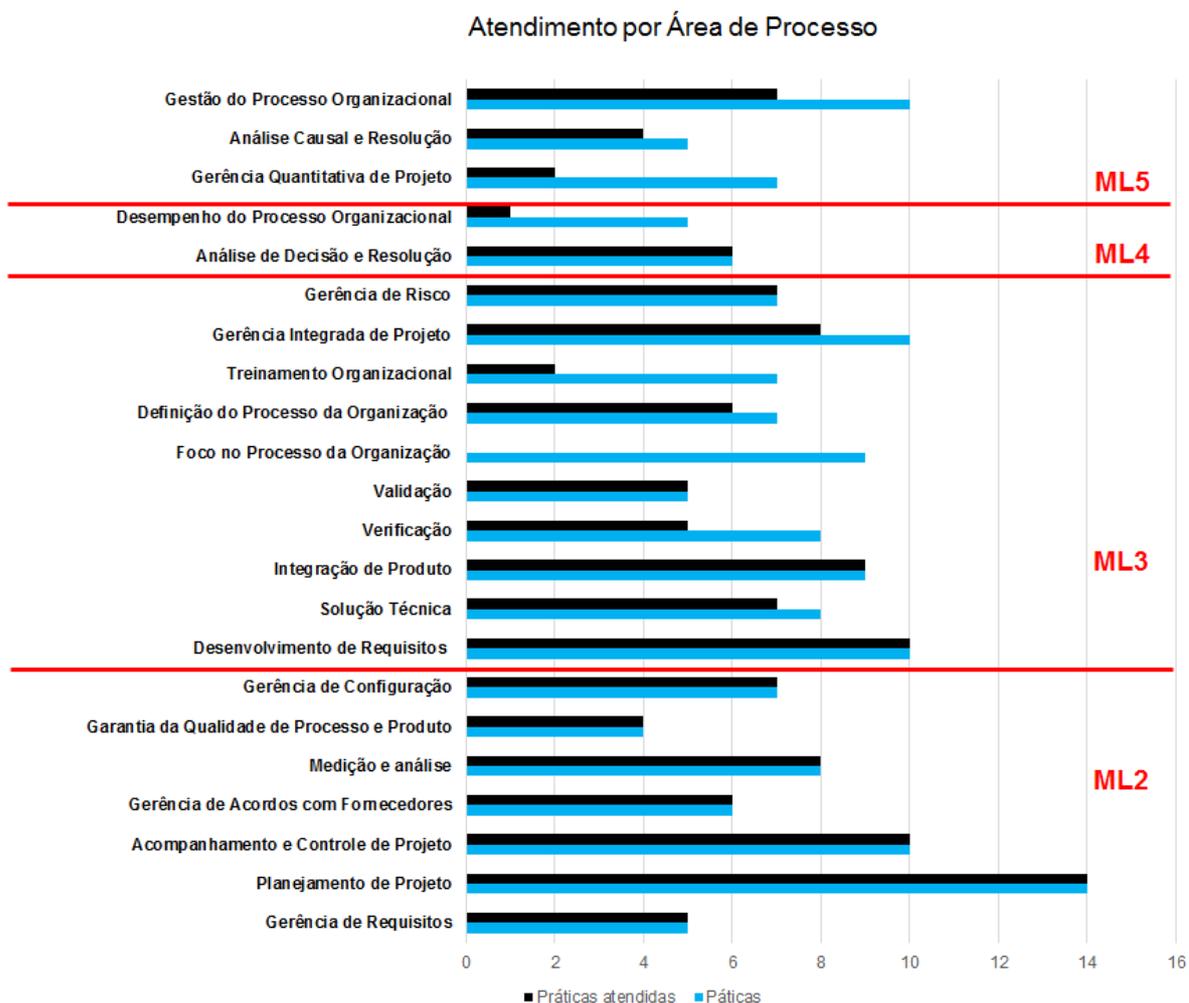


Figura 7.31 – Gráfico de atendimento por área de processo.

De forma geral, no processo proposto as práticas que não foram cobertas focam especificamente o desempenho do processo de desenvolvimento ou o desempenho organizacional, sem se preocupar com o desempenho do produto.

As diretrizes oferecidas por esse processo buscam aumentar a qualidade do componente de sistema embarcado, especialmente a parte *software*. Sendo assim, é difícil para o processo atender também aos requisitos de desempenho organizacional. Essa preocupação, em geral e especialmente na área espacial, é do projeto do produto, que segue modelos e padrões para o desenvolvimento de missões espaciais que já contemplam essas práticas. Quando essa situação existe, a união desse processo no contexto de tais modelos pode contribuir para que o projeto realizado atenda às demais práticas.

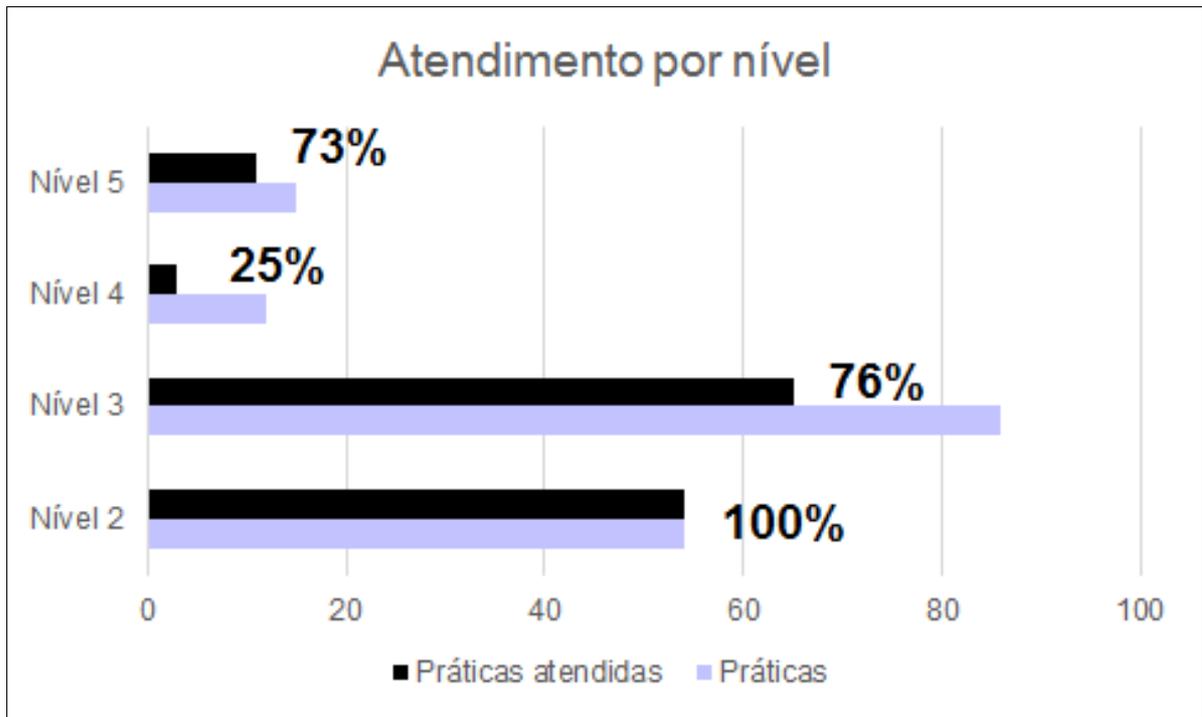


Figura 7.32 – Gráfico de atendimento por nível.

O nível 2 de maturidade do CMMI-DEV fornece diretrizes para o gerenciamento do processo. Todo processo, para ter sucesso, deve começar com o gerenciamento total do projeto. Como consequência, o processo atende a todas as práticas do nível 2 do CMMI-DEV, o que permite intitulá-lo “Uma abordagem para processo de desenvolvimento de sistema embarcado que atende ao nível 2 de maturidade do CMMI-DEV”.

7.6 Atendimento as normas ECSS

A realização das atividades das fases do processo apresentado e a consequente confecção de seus *templates* durante o projeto atendem às exigências do padrão ECSS apontados nesta tese. A Tabela 7.24 mostra a relação de atendimento do processo apresentado com os requisitos da norma ECSS-Q-80 para produtos de *software* espacial, e a Tabela 7.25 mostra a relação de atendimento do processo apresentado com os requisitos da norma ECSS-E-ST-40 para produtos de *software* espacial.

Tabela 7.24 – ECSS-Q-80 - Garantia de Produto Espacial / Garantia de Produto de *Software*

Implementação do programa de garantia do produto de Software		
5.1	Implementação do programa de garantia de produto Software	
5.2	Gestão de programa de garantia de produto de software	
5.3	A gestão de riscos e controle de item crítico	X
5.4	A seleção de fornecedores e controle	X
5.5	Aquisição	X
5.6	Ferramentas e ambiente de apoio	X
5.7	Processo de avaliação e melhoria	X
Garantia de processo de software		
6.1	Ciclo de vida de desenvolvimento Software	X
6.2	Requisitos aplicáveis a todos os processos de engenharia de software	X
6.3	Exigências aplicáveis aos processos de engenharia de software individuais ou atividades	X
Software de garantia de qualidade do produto		
7.1	Objetivos de qualidade do produto e do sistema métrico	X
7.2	Requisitos de qualidade do produto	X
7.3	Software destinado a reutilização	X
7.4	Hardware terreno padrão e serviços para o sistema operacional	
7.5	Firmware	

Tabela 7.25 – ECSS-E-ST-40- Engenharia de Software Espacial

Software relacionado com os requisitos do sistema		Processo de validação de Software			
5.2.2	Análise de requisitos do sistema de software relacionados	x	5.6.2 Processo de Validação	x	
5.2.3	Verificação do sistema de software relacionado	x	5.6.3 Validação técnica	x	
5.2.4	Integração do sistema de software relacionado e controle	x	5.6.4 Validação requisitos	x	
5.2.5	Exigência sistema de revisão	x	Processo de aceitação e entrega de software		
Processo de gestão de software		5.7.2	Entrega e Instalação de Software		
5.3.2	Software de gerenciamento de ciclo de vida		5.7.3	Aceitação de Software	x
5.3.3	Processo de revisão conjunta		Verificação de Software		
5.3.4	Descrição revisão do projeto de Software	x	5.8.2	Implementação do processo de Verificação	x
5.3.5	Descrição revisões técnicas de Software	x	5.8.3	Atividades de verificação	x
5.3.6	Revisão gradual	x	Processo de operação de software		
5.3.7	Gestão da interface	x	5.9.2	implementação do processo	
5.3.8	Orçamento técnica e de gestão de margem	x	5.9.3	Testes de operação	x
Requisitos de software e processo de engenharia de arquitetura		5.9.4	Suporte à operação de software		
5.4.2	Análise de requisitos de software	x	5.9.5	Suporte ao Usuário	
5.4.3	Software architectural design	x	Processo de manutenção de software		
5.4.4	PDR	x	5.10.2	Implementação do processo	
Design de software e processo de engenharia de aplicação		5.10.3	Análise de problemas e modificação		
5.5.2	Projeto de itens de software	x	5.10.4	Implementação de modificação	
5.5.3	Codificação e testes	x	5.10.5	Realizar revisões de manutenção	
5.5.4	Integração	x	5.10.6	Migração de software	
			5.10.7	Aposentadoria de software	

Como já foi visto em capítulos anteriores, tanto o padrão ECSS quanto o modelo CMMI requerem diretrizes e resultados a serem alcançados, mas não apresentam claramente as atividades que devem ser realizadas para que se atinjam estes objetivos.

O processo apresentado nesta tese complementa o padrão ECSS, indicando, de forma clara e com diretrizes efetivas, as atividades que devem ser realizadas para atender a tais requisitos. Sua relação está apresentada na Figura 7.33.

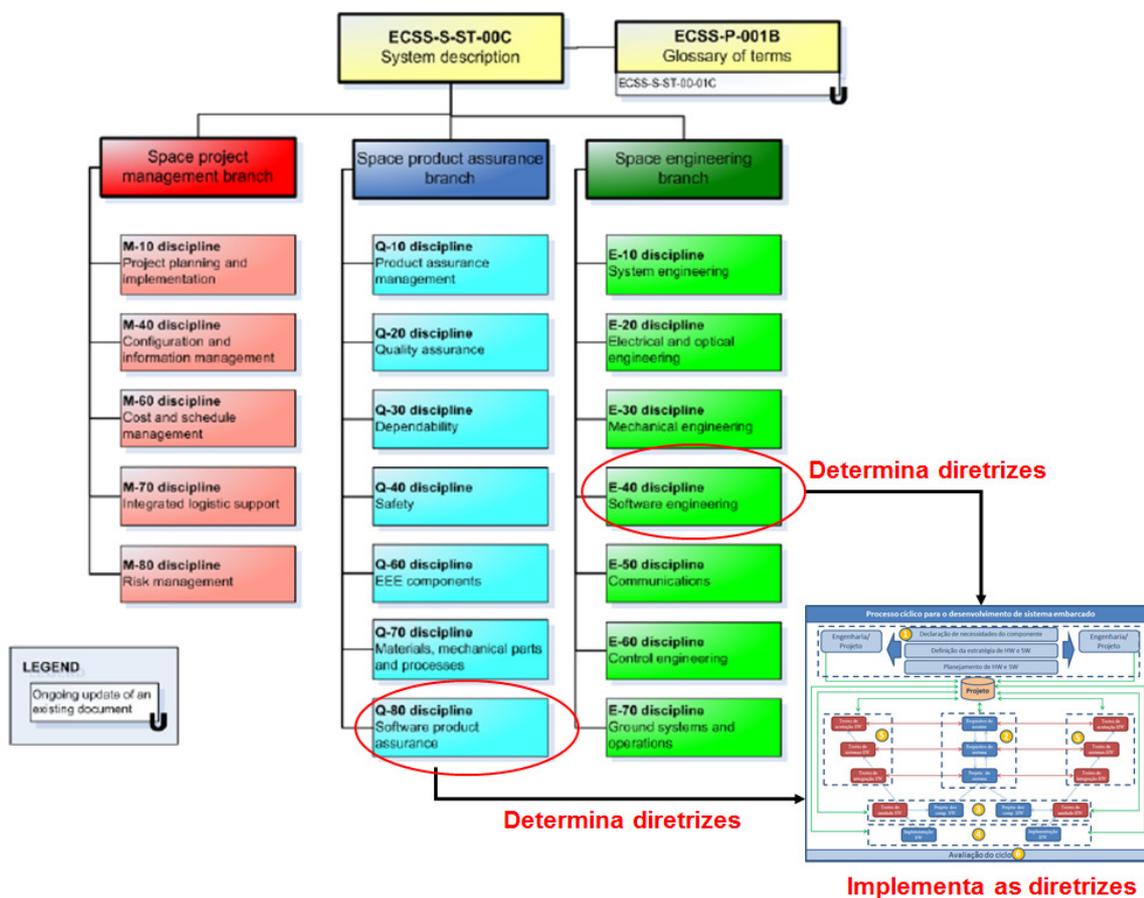


Figura 7.33 – Relação ECSS e Processo Cíclico.

Fonte: adaptado de ECSS (2010)

O padrão ECSS e o modelo CMMI apresentam semelhanças, uma vez que ambos têm bases sólidas na norma ISO/IEC 15504, o que torna viável para o processo cíclico o atendimento de seus requisitos.

7.7 Estudo de caso

Os estudos de caso desta tese estão sendo realizados da seguinte forma:

O estudo de caso (a) tem os seguintes objetivos:

- i. Avaliar a qualidade dos *templates* do processo.
 - ii. Avaliar o uso dos *templates* do processo desassociado das atividades do processo (boas práticas).
 - iii. Avaliar a qualidade e a completude da documentação produzida no estudo de caso em relação à documentação original.
- a) Elaborar a documentação do projeto “*Software* de controle para o conjunto de medida de velocidade de satélites”, realizado pelo laboratório de *hardware* e *software* da Divisão de Desenvolvimento de Sistemas de Solo (DSS) da Coordenação de Engenharia e Tecnologia Espaciais (ETE) do INPE.

Este estudo de caso se deu a partir de um projeto de desenvolvimento de um *software* embarcado já concluído; toda a documentação desse produto foi refeita utilizando os modelos de documento do processo apresentado por esse estudo. Foi avaliada a diferença entre os documentos produzidos pelo desenvolvimento *ad hoc* (original) e os documentos resultantes do processo, obtendo-se a seguinte conclusão:

- A documentação original do projeto não estava padronizada.
- Os documentos originais do projeto não foram gerados por meio de procedimentos planejados, e o resultado não é suficiente para refletir o produto desenvolvido.
- Os documentos originais não contêm informações completas necessárias para desenvolver o componente. Em caso de novo desenvolvimento desse mesmo produto, será necessária a realização de boa parte das atividades novamente (retrabalho), como levantamento, especificações, modelagem, entre outras ações necessárias para seu entendimento e desenvolvimento.

- A nova documentação produzida utilizou os *templates* apresentados nesta tese, porém, sua fonte de informação e de dados foi a documentação original (incompleta). Sendo assim, os novos documentos também não ficaram completos, tornando-se ineficiente. Isso mostra que a atividade de preencher simplesmente os documentos poderá não garantir a qualidade desejada. Os documentos produzidos em um projeto de desenvolvimento de *software* devem ser resultado de atividades que norteiam as boas práticas, seguindo um processo definido (ver Anexo A).

Os estudos de caso (b, c, e d) têm os seguintes objetivos:

- i. Avaliar a utilização do processo para desenvolver as partes *hardware* e *software* do componente.
 - ii. Avaliar a qualidade dos *templates* do processo.
 - iii. Avaliar o uso dos *templates* do processo associado às atividades do processo, seguindo as diretrizes e realizando as atividades do processo (boas práticas).
 - iv. Avaliar a qualidade e completude da documentação produzida no estudo de caso.
 - v. Avaliar a viabilidade do uso do processo com relação à facilidade de uso, entendimento e aplicabilidade.
- b) Utilizar essa documentação no projeto Unidade de Teste e Verificação (UTV) para testes, validação dos equipamentos da família Procod, Processador de Coleta de Dados, sistema utilizado nas estações terrenas, que está sendo realizado no laboratório de *hardware* e *software* da Divisão de Desenvolvimento de Sistemas de Solo (DSS) da Coordenação de Engenharia e Tecnologia Espaciais (ETE) do INPE.
- Este projeto está em fase inicial e utilizará o processo conforme descrito nesta tese.
- c) Utilizar essa documentação no Instituto Mauá de Tecnologia, Núcleo de Sistemas Eletrônicos Embarcados (NSEE), no desenvolvimento de um sistema de acionamento para motores sem escovas (*brushless*). O motor utilizado como motivação de estudos

terá como diretrizes as condições de contorno impostas pela Plataforma Multimissão (PMM) do INPE para rodas de reação – Projeto Citar.

- Este projeto está em andamento e utilizando o processo conforme descrito nesta tese.
 - Os documentos deste projeto estão sendo produzidos por meio do sistema LATEX, gerando atividades excedentes para os desenvolvedores, pois, originalmente, os *templates* do processo foram elaborados em Word (ver Anexo B).
 - Foram necessárias adaptações (inclusões) de itens nos *templates* utilizados para atender ao projeto.
 - Até o momento, a utilização do processo auxiliou os desenvolvedores nas atividades realizadas, além de oferecer mecanismos para o planejamento e o controle do projeto.
 - O processo apresentado até o momento foi de simples entendimento e utilização.
- d) Utilização no Instituto Mauá de Tecnologia, Núcleo de Sistemas Eletrônicos Embarcados (NSEE), no *Planetary Transits and Oscillations of Stars (PLATO)* Observatório espacial da Agência Espacial Europeia (Fase de verificação da viabilidade - ver Anexo C).
- O processo apresentado foi avaliado e autorizado para uso no projeto pelo *Plato Consortium*.

8 CONSIDERAÇÕES FINAIS

Esta tese apresentou um processo de desenvolvimento de sistemas embarcados para viabilizar o desenvolvimento desses produtos com a mesma qualidade de outras categorias de sistemas, ou seja, um processo que acompanhe o componente embarcado em todas as fases de desenvolvimento, com ciclo de vida completo, atividades definidas em todas as fases e *templates* a serem utilizados. Este processo deve direcionar o desenvolvimento de sistemas para a realização de boas práticas, de maneira que os documentos produzidos durante a execução do projeto sejam efetivamente resultado da aplicação dessas práticas, e não de uma atividade de documentação de sistemas burocrática e inaplicável.

O uso do CMMI-DEV para nortear o processo ofereceu maior segurança para sua estrutura, pois todo o modelo requer que determinadas atividades sejam executadas. Uma vez atendidas essas normas, o produto gerado alcança maior garantia de qualidade, pois foi concebido com um conjunto de atividades planejadas e aderentes aos padrões internacionais.

8.1 Trabalhos correlatos e o processo apresentado

O diferencial dessa tese para os trabalhos correlatos é a indicação do foco adotado por seus realizadores. Esses trabalhos podem ser utilizados como referência na realização de alguma fase do processo em desenvolvimento nesta tese, além de serem complementares entre si.

A Figura 8.1 apresenta a relação entre o processo resultado desta tese e os trabalhos correlatos, referenciados pela letra indicada na seção de apresentação desses trabalhos e na tabela em que eles são sintetizados. Cada trabalho correlato foi distribuído/alocado em apenas uma fase do processo, de acordo com a característica de maior representação. O retângulo externo representa esta tese; o retângulo à esquerda refere-se ao trabalho que exerceu influência sobre esta tese; e os retângulos menores representam o posicionamento dos trabalhos correlatos com esta proposta.

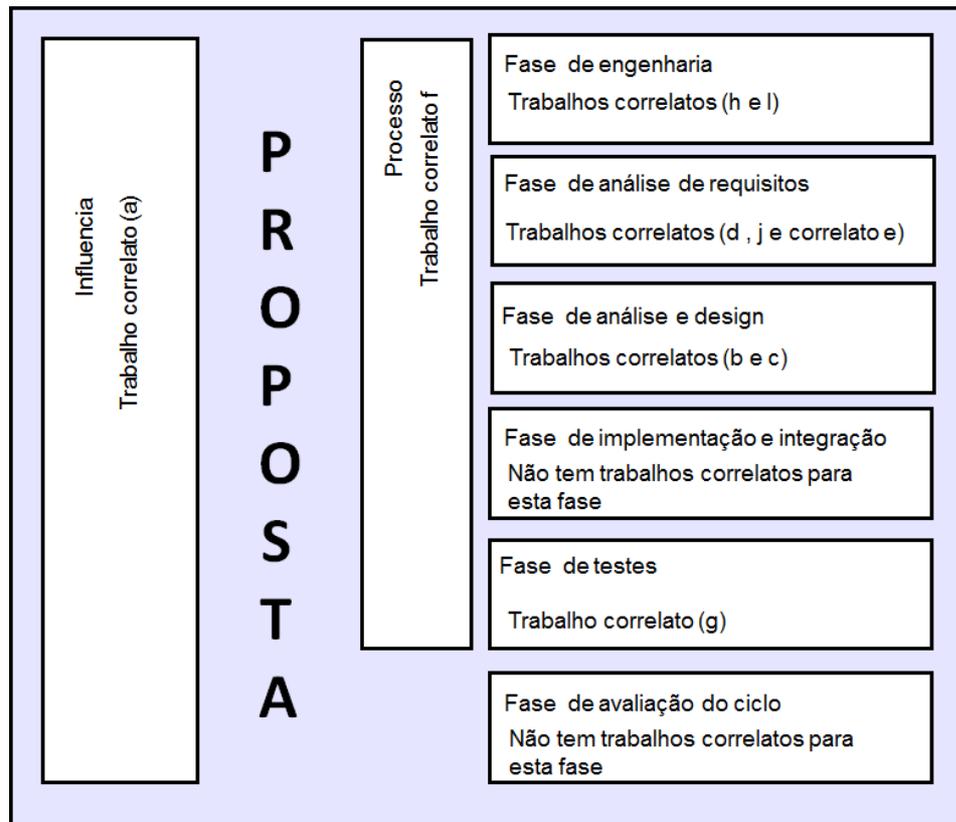


Figura 8.1 – Representação da relação do processo desta tese com os trabalhos correlatos.

8.2 Objetivos da tese x resultados apresentados

O estudo de processos para o desenvolvimento de sistemas embarcados é vasto. As soluções apresentadas por esta tese buscam oferecer diretrizes a serem seguidas em um processo completo, mas que podem motivar novas pesquisas segmentadas em pontos específicos aqui apontados.

8.2.1 Com relação ao objetivo geral

- i. Os principais objetivos desta tese são identificar mecanismos e propor um processo flexível para o desenvolvimento de um sistema embarcado que atenda às necessidades particulares e aos aspectos individuais de *software* e *hardware*.

Solução - Nesta tese foi idealizado um processo de desenvolvimento de sistema embarcado particionado para o desenvolvimento da parte *hardware* e da parte *software* do componente embarcado, porém de forma compartilhada em todas as suas fases.

Limitação Foram apresentados, para cada fase, fluxos de todas as atividades que devem ser realizadas, porém elas não apontam técnicas e ferramentas a serem utilizadas que possam aumentar a eficiência do processo. Estudos que abordem cada fase de forma independente, bem como técnicas e ferramentas, podem contribuir para aumentar a qualidade dos componentes desenvolvidos.

8.2.2 Com relação aos objetivos específicos

- i. Definir e caracterizar conceitualmente sistemas, componentes, *software* e *hardware* embarcados.

Solução - Por meio da revisão bibliográfica foram apresentados no Capítulo 2 conceitos encontrados na literatura com diferentes aspectos e fatores entre si. O Capítulo 5 foi dedicado à caracterização de sistemas embarcados a serem explorados nesta tese.

Limitação - As diferentes abordagens encontradas na literatura dificultaram uma definição precisa e comum, mas foram suficientes para os objetivos deste estudo. É importante e necessária a continuidade de estudos com esse objetivo, pois a evolução dos componentes embarcados e de seu uso poderá estimular novas pesquisas.

- ii. Definir as características e fronteiras de um processo para o desenvolvimento de sistemas embarcados.

Solução - Foram apresentados, no Capítulo 5, a relação de cada parte de um sistema embarcado, as características do ambiente em que está inserido, os fluxos para o projeto de um sistema embarcado, ciclos de vida e fatores que devem ser considerados em sistemas embarcados.

Limitação - Pela própria concepção dos sistemas embarcados, é difícil definir uma fronteira que identifique as atividades relacionadas a cada parte; limiares sutis podem refletir grandes diferenças. É importante a realização de estudos aprofundados com esse objetivo.

- iii. Definir um processo de desenvolvimento de sistemas embarcados que ofereça atividades individualizadas para as partes *software* e *hardware* dos componentes do sistema embarcado.

Solução - No Capítulo 7 foi apresentado o resultado do esforço principal desta tese, que foi elaborar diretrizes para o desenvolvimento do sistema embarcado, com diferenciações significativas entre as atividades da parte *hardware* e da parte *software* do componente, ao mesmo tempo mantendo a unicidade entre elas, já que se tornarão uma só quando integradas.

Limitação O processo apresenta, de forma geral, as atividades individualizadas que devem ser detalhadas em cada ciclo de desenvolvimento, de acordo com a especificidade do componente a ser construído.

- iv. Definir um processo que particione os componentes dos sistemas embarcado para que sejam produzidos individualmente.

Solução - Conforme apresentado no Capítulo 7, neste estudo foi elaborado um processo cíclico organizado em fases, com atividades independentes e compartilhadas entre o *software* e o *hardware* do componente embarcado. Cada ciclo de desenvolvimento é a realização sequencial de todas as fases e atividades do ciclo de vida, que terá como resultado um componente de um sistema embarcado e todos os documentos produzidos durante seu desenvolvimento.

Limitação - O processo apresentado oferece diretrizes e atividades que devem ser realizadas, mas não oferecem ferramentas que automatizem essas atividades. Estudos para o desenvolvimento de uma ferramenta que apoie todo o processo de desenvolvimento, oferecendo também mecanismos de adaptação de controles automatizados, poderão contribuir ainda mais para o aumento da qualidade dos componentes embarcados.

- v. Definir um processo que atenda aos modelos de qualidade CMMI e ao modelo MPS.Br.

Solução - Este trabalho apresentou a construção de um processo para o desenvolvimento de sistemas embarcados simplificados, de fácil utilização e que atendam às exigências do modelo CMMI. A ideia é direcionar os desenvolvedores a boas práticas que permitam o uso de mecanismos para o atendimento de atividades específicas para cada tipo de componente a ser desenvolvido e abrir caminho para o aprofundamento e novos estudos envolvendo não só sistemas embarcados, mas a engenharia de *software* como um todo, pois as dificuldades de utilização de modelos de qualidade não são exclusividade dos sistemas embarcados.

Limitação - Algumas práticas a serem realizadas, em especial as práticas genéricas, para serem atendidas exigem envolvimento organizacional para que ela seja institucionalizada.

- vi. Definir um processo que seja absorvido pelas normas ECSS-E-ST-40C e ECSS-Q-ST-80 do padrão ECSS.

Solução - Foram apresentados, no Capítulo 4, a organização de uma missão espacial e os requisitos para o desenvolvimento de *software* na área especial segundo o modelo ECSS, e na seção 7.6 o processo apresentado foi relacionado com as normas do padrão ECSS.

Limitação - O processo atende aos requisitos referentes ao desenvolvimento do componente de forma individual; não se olhou para o programa como um todo, pois não era esse o objetivo da tese.

- vii. Definir mecanismos de modelagem prévia do componente, controle e integração dos componentes de *software* e de *hardware* do sistema embarcado para minimizar problemas após a implementação.

Solução - A fase de análise e projeto do processo apresentado no Capítulo 7 indica atividades a serem realizadas e artefatos a serem produzidos que conduzem o desenvolvedor a elaborar e validar protótipos das partes *software* e *hardware* do componente.

Limitação - O processo induz o desenvolvedor à realização das atividades de validação dos modelos, mas não apresenta técnicas e ferramentas para auxiliar nessas atividades. Devem ser realizados estudos aprofundados para essa atividade, o que, atualmente, é o maior desafio para a engenharia de *software*, além de oferecer mecanismos práticos e eficientes para a construção e a validação dos modelos dos componentes embarcados antes de sua implementação.

- viii. Definir um processo adaptável para o desenvolvimento de componentes para sistemas embarcados em que cada componente do sistema seja produzido em um ciclo independente e cada execução do ciclo possa adaptar o processo às características do novo componente a ser produzido.

Solução - No processo apresentado no Capítulo 7, cada ciclo deve estabelecer as relações de trabalho, definir as atividades a serem realizadas durante o projeto que atendam aos requisitos do componente a ser desenvolvido, definir e distribuir os papéis aos membros da equipe, bem como definir metas, estratégias e planos de trabalho para todas as fases do processo. Devemos lembrar que um componente é composto pelas partes *software* e *hardware*.

Para cada novo componente a ser desenvolvido, deve realizar modificações no processo, de acordo com as características do componente e com o resultado da avaliação dos ciclos anteriores. Quando um sistema embarcado for composto por diversos componentes, deverá ter diferentes ciclos de desenvolvimento, sendo um para cada componente de sistemas embarcado.

Limitação - O processo apresentado induz à adaptação do processo a cada ciclo, mas não oferece diretrizes para a adaptação. Devem ser realizados estudos aprofundados que ofereçam mecanismos de adaptação para o processo de desenvolvimento de sistemas embarcados, tornando mais fácil sua adaptação.

- ix. Definir um processo no qual os documentos produzidos sejam resultado das boas práticas das atividades realizadas em todo o processo.

Solução - A estrutura do processo apresentado no Capítulo 6 direciona o desenvolvedor a produzir os documentos de acordo com as atividades realizadas, alimentados ao longo do

projeto e utilizados em todas as fases. Os documentos utilizados foram adaptados dos *templates* do RUP.

Limitação - O processo não inibe a produção os documentos após o encerramento do projeto, assim como todas as diretrizes apontadas por este trabalho. Esse progresso fica a cargo do planejamento e do acompanhamento do projeto.

- x. Aplicar os resultados da pesquisa em casos reais.

Solução – O processo proposto está sendo utilizado em quatro projetos de desenvolvimento de sistemas embarcados.

Limitação – As análises apresentadas são de projetos em andamento, portanto parciais. Seu uso está associado a projetos de instituição de ensino e de pesquisa que fazem parte de um contexto maior. O progresso do estudo de caso deve estar em sintonia com todo o programa ao qual está subordinado.

8.3 Contribuição acadêmica

O texto desta tese foi produzido de forma que possa servir de material didático nas disciplinas Engenharia de Software e Qualidade de Software, em especial com foco em sistemas embarcados.

O Capítulo 2 e o Capítulo 5 apresentam conceitos básicos fundamentais para a disciplina de Introdução a Computação.

O Capítulo 3 pode ser utilizado na disciplina Qualidade de Software, pois apresenta, de forma simplificada, o modelo CMMI-DEV e todas as áreas de processo, bem como sua aderência ao MPS.Br.

O Capítulo 7 pode ser utilizado nas disciplinas Engenharia de Software e Qualidade de Software.

O Apêndice B é um processo de desenvolvimento de *software* completo para ser utilizado como modelo para as disciplinas Engenharia de *Software* e Qualidade de *Software*, bem como pelas empresas que buscam implementar processos de *software*.

8.4 Conclusão

Uma das maiores dificuldades para a comunidade de desenvolvimento de sistemas embarcados é a inviabilidade do uso de modelos de qualidade em seus projetos. Isso ocorre pelo fato de tais modelos, em geral, buscarem apoio para o processo, e não para o produto final. Além disso, os componentes embarcados exigem condições especiais de desenvolvimento e de uso, o que limita o apoio dos desenvolvedores.

Os desenvolvedores de sistemas embarcados ou componentes embarcados buscam alternativas para aumentar a qualidade de seus produtos, bem como para reduzir custos e o tempo de desenvolvimento.

Os principais itens desta tese que podem auxiliar a resolver essa situação são:

- Apresentação de um ciclo de vida completo para o desenvolvimento de sistemas embarcados.
- Especificação de todas as fases dos ciclos de vida, apresentando atividades, documentos e resultados esperados no desenvolvimento de cada componente de um sistema embarcado.
- Disponibilização para a comunidade de pesquisa de toda a documentação de um modelo de processo de desenvolvimento de sistemas embarcados com modelos de documentos que auxiliem na especificação individual dos componentes, viabilizando, assim, seu reúso, além de facilitar a avaliação de utilização de componentes externos.
- Utilização de modelos de qualidade de *software* para um processo de desenvolvimento de um produto específico.
- Viabilização e documentação de diferentes modelos para o componente produzido antes de sua implantação, reduzindo assim o custo com descarte de itens utilizados em sua produção.

Embora, não seja o objeto de estudo desta tese e, portanto, seja classificada como uma contribuição secundária, porém muito importante para a pesquisa em engenharia de *software* e para as empresas de desenvolvimento de *software*, a contribuição da engenharia de *software* é viabilizar o uso de processos de *software* por pequenas empresas ou pequenos produtos.

Com relação à adoção do processo de desenvolvimento de *software* por pequenas empresas ou no desenvolvimento de pequenos produtos de *software*, uma vez que muitas características de tal desenvolvimento se assemelham com as do desenvolvimento de sistemas embarcados, principalmente no que se refere a tamanho e diretrizes organizacionais, bem como à resistência no uso de modelos de qualidade por seus desenvolvedores, acredita-se que ele torna o desenvolvimento do projeto mais burocrático e difícil de ser realizado.

Os principais itens desta tese que podem auxiliar nesses projetos são:

- Melhor entendimento do modelo CMMI-DEV e, conseqüentemente, do modelo MPSBr, permitindo nova cultura de uso de modelos de qualidade e demonstrando que são as interpretações enganosas, mais do que os modelos, que burocratizam os processos de desenvolvimento de *software*. Esta tese apresenta, de forma simplificada, as práticas do CMMI.
- Conhecimento da aderência entre o CMMI-DEV e o MPSBr e estímulo para a busca de incentivos governamentais para a implantação do modelo MPSBr e, como consequência, do CMMI-DEV.
- Disponibilização para a comunidade de pesquisa de toda documentação de um modelo de processo de desenvolvimento de sistemas, com a definição e o detalhamento do ciclo de vida, de todas as fases, atividades, documentos e resultados que devem ser esperados por um processo, sendo este processo de pesquisa científica e aprovado por uma banca composta por pesquisadores.

As diretrizes sugeridas por este processo de desenvolvimento de sistemas embarcados procuram atingir os seguintes resultados:

- Aumentar a qualidade dos componentes embarcados.
- Aumentar a qualidade dos produtos de *software*.

- Estimular o uso de processos para o desenvolvimento de *software*.

8.5 Publicações

Como resultado da pesquisa que resultou nesta tese, foram publicados e/ou apresentados em conferências de engenharia e sistemas embarcados os seguintes artigos científicos:

MIYASHIRO, M. A. S.; FERREIRA, M.G.V.; SANNT'ANA, N. Viability of Application of the Process Areas of CMMI-DEV Processes Development of Critical Embedded Systems. Science and Information Conference - Londres. Aprovado para publicação e apresentação na edição 2015 do congresso.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. One approach to the use of the practices of CMMI-DEV V1.3 level 2 in a process of development of Embedded Systems. The Fifth International Conference on Information, Intelligence, Systems, and Applications - Grécia, 2014.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. Process for the development of Embedded System Following the Practices of CMMI Level 2 - Science and Information Conference - Londres, 2014.

O trabalho apresentado nesta conferência despertou o interesse da University of Huddersfield - School of Computing and Engineering (Electronic and Electrical Engineering) Dr. Violeta Holmes, HPC Research Group Leader, para uso do processo no desenvolvimento de sistema embarcado para a indústria automobilística, a ser realizado por um aluno de seu grupo de pesquisa, o que está em análise de viabilidade pelos envolvidos.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. The Practices of Level 2 (Managed) CMMI-DEV V1.3 in Development of Embedded Systems - International Conference on Electrical and Electronic Engineering - Hong Kong, 2014.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. Phase Cyclical Process Requirements for the Development Of Embedded Systems - IEEE PUBLICATION TITLE (Journal, Magazine,

Conference, Book): 2013 III Brazilian Symposium on Computing Systems Engineering - Brasil.

8.6 Trabalhos futuros

É crescente o direcionamento dos esforços de pesquisa para o desenvolvimento de sistemas embarcados. Muitos trabalhos estão sendo desenvolvidos e, a partir da pesquisa que embasou esta tese, outros trabalhos podem ser elaborados para dar continuidade ao trabalho aqui apresentado. Algumas sugestões de temas são:

- Estudo de técnicas e ferramentas para o desenvolvimento de sistemas embarcados que possam ser utilizadas em cada fase de um processo.
- Implementação de uma ferramenta com todas as fases, atividades e documentos do processo com controles, checagem de resultados dos testes e portátil com diferentes ferramentas que possam dar suporte à execução do processo.
- Elaboração de diretrizes para adaptação do processo de sistemas embarcados de acordo com o componente a ser desenvolvido.
- Estudo de mecanismos eficientes para auxiliar na modelagem e validação dos componentes em fases anteriores à implementação.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). NBR ISO 8402 – *Gestão da qualidade e garantia da qualidade*. Rio de Janeiro: ABNT, 1994.

_____. *NBR ISO 9000* – sistemas de gestão da qualidade - fundamentos e vocabulário. Rio de Janeiro: ABNT, 2005.

_____. *NBR ISO/IEC 12119* – tecnologia de informação – pacotes de *software* – testes e requisitos de qualidade. Rio de Janeiro: ABNT, 1998.

_____. *NBR ISO/IEC 14598*- tecnologia de informação – avaliação de produto de *software* – visão geral. Rio de Janeiro: ABNT, 2001.

_____. *NBR ISO/IEC 15504*- tecnologia da informação – avaliação de processo. Rio de Janeiro: ABNT, 2008.

_____. *NBR ISO/IEC 25051* - engenharia de *software* – requisitos e avaliação da qualidade de produto de *software* (SQuaRE) – requisitos de qualidade de produto de *software* comercial de prateleira (COTS) e instruções para teste. Rio de Janeiro: ABNT, 2008.

_____. *NBR ISO/IEC 25030* - engenharia de *software* – requisitos e avaliação da qualidade de produto de *software* (SQuaRE) – requisitos de qualidade. Rio de Janeiro: ABNT, 2008.

_____. *NBR ISO/IEC 9126* - tecnologia da informação qualidade de produto de *software*. Rio de Janeiro: ABNT, 2003.

Accenture. *Embedded software for the electronics and high-tech industry*. 2006. Disponível em: <www.accenture.com/SiteCollectionDocuments/PDF/EmbeddedSoftware2.pdf>. Acesso em: fev. 2014.

ALBUQUERQUE, I. S. *Modelo para o gerenciamento da configuração e gerenciamento da informação e documentação do programa espacial brasileiro*. 2012. 150 p. (sid.inpe.br/mtc-m19/2011/11.28.18.12-TDI). Dissertação (Mestrado em Engenharia e Gerenciamento de Sistemas Espaciais) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011. Disponível em:<<http://urlib.net/8JMKD3MGP7W/3ASHBPL>>. Acesso em: jan. 2015.

ALONSO, J. D. D. *Guia para padronização do desenvolvimento de software crítico para aplicações espaciais*. Dissertação (Mestrado) – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, 1998.

ALVARO, A.; ALMEIDA, E. S.; MEIRA, S. R. L. *A software component certification: a survey*. In: The 31st IEEE Euromicro conference on software engineering and advanced applications (SEAA). Component-Based *Software Engineering* (CBSE) Track. Porto, Portugal, 2005.

AMARAL, L.M.G. *Gap analysis methodology for the team software process* - Dissertation Report Integrated Master in Informatics and Computation Engineering, Porto, Portugal 2009.

AMARAL, L.M.G.; FARIA, J.P. *A gap analysis methodology for the team software process..* In: INTERNATIONAL CONFERENCE ON THE QUALITY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY (QUATIC). 7., 2010. Porto, Portugal. Proeedings... Porto: IEEE, 2010.

ANDRADE, P. I. *Qualidade nos processos do ciclo de vida do produto com CMMI: uma aplicação prática de gerência de configuração na Compsis*. 2001. Monografia – Departamento de Ciência da Computação, Universidade Federal de Lavras, MG, 2001.

ARRUDA, P.; COUTO, I; SOUZA F.L. *Modelos*. Universidade Gama Filho – Ciência da Computação, Candelária, RJ, 2008

BELLOQUIM, A. *PSP e TSP - Personal and team software process* - Gnosis - São Paulo – Brasil, 2005. Disponível em <<http://www.spinsp.org.br/apresentacao/PSPTSPSPIN.ppt>> Acesso em dez 2010.

AVELINO, A. M. *Processamento embarcado aplicado a detecção de vazamento*. 2009. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte, RN, 2009.

BALARIN, F. et al. *Hardware-software co-design of embedded systems: the POLIS approach*. Norwell: Kluwer, 1997.

BARR, M. *Programming embedded systems in C and C++*. California, USA: O'Reilly & Associates, 1999.

BARROSO, M. A. *Um processo de desenvolvimento para sistemas computacionais aderente ao MPS.BR Nível G*. 2010. Tese (Mestrado em Informática Aplicada) – Centro de Ciências Tecnológicas, Universidade de Fortaleza, Fortaleza, CE, 2010.

- BASS, L. et al. *Market assessment of component-based software engineering*. In: Software Engineering Institute (SEI). May 2000. Technical Report, v. I.
- BELNUOVO, V. et al. *Estudo de viabilidade da utilização do modelo MPS.BR nível G na metodologia Scrum*. 2010. Monografia - Curso de Sistemas de Informação – Faculdade de Exatas e Tecnologia, Universidade Metodista de São Paulo, São Bernardo do Campo, SP, 2010.
- BEM, F. *Avaliação do processo de gerência de requisitos em empresa desenvolvedora de software: um estudo de caso*. 2010. Tese (Pós-graduação em Engenharia e Qualidade de Software) – Universidade Comunitária da Região de Chapecó, Santa Catarina, SC, 2010.
- BEQUE, L. T. *Avaliação dos requisitos para teste de um sistema operacional embarcado*. 2009. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, RS, 2009.
- BERGMANN, T. *Avaliação de implantação do MPS.BR – nível G*. 2008. Monografia - Instituto de Informática e Estatística, Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC, 2008.
- BOEHM, B. et al. *Characteristics of Software Quality*. North Holland: Elsevier, 1978.
- BORBA, F. S. *Dicionário Unesp do português contemporâneo*. São Paulo: Unesp, 2004.
- CAROSIA, J. S. *Levantamento da qualidade do processo de software com foco em pequenas organizações*. 2004. Dissertação (Mestrado do Curso da Pós-graduação) em Computação Aplicada – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, 2004.
- CARRO, L.; WAGNER, F. *Sistemas computacionais embarcados*. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, 22., 2003, Campinas, SP. *Jornadas...* Ampinas: Sociedade Brasileira de Computação, 2003.
- CARVALHO, F. *An embedded software component quality verification framework*. 2010. Tese (Doutorado) – Universidade Federal de Pernambuco, PE, 2010.
- CARVALHO, F. et al. *A research of embedded software component quality and certification*. In: 11th Brazilian Workshop on Real-Time and Embedded Systems (Wtr). Recife, PE, 2009.
- CHAOS MANIFESTO. *CHAOS MANIFESTO 2013* Disponível em: <www.versionone.com/assets/img/files/chaosmanifesto2013.pdf>. Acesso em: out. 2014.
- _____. *Chaos Report 2009 - Novas informações, velhos problemas!* Disponível em: <<http://blogs.msdn.com/b/andredias/archive/2009/07/08/chaos-report-2009-novas-informa-es-velhos-problemas.aspx>>. Acesso em: jun. 2011.

- CORTADA, J. W.; QUINTELLA, H. M. *TQM: gerência da qualidade total*. Trad. E. Kanner. Rev. téc. H. M. Quintella. São Paulo: Makron Books, 1994.
- COUTO, A. B. *CMMI – Integração dos Modelos de Capacitação e Maturidade de Sistemas*. Rio de Janeiro: Ciência Moderna, 2007.
- CRESPO, A. B. et al. *Cobertura dos critérios potenciais-usos e a confiabilidade do software*. In: Simpósio Brasileiro de Engenharia de Software, 11. Fortaleza, 13-15 out. 1997. *Anais...* Fortaleza: Universidade Federal do Ceará, 1997.
- CURTIS, B. *The global pursuit of process maturity*. IEEE Software, v.17, n.4, p.76,78, Jul/Aug 2000. doi: 10.1109/MS.2000.854072.
- DEMARCO, T. *Controle de projetos de software*. Trad. M. E. Comenale et al. Rio de Janeiro: Campus, 1989.
- INTRODUÇÃO a teste de Software - qualidade de software. In: Devmedia. Revista Engenharia de software – Disponível em <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>>, São Paulo 2013. Acesso em: nov. 2014.
- DIB, L. A. R. *O processo de internacionalização de pequenas e médias empresas e o fenômeno Born Global: estudo do setor de software no Brasil*. Rio de Janeiro: COPPEAD/UFRJ, 2008.
- DIJKSTRA, E. W. *The Humble programmer*. *Commun. ACM*, v.15, n.10, p. 859–866, 1972.. Turing Award Lecture 1972.
- DONG-HYUN, LEE et al. *A survival kit: adaptive hardware/software codesign life-cycle model*. *Computer*, v. 42, n. 2, p.100-102, 2009.
- DOUGLASS, B. P. *Real time UML*. Second Edition – developing – efficient objects for embedded systems. São Paulo: Addison Wesley, 2000.
- DUNN, R. H.; ULLMAN, R. S. *TQM for computer software*. 2. ed. New York: McGraw-Hill, 1994. (McGraw-Hill system design and implementation series).
- EDWARDS, S. et al. *Design of embedded systems: formal models, validation and synthesis*. *Proceedings of IEEE*, v. 85, n. 3, p. 366-390, 1997.
- EUROPEAN COOPERATION FOR SPACE STANDARDIZATION (ECSS). *ECSS Secretariat ESA-ESTEC ECSS-E-10 Part 1B. Space engineering – requirements & standards division*. Noordwijk: The Netherlands, 2009a.

_____. *ECSS Secretariat ESA-ESTEC ECSS-Q-ST-80C- space product assurance – software product assurance*. Noordwijk: The Netherlands, 2009b.

_____. *ECSS Secretariat ESA-ESTEC ECSS-40C- Space engineering – software*. Noordwijk: The Netherlands, 2009c.

_____. *ECSS secretariat ESA-ESTEC ECSS release note for ECSS standards CD-ROM -* Noordwijk, The Netherlands, 2010. *Note Issue 1.2*.

FALLA, M. *Results and achievements from the DTI/EPSRC R&D programme in safety critical systems*. 1997. Disponível em: <www.comp.lancs.ac.uk/computing/resources/scs/index.html>. Acesso em: fev. 1998.

GAJSKI, D. D. et al. *Embedded system design – modeling. synthesis*. In: GUSTAFSON, D. A. *Engenharia de software*. Porto Alegre: Bookman, 2003.

GIBBS, W. W. *Software's chronic crisis*. Scientific American, p. 72-81, set. 1994..

GIOVANNI, M. F. R. K. G. *Hardware/software co-design*. Pennsylvania: IEEE, 1997.

GOMES, H. V. *Metodologia de projeto de software embarcado voltado ao teste*. 2010. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul, RS, 2010.

GONÇALVES, J. M.; BOAS, A. V. *Modelo de Maturidade e Capabilidade de Software (CMM)*. Versão 1.2. Campinas/SP: Fundação CPqD, 2001.

GRILLO, N. A. et al. *Mapeamento do modelo CMMI-DEV-V1.3 em questões objetivas para auxiliar as organizações a identificarem fragilidades em seus processos de desenvolvimento de software*. São Bernardo do Campo, SP: UMESP-FACET, 2011.

GUERRA, A. C.; COLOMBO, R. M. T. *Qualidade de produto de software*. Brasília: PBQP Software, 2009.

HAYES, B. *Measuring customer satisfaction: development and use of questionnaires*. Milwaukee: ASQC, 1992.

HERBSTER, R.F. *Desenvolvimento de software para dispositivos móveis baseados na plataforma maemo™*. Campina Grande, Paraíba, Brasil: Universidade Federal de Campina Grande Centro de Engenharia Elétrica e Informática, 2006

HERNÁNDEZ L., A. et al. *Team software process in GSD teams: a study of new work practices and models*. International Journal Of Human Capital And Information Technology Professionals, v. 1, n. 3,

2012. IBM. Rational Unified Process (RUP). Disponível em: <www01.ibm.com/software/rational/rup>. Acesso em: out. 2012.

HUMPHREY, W.S. *The Team Software (TSP SM) – CMU/SEI-2000-TR-023 ESC-TR-2000 023 – SEI Joint Program Office – HQ ESC/DIB 5 Eglin Street Hanscom AFB, MA 01731-2116*. Pensilvânia: Carnegie Mellon University, Nov. 2000.

IEEE. *Std 1012-2004 – IEEE standard for software verification and validation*. Revisão do padrão IEEE Std 1012-1998, 2004.

_____. *Standard glossary of software engineering terminology*. New York: IEEE Standard, 1990. (ANSI/IEEE Standard 610.121990).

INTELLIGES. *Desenvolvimento de produtos* Disponível em: <www.intelliges.com.br/servico-produtos-eletronicos.html>. Acesso em: dez. 2011.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 25000:2014. - software engineering – Software product Quality Requirements and Evaluation (SQuaRE)*. Genebra: Guide to SQuaRE, 2005.

_____. *ISO/IEC 12207. systems and software engineering – software life cycle processes*. Genebra: ISO, 2008.

KEEN, P. G. W. *Guia gerencial para a tecnologia da informação: conceitos essenciais terminologia para empresas e gerentes*. Rio de Janeiro: Campus, 1996.

KONDO, M. N. S. et al. *Estudo de requisitos do software embarcado no segmento da telemedicina*. São Paulo: Núcleo de Telemedicina, Laboratório de Sistemas Integráveis/Departamento de Engenharia de Sistemas Eletrônicos da Escola Politécnica da Universidade de São Paulo/Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos da Escola Politécnica da USP, 2011.

KOSCIANSKI, A. S. M. *Qualidade de software*. 2. ed. Rio de Janeiro: Novatec, 2007.

KOTONYA, G.; SOMMERVILLE, I. *Requirements engineering processes and techniques*. [S.l.]: Wiley, 1997.

KRUEGER, C. W. *Software Reuse*. ACM Computing Surveys, v. 24, n. 2, p. 131-183, 1992.

KUCINSKIS, F. N. *Considerações sobre o aumento da autonomia operacional no segmento espacial das missões do INPE*. 2011. Trabalho para o exame de qualificação de doutorado do curso de Pós-Graduação em Engenharia e Tecnologia Espacial – área de concentração em Engenharia e Gerenciamento de Sistemas Espaciais, 2011.

KUMAR, S. et al. *The codesign of embedded systems: a unified hardware/software representation*. Massachusetts: Kluwer, 1996.

KÜNDIG, A. T. *A note on the meaning of “embedded systems”*. In: KÜNDIG, A.; BÜHRER, R. E.; DÄHLER, J. (eds.). *Embedded systems: new approaches to their formal description and design. an advanced course*. Zürich, Switzerland: Springer Verlag, p. 1-6, 1987. . Lecture notes in computer science, v. 284.

LABROSSE, J. J. *Embedded systems building blocks: complete and ready-to-use modules in C*. 2. ed. San Francisco, CA: CMP, 2002.

_____. *Micro/Os II – the real time kernel*: [s.l.]: CMP Media, Inc., 2002. 606 p, 2002.

LAMAISON, D. A.; AMARAL, D. L. do. *Tolerância a falhas em uma aplicação de banco de dados de missão crítica*. 2007. Monografia (Bacharelado em Informática) - Universidade de Passo Fundo, Rio Grande do Sul, 2007.

LANNINO, A. et al. *Criteria for software reliability model comparisons*. IEEE Trans. Software Engineering, SE-10, n. 6, p. 687-691, nov. 1984.

LEITE, J. C. *Engenharia de software*. Disponível em: <<http://engenhariadesoftware.blogspot.com/2007/02/sistemas-computacionais.html>>. Acesso em: fev. 2006.

LOUREIRO, G. A. *A System engineering and concurrent engineering framework for the integrated development of complex products*. Loughborough, USA: Loughborough University, 1999.

_____. *CSE – engenharia de sistemas*. Engenharia e Tecnologia Espacial/ETE – Instituto Nacional de Pesquisas Espaciais – INPE. São José dos Campos, SP, 2010 – Nota de Aula.

LUCRÉDIO, D. et al. *Software reuse: Brazilian industry scenario*. *Journal of Systems and Software*, v.81, n.6, p.996-1013, Elsevier, 2007.

MACHADO, M. P.; SOUZA, S. F. *Métricas e qualidade de software*. Disponível em: <www.fattocs.com.br/download/qualidade-sw.pdf> 2008. Acesso em: jul 2010.

MARCONDES, H. *Uma arquitetura de componentes híbridos de hardware e software para sistemas embarcados*. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

MARWEDEL, P. *Embedded system design*. [S.l.]: Kluwer Academic Publishers, 2003.

MCCALL, J.; RICHARDS, P.; WALTERS, G. *Factors in software quality*, 1977. (NTIS AD-AO49-014, 015, 055).

MEDEIROS, H. *CMMI* – introdução. Disponível em: <<http://blog.hallanmedeiros.com/2012/12/05/cmmi-introducao/>>. Acesso em: jul. 2013.

MEYERS, C. B.; FEILER, P. H.; MARZ, T. *Proceedings of the real-time systems Engineering Workshop*. Special Report, CMU/SEI-2001-SR-022, ago. 2001.

MIYASHIRO, M. A. S. *Identificação e melhoria do nível de maturidade de uma organização explorando técnicas de inteligência computacional*. 2007. (INPE-15770-TDI/1513). Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais - INPE, São José Dos Campos, SP, 2009. Disponível em: <<http://urlib.net/8JMKD3MGPBW/328D38P>>. Acesso em: 20 mar. 2015.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. *One approach to the use of the practices of CMMI-DEV V1.3 level 2 in a process of development of Embedded Systems*. in: International Conference On Information, Intelligence, Systems, And Applications, 5., 2014, Grécia, 2014a.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. *Process for the development of embedded system following the practices of CMMI Level 2*. In: - Science and Information Conference (SAI2014), 2014, Londres. *Proceedings...* 2014b. DVD. .

MYASHIRO, M. A. S.; FERREIRA, M.G.V. *The practices of level 2 (Managed) CMMI-DEV V1.3 in development of embedded systems*. In: INTERNATIONAL CONFERENCE ON ELECTRICAL AND ELECTRONIC ENGINEERING, 2014, Hong Kong. *Proceedings...* 2014c. In: INTERNATIONAL CONFERENCE ON ELECTRICAL AND ELECTRONIC ENGINEERING, 2014, Hong Kong. *Proceedings...* 2014. DVD.

MIYASHIRO, M. A. S.; FERREIRA, M.G.V. *Phase cyclical process requirements for the development of embedded systems*. In: BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING., 2013, Niteroi. *Proceedings...* 2013. p. 155-156. DVD.

MIRACHI, S.; CARVALHO, F. C. *Desenvolvimento de sistemas embarcados para aplicações espaciais*. – 2009 - BRAZILIAN SYMPOSIUM ON AEROSPACE ENG. & APPLICATIONS COPYRIGHT © 2009 by AAB, 3rd CTA-DLR WORKSHOP ON DATA ANALYSIS & FLIGHT CONTROL September 14-16, 2009, S. J. Campos, SP, Brazil

MOSSBERG, W. S. *In the garden they're bugs; in PCs, they're defects*. *Wall Street Journal* (Eastern Edition), p. B1, Mar. 16, 1995.

NETO, R. J. *Processo para desenvolvimento de sistemas de tempo real e embarcados*. 2002. Disponível em: <<http://hp.br.inter.net/jrotta/docs/sistemastemporeal.pdf> >Acesso em jul 2010.

- PADUA, W. *Engenharia de software – fundamentos, métodos e padrões*. Rio de Janeiro: LTC, 2003.
- PÁSCOA, J. E. P. *Fatores e subfatores para avaliação da segurança em software de sistemas críticos*. Dissertação (Mestrado em Engenharia) – Escola Politécnica da Universidade de São Paulo, SP, 2002.
- PAULK, M. C. al. *Key practices of the capability maturity model SM, version 1.1*. Pittsburgh, PA: SEI/CMU, 1993. Technical Report – CMU/SEI-93-TR-025 – ESC-TR-93-178. _____. *The capability maturity model: guidelines for improving the software process/CMU/SEI*. Reading: Addison-Wesley, 1995a. (SEI series in software engineering).
- PERRY, W. E. *Quality concerns in software development*. Information Systems Management, v.9, p. 48-52, Spring 1992.
- PMBOK. *Guia PMBOK – Um guia do conjunto de conhecimentos em gerenciamento de projetos*. 3. ed., 2004.
- PRATES, L. A. *Diretrizes para desenvolvimento de software para sistemas embarcados*. Belo Horizonte: Universidade Federal de Minas Gerais/Instituto de Ciências Exatas – Departamento de Ciências da Computação, 2012.
- PRESSMAN, R. S. *Engenharia de software*. 6. ed. Rio de Janeiro: McGraw-Hill, 2006.
- _____. *Engenharia de software*. 6. ed. Rio de Janeiro: McGraw-Hill, 2011.
- QUEIROZ, A. D. *Uma abordagem para identificação da maturidade do processo de desenvolvimento de software utilizando o modelo MPS.BR a partir da sua aderência e mapeamento às práticas do CMMI V1.3*. São Bernardo do Campo: UMESP-FACET, 2011.
- REED, K. *Software engineering – a new millennium?* IEEE Software, p. 17-201, Jul./Ago., 2000.
- REGINATO, J.P.M. *Uma proposta de aperfeiçoamento de um processo de gerenciamento de requisitos de sistemas e de software e sua aplicação a sistemas espaciais e aeronáuticos embarcados*. 2012. 171 p. (sid.inpe.br/mtc-m19/2012/03.26.23.04-TDI). Dissertação (Mestrado em Engenharia e Gerenciamento de Sistemas Espaciais) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2012. Disponível em:<<http://urlib.net/8JMKD3MGP7W/3BJTJGH>>. Acesso em: 20 mar. 2015.
- REY, M. D. *ISO/IEC SQuaRE - the second generation of standards for software product quality* IASTED 2003. Califórnia: 2003.

- RIBEIRO, L. C. M. et al. *Definição de um processo de engenharia de requisitos para software embarcado na indústria automotiva baseada em uma arquitetura de processos de software*. In: WORKSHOP ANUAL DO MPS (WAMPS 2011), 2011, Campinas. *Workshop...* Campinas, 2011.
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBWR, K. C. *Qualidade de software – teoria e prática*. São Paulo: Prentice Hall, 2001.
- ROCHA, M. P. P. *Processos para sistemas de software de segurança crítica*. Rio de Janeiro: COPPE/UFRJ, 2004.
- SATO, G. Y. *Desenvolvimento de hardware utilizando ferramentas CAD/CAE – Desenvolvimento de material didático ou instrucional – Apostila 2002*.
- SALVIANO, C. S. *Melhoria e avaliação de processo de software com o modelo ISO/IEC 15504-5*. Tese (Pós-Graduação) – Universidade Federal de Lavras, MG, 2006.
- SANTOS, J. T.; SANTOS, J. B.; SANTOS, O. J. *Indústria sotropolitana de software: aspectos de sua atuação no desenvolvimento na sociedade da informação*. Salvador: Universidade Federal da Bahia, 2005.
- SAOTOME, O.; VINCE, E. *Arquitetura de hardware do computador de bordo para o satélite universitário Itasat e confiabilidade*. In: BRAZILIAN SYMPOSIUM ON AEROSPACE ENG. & APPLICATIONS 3RD CTA-DLR WORKSHOP ON DATA ANALYSIS & FLIGHT CONTROL, 2009, São José dos Campos. *Workshop...* São José dos Campos: AAB. , 2009.
- LUIZ S.O, O. D. *Arquiteturas de hardware para sistemas embarcados – sistemas embarcados*. Paraíba: Universidade Federal de Campina Grande, 2010.
- SCHULZ, S. et al. *Model-based codesign*. Computer, v. 31, n. 8, p. 60-67, Aug. 1998.
- SHANKAR, K., LYSECKY, R. *Control focused soft error detection for embedded applications*. IEEE Embedded Systems Letters, v. 2, n. 4, Dec. 2010.
- SICKLE, T. V. *Reusable software components: object-oriented embedded systems programming in C*. Upper Daddle River, NJ: Prentice Hall, 1997. (Prentice Hall Series on Programming Tools and Methodologies.).
- SIEGERT, M. E.; BRISOLARA, L. *MDEReq: uma abordagem para engenharia de requisitos no domínio de software embarcado milena rota senA*. Centro de Desenvolvimento Tecnológico – CDTEc – XIV ENPOS. Universidade Federal de Pelotas, RS, 2013.

SOFTWARE ENGINEERING INSTITUTE (SEI). *CMMI for Development, Version 1.3 CMMI-DEV, V1.3 – CMMI Product Team – Improving processes for developing better products and services – Technical Report*. Pittsburgh: Carnegie Mellon University, 2010.

SOFTWARE ENGINEERING *Report on a conference sponsored by the NATO science committee*. Garmisch, Germany, 7th to 11th October 1968.

_____. *Techniques report on a conference sponsored by the NATO science committee*. Rome, Italy, 27th to 31st October 1969.

SOFTEX. Disponível em: <www.softex.br/mpsbr/>. Acesso em: nov. 2010.

SOMMERVILLE, I. S. *Engenharia de software*. 8. ed. São Paulo: Pearson Education do Brasil, 2007.

_____. *Engenharia de software*. 9. ed. São Paulo: Pearson Education do Brasil, 2011. (Material Adicional do Professor.)

SOUZA, E. P. *Elementos fundamentais na melhoria da qualidade de software*. 2004. Dissertação (Mestrado) – Faculdade de Engenharia Mecânica. Universidade Estadual de Campinas, São Paulo, 2004.

SOUZA, P. B. *Melhoria do software embarcado em satélites e o INPE: proposta para um passo a mais*. 2002. 144 p. (INPE-14616-TDI/1195). Dissertação (Mestrado do Curso de Pós-Graduação em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais – INPE, São José dos Campos, SP, 2002. Disponível em: <<http://urlib.net/sid.inpe.br/iris@1905/2005/08.04.03.33>>. Acesso em: 24 mar. 2015.

SOUZA, P. N. *CSE-200-4 – introdução à tecnologia de satélites*. Notas de aula - Coordenação Geral de Engenharia e Tecnologia Espacial/ETE – Instituto Nacional de Pesquisas Espaciais INPE. São José dos Campos, SP, 2008.

SOUZA, W.T.S. *Estudo da implantação do modelo de qualidade CMMI nas organizações*. Trabalho de Conclusão de Curso - Engenharia Eletrônica e de Computação - Escola Politécnica - Universidade Federal do Rio de Janeiro, RJ, Brasil, 2006

SPIN. Disponível em: <www.spin.org.br/>. Acesso em: nov. 2010.

SPINOLA, M. *Diretrizes para o desenvolvimento de software de sistemas embarcados*. 1998. Tese (Doutorado) – Universidade de São Paulo, SP, 1998.

SPINOLA, M.; PESSÔA, M. S. P. *Produtos que embutem software: aspectos relevantes para a melhoria do processo de desenvolvimento*. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE

SOFTWARE, 11. WQS'97 – WORKSHOP QUALIDADE DE SOFTWARE, 1997, Fortaleza. *Anais...* Fortaleza, CE: Universidade Federal do Ceará, 1997.

SPINOLA, M.; PESSÔA, A.; VOLPE, R. L. D. *Uma experiência na implantação do modelo CMM*. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 11. WQS'97 – WORKSHOP QUALIDADE DE SOFTWARE, 1997, Fortaleza. *Anais...* Fortaleza, CE: Universidade Federal do Ceará, 1997.

STANKOVIC, J. et al. *Strategic directions in real-time and embedded systems*. *ACM Computing Surveys*, v. 28, n. 4, 1996.

STRAUSS, S.; EBENAU, R. G. *Software inspection process*. New York: McGraw-Hill, 1994. (System design and implementation series.).

THOMAS, D. E.; ADAMS, J. K.; SCHMIT, H. *A model and methodology for hardware-software codesign*. *Design & Test of Computers*, IEEE, v.10, n. 3, p. 6-15, Sept. 1993.

VERAS, M. *Gestão em TI*. Disponível em: <<http://gestaodeprojetos10.blogspot.com.br/2010/02/scrun.html>>. Acesso em: dez. 2014.

VÉRAS, P. C. *Modelagem e análise do software embarcado de piloto automático de um VANT*. 2007. Tese (Mestrado) – Instituto Tecnológico de Aeronáutica - ITA, São José dos Campos, SP, 2009.

VOAS, J. M. *Certifying off-the-shelf software components*. *IEEE Computer*, v. 31, n. 6, p. 53-59, 1998.

WAGNER, F. R. *Engenharia de software para sistemas embarcados - Minicurso*. III SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SISTEMAS COMPUTACIONAIS – SBESC/2013. Rio de Janeiro: Universidade Federal do Rio de Janeiro, 2013.

WAGNER F. R.; NASCIMENTO, F. A. M.; OLIVEIRA, M. F. S. *Model-driven engineering of complex embedded systems: concepts and tools*. Porto Alegre, RS: Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS)/Cooperative Computing and Communication Laboratory (C-LAB), University of Paderborn, Paderborn, Germany, 2011.

WALLNAU, K. *Software component certification: 10 useful distinctions*. In: *Technical Note CMU/SEI-2004-TN-031*, 2004.

WEHRMEISTER, M. A.; BECKER, L. B.; PEREIRA, C.E. *Metodologia de projeto orientada a objetos baseada em plataformas para sistemas tempo-real embarcados*. In: SIMPÓSIO BRASILEIRO

DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC2007), 25., 2007, Belém do Pará, PA. *Anais...* Belém, 2007.

Wikipédia. *A enciclopédia livre - conteúdos diversos*. Disponível em: <
http://pt.wikipedia.org/wiki/Wikip%C3%A9dia:P%C3%A1gina_principal >. Acesso em: jan 2014.

WOLF, W. *Hardware-software codesign*. IEEE Design & Test of Computers, v.10, n.3, p. 29-41, Sep. 1993.

YASSUDA, I. S. *Ciclo de vida de projetos na área espacial*. São José dos Campos: INPE, 2010. 32 p. (sid.inpe.br/mtc-m19@80/2010/03.02.19.04-PUD).. Exame de Qualificação – Gerenciamento e Engenharia de Sistemas Espaciais 2010 – INPE-16663-PUD/217. Disponível em:
<<http://urlib.net/8JMKD3MGP7W/3746NA8>>. Acesso em: 24 mar. 2015

YEN, T. WOLF, W. *Hardware-software co-synthesis of distributed embedded systems*. Kluwer, 1996.

ZURITA, M. E. P. V. *Projeto de sistemas embarcados*. Teresina, PI: Universidade Federal do Piauí, 2010.

Apêndice A – Glossário

Glossário

Glossário

1. Introdução

Este glossário apresenta termos utilizados na tese que possam oferecer dúvidas ao leitor. As definições dadas aqui têm o sentido com que os termos são utilizados na tese; quando houver outras definições podem ser apresentadas na sequência. Os sinônimos de cada termo também são apresentados nesta seção.

Os principais termos utilizados nesta tese foram incluídos também no Capítulo 2.

1.1 Visão geral

Os termos estão dispostos em ordem alfabética e apresentam a seguinte estrutura:

- a) nome do termo ou sigla;
- b) descrição conceitual do termo;
- c) apresentação de termos sinônimos;
- d) para as siglas em inglês apresenta-se o significado em inglês e sua tradução.

1.2 Definições

Os termos definidos aqui fazem parte essencial da tese de doutorado “Uma abordagem para o processo de desenvolvimento de sistema embarcado que atende ao nível 2 de maturidade do CMMI-DEV” e foram selecionados a partir de sua pesquisa bibliográfica.

2. Termos

- **A2100**

Família de satélites que fornecem banda larga, comunicações móveis e militares, bem como serviços de GPS para milhões de pessoas ao redor do globo, com mais de 40 satélites em órbita atualmente.

- **Ad Hoc**

Para este mesmo efeito

Ad hoc é uma expressão latina que significa “para isto” ou “para esta finalidade”. Na Engenharia de Software, a expressão ad hoc é utilizada para referenciar ciclos completos de construção de software que não foram projetados. Em um processo ad hoc, nenhuma técnica de uso geral é empregada, pois as fases variam a cada aplicação. Hipóteses ad hoc não são necessariamente incorretas.

- **AEB**

Agência Espacial Brasileira

Criada em 10 de fevereiro de 1994, a Agência Espacial Brasileira (AEB) é responsável por formular e coordenar a política espacial brasileira. Autarquia federal vinculada ao Ministério da Ciência, Tecnologia e Inovação (MCTI), a AEB tem dado continuidade aos esforços empreendidos pelo governo brasileiro, desde 1961, para promover a autonomia do setor espacial.

- **AR**

Acceptance review

Revisão de aceitação

- **Áreas de processo**

Ver PA.

- **Artefato**

Em ciência da computação, é o produto de uma ou mais atividades dentro do contexto do desenvolvimento de um *software* ou sistema.

- **Capabilidade**

Software Process Capability

Intervalo de resultados esperados que podem ser alcançados, seguido de um processo de *software*, tornando-os previsíveis. Usa-se também o termo *capacidade*.

- **Cargas úteis**

Equipamentos necessários para o cumprimento de uma missão, como antenas, sensores, transmissores, câmeras, experimentos científicos, entre outros.

- **Caráter intangível do *software***

Que não apresenta características suficientes para ser percebido ou entendido; que tende a enganar a percepção ou o entendimento.

- **CBERS**

China-Brazil Earth Resources Satellite

Satélite Sino-Brasileiro de Recursos Terrestres

O Programa CBERS iniciou-se com dois satélites de sensoriamento remoto. Em 2002, foram construídos dois novos satélites: o CBERS-3 e o CBERS-4, com novas cargas úteis; em 2007, foi lançado o CBERS-2B. O CBERS-3 foi lançado em 9 de dezembro de 2013, mas, devido a uma falha ocorrida no veículo lançador Longa Marcha 4B, não foi colocado na órbita prevista, o que resultou em sua reentrada na atmosfera da Terra. Após a falha do lançamento, Brasil e China decidiram antecipar o lançamento do CBERS-4 – originalmente previsto para dezembro de 2015 – para dezembro de 2014.

- **CDR**
Critical design review
Revisão de projeto detalhado.
- **Ciclo**
Conjunto de fatos (atividades) que se sucedem em determinada ordem.
- **Ciclo de vida**
O ciclo de vida consiste no conjunto de fases que o compõem, geralmente em ordem sequencial de acontecimentos, relacionados de forma lógica, e sua conclusão pode ser marcada pela entrega de um ou mais artefatos.
É uma estrutura contendo processos, atividades e tarefas envolvidos no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema desde a definição de seus requisitos até o término de seu uso.
- **Ciclo independente**
Ciclo de vida que se realiza com autonomia diante de um projeto maior; é autônomo, independente, realizado em um projeto exclusivo.
- **CMM**
Capability Maturity Model
Modelo de Maturidade em Capacitação
Um modelo estruturado em cinco níveis para avaliação da maturidade dos processos de software de uma organização e para identificação das práticas que são requeridas para aumentar a maturidade desses processos. O primeiro CMM foi concebido pelo SEI para organizações de software e publicado no livro *The Capability Maturity Model: Guidelines for Improving the Software Process*.
- **CMMI**
Capability Maturity Model – Integration
Modelo de Maturidade em Capacitação – Integração
A versão atual do CMMI (versão 1.3) foi publicada em 27 de outubro de 2010 e apresenta três modelos: CMMI for Development (CMMI-DEV), voltado ao processo de desenvolvimento de produtos e serviços; CMMI for Acquisition (CMMI-ACQ), voltado aos processos de aquisição e terceirização de bens e serviços; CMMI for Services (CMMI-SVC), voltado aos processos de empresas prestadoras de serviços.
- **CMU**
Carnegie Mellon University
Instituição privada de ensino e pesquisa localizada na cidade de Pittsburgh, no estado da Pensilvânia, nos Estados Unidos.
- **Cobertura PA**
Alcance ou abrangência de realizações das práticas requeridas em determinada área de processo.
- **Componentes embarcados**
Dispositivo que pode ser parte de um sistema embarcado que, muitas vezes, tem em sua concepção parte hardware e parte software.

- **Componentes externos**

Dispositivo embarcado que foi ou está sendo produzido externamente ao projeto corrente na mesma empresa ou empresa contratada.

- **CTA**

Comando-Geral de Tecnologia Aeroespacial

O Centro Técnico de Aeronáutica é o órgão científico e técnico do Ministério da Aeronáutica com o objetivo de exercer suas atividades em prol da Força Aérea Brasileira, da Aviação Civil e da futura Indústria Aeronáutica, segundo os programas e planos do Ministério.

Também é chamado de Centro Técnico Aeroespacial.

- **Descarte**

Ato de descartar ou colocar em desuso um produto ou um artefato.

- **DOD**

Department of Defense – DoD Washington, DC: Office of the Under Secretary of Defense

Departamento de Defesa dos Estados Unidos da América

Também é conhecido como Departamento de Defesa dos Estados Unidos.

- **ECSS**

European Cooperation for Space Standardization

Iniciativa criada para desenvolver um conjunto de normas espaciais para utilização por toda a comunidade espacial europeia.

- **ECSS-M-ST-10C**

Space Project Management / Project Planning and Implementation

Gerenciamento de Projetos Espaciais / Planejamento do Projeto e Implementação

- **EIA/IS**

Electronic Industries Alliance Interim Standard (EIA)

- O EIA (Electronic Industries Association até 1997) foi uma organização de padrões e de comércio composta como uma aliança de associações comerciais para os fabricantes de eletrônicos nos Estados Unidos. Desenvolveu normas para garantir que equipamentos de diferentes fabricantes fossem compatíveis e intercambiáveis. Suas operações foram encerradas em 11 de fevereiro de 2011.

- **Estado perigoso**

Característica ou condição de um produto que, em caso de falha, pode provocar morte ou grandes catástrofes.

- **Failure**

Falha/erro

Qualquer manifestação de defeito que pode ocorrer em um componente.

- **Fault**

Falta/bug/defeito/problema

Problema identificado em um componente por ser requisito implementado de forma indevida ou por falta de requisito, sem que tal problema tenha se manifestado.

- **Firmware**

Conjunto de instruções operacionais programadas diretamente no hardware de um equipamento eletrônico. É armazenado permanentemente em um circuito integrado.

- **FRR**

Flight Readiness Review

Revisão de prontidão para voo.

- **GPS**

Sistema de Posicionamento Global

Sistema de navegação por satélite que fornece a um aparelho receptor móvel sua posição, assim como informação horária sobre todas condições atmosféricas, a qualquer momento e em qualquer lugar na Terra, desde que o receptor encontre-se no campo de visão dos satélites.

- **Hubble Space Telescope (HST)**

Telescópio espacial Hubble

Satélite astronômico artificial não tripulado que transporta um grande telescópio. Foi lançado pela NASA, agência espacial estadunidense, em 24 de abril de 1990, a bordo do ônibus espacial Discovery. Já recebeu várias visitas espaciais da NASA, para manutenção e substituição de equipamentos obsoletos ou inoperantes.

- **Humphrey**

Watts Humphrey, um dos ícones da engenharia de software, trouxe a engenharia para engenharia de software. Seu trabalho teve um impacto incomensurável na comunidade global de software, de forma incansável, exortando a comunidade para enfatizar a qualidade, medição e performance.

Conhecido como o “Pai da Qualidade de Software”, Humphrey dedicou a maior parte de sua carreira à resolução dos problemas no desenvolvimento de software, incluindo atrasos no cronograma, aumentos de custos, problemas de desempenho e defeitos. No SEI, liderou o desenvolvimento do Capability Maturity Model Software (CMM), e introduziu os métodos de avaliação da capacidade de processo de software. Mais tarde, esses métodos tornaram-se a base para o desenvolvimento do Capability Maturity Model Integration (CMMI), um quadro das melhores práticas de engenharia de software que vêm sendo adotadas por milhares de organizações em todo o mundo. Humphrey também liderou o desenvolvimento do Processo de Software Pessoal (PSP) e o Team Software Process (TSP). Em 2005, recebeu a Medalha Nacional de Tecnologia, a mais alta honraria concedida pelo presidente dos Estados Unidos para os maiores inovadores da América. Humphrey morreu em 28 outubro de 2010, com 83 anos de idade.

- **IEC**

International Electrotechnical Commission

Comissão Electrotécnica Internacional

A IEC é uma organização internacional de padronização de tecnologias elétricas, eletrônicas e relacionadas. Alguns de seus padrões são desenvolvidos em parceria com a Organização Internacional para Padronização (ISO). A sede da IEC, fundada em 1906, fica em Genebra, na Suíça.

- **IEC 14598**

Engenharia de Software – Avaliação da Qualidade de Produto de Software

Processo de avaliação da qualidade de produto de software, contando com as principais características de um processo desse tipo.

- **Incremental**

Forma de realização das tarefas em que partes iniciais são feitas e acrescentadas de acordo com o planejamento.

- **INPE**

Instituto Nacional de Pesquisas Espaciais

O INPE surgiu logo após os lançamentos dos satélites Sputnik 1, da então União Soviética, e Explorer 1, dos Estados Unidos, que mobilizaram dois alunos de engenharia do Instituto Tecnológico de Aeronáutica (ITA), Fernando de Mendonça e Júlio Alberto de Moraes Coutinho, com o objetivo de construir uma estação para receber sinais do satélite norte-americano em cooperação com o Laboratório de Pesquisa Naval da Marinha daquele país.

- **Integração**

Atividade para combinar física e funcionalmente partes de um produto (hardware ou software) para obter uma configuração especial funcional.

- **INTEGRAL**

Internacional Astrophysics Laboratory Gamma-Ray

Laboratório Internacional de Astrofísica para Raio Gama

- **ISO**

**International Organization for Standardization / International Organization for Standardization
Organização Internacional para Padronização / Organização Internacional de Normalização**

Popularmente conhecida como ISO1, é uma entidade que congrega os grêmios de padronização/normalização de 170 países.

- **ISO 8402**

Gestão da qualidade e garantia da qualidade – Terminologia

Define os termos fundamentais relativos aos conceitos da qualidade, aplicáveis a todas as áreas, para a elaboração e utilização de normas sobre qualidade, e para entendimento mútuo nas comunicações nacionais e internacionais.

- **ISO/IEC 12119**

Tecnologia de informação – Pacotes de *software* – Teste e requisitos de qualidade

Esta norma é aplicável a pacotes de software, como processadores de texto, planilhas eletrônicas, bancos de dados, software gráficos, programas para funções técnicas ou científicas e programas utilitários.

- **ISO/IEC 9126**

Norma ISO para qualidade de produto de software, que se enquadra no modelo de qualidade das normas da família 9000. Foi substituída pela norma ISO 25000.

- **ISO/IEC 15504**

Software Process Improvement and Capability dEtermination

Padrão Internacional para Avaliação de Processos de Software

Desenvolvida pela ISO e pelo IEC, com o apoio do projeto SPICE, segue como modelo de referência de processo a norma ISO/IEC 12207.

- **ISO/IEC12207**
Norma Internacional de Processos de Desenvolvimento de Software
Estabelece processos, atividades e tarefas a serem executados durante os processos de aquisição, fornecimento, operação, desenvolvimento e manutenção de software.
- **ISO 25000**
Software Product Quality Requirements and Evaluation
Requisitos e Avaliação da qualidade de produto de software
A ISO/IEC 25000 é o resultado da evolução dos vários outros padrões, especificamente da norma ISO/IEC 9126, que define um modelo para avaliação da qualidade do produto de software, e da norma ISO/IEC 14598, que define o processo de avaliação do produto de software.
- **Iterativo**
É uma característica de fazer de novo, isto é, de realizar inúmeras vezes um grupo de atividades. Consiste na realização parcial de todas as disciplinas de uma fase, em que cada uma delas pode ter “n” iterações.
- **MA-MPS**
Modelo de avaliação.
- **Mars Express**
Missão espacial não tripulada da Agência Espacial Europeia (ESA) e da Agência Espacial Italiana, destinada a estudar o planeta Marte. A sonda foi lançada em 2 de junho de 2003, no Cosmódromo de Baikonur, no Cazaquistão, pelo foguete Soyuz-Fregat, fabricado pela firma Starsem, uma companhia russo-europeia. A expressão “Express de Mars Express” deve-se ao menor tempo de viagem que a sonda levaria a Marte, pois esse planeta e a Terra estavam em posição de máxima aproximação, o que ocorre uma vez a cada 60 mil anos. Também recebeu esse nome pela rapidez com que a missão foi projetada e executada – em um tempo menor que o de qualquer missão anterior.
- **Maturidade**
Grau da implantação de processos documentados, gerenciados, medidos, controlados e melhorados continuamente, realizada de forma consistente por uma organização. A maturidade organizacional pode ser medida por meio de avaliações. Desenvolvimento pleno, firmeza, decisão.
- **Metodologia**
Estudo dos métodos, ou seja, das etapas a serem seguidas em determinado processo para oferecer maior controle sobre os recursos que serão utilizados no projeto. Consiste na explicação minuciosa, detalhada, rigorosa e exata de toda ação desenvolvida no método (caminho) do trabalho de pesquisa.
- **MCT**
Ministério da Ciência e Tecnologia
- **MDR**
Mission Definition Review
Revisão de definição de missão.
- **Mitigar Riscos**
Eliminar, reduzir o impacto ou diminuir as consequências, de um risco detectado por meio de planos de ação.
- **MN-MPS**
Modelo de negócio.

- **Modelo**
Espécie de guia ou manual das melhores práticas encontradas por organizações de sucesso, dada uma área de aplicação. Recomenda ações e não define passos ou ferramentas para a execução de uma atividade. O modelo não é um programa de melhoria de processos; ele apresenta as observações que devem ser feitas na elaboração de um programa.
- **Modelo de referência**
Modelo utilizado para comparação ao se medir algum atributo.
- **MPS.BR**
Melhoria de Processo do *Software* Brasileiro
Programa mobilizador de longo prazo criado em dezembro de 2003 e coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), que conta com o apoio do Ministério da Ciência e Tecnologia (MCT), da Financiadora de Estudos e Projetos (FINEP), do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (Sebrae) e do Banco Interamericano de Desenvolvimento (BID). O objetivo do programa MPS.BR é a melhoria de processo do software brasileiro.
- **MR-MPS**
Modelo de referência
Contém a descrição geral do modelo MPS e detalha seus componentes e as definições comuns necessárias para seu entendimento e aplicação.
- **NASA**
National Aeronautics and Space Administration
É uma agência do governo dos Estados Unidos da América responsável pela pesquisa e pelo desenvolvimento de tecnologias e programas de exploração espacial.
- **Necessidades explícitas**
Necessidades solicitadas pelos clientes na definição dos requisitos. Refletem as características e as condições para o produto ser utilizado, atendendo aos objetivos, às funções e ao desempenho esperado.
- **Necessidades implícitas**
Requisitos que estão subentendidos em vez de explicitamente expressados. Em geral, referem-se a conceitos técnicos desconhecidos pelo solicitante, mas são necessários para a realização do requisito. Em geral, tais requisitos não são solicitados pelo cliente, por serem transparentes ao seu conhecimento ou óbvios a sua necessidade, mas precisam ser atendidos pelo *software*.
- **ORR**
Operational Readiness Review
Revisão de Prontidão de Operação
- **PA**
Áreas de processo
Conjunto de práticas relacionadas em uma área que, quando implementadas conjuntamente, satisfazem a um conjunto de metas consideradas importantes para a realização de melhorias nessa área.
- **Paradigma**
Representação de um padrão a ser seguido, conceito. Consiste nas normas orientadoras que estabelecem limites e o modo como um indivíduo deve agir dentro deles. Também é chamado de processo.

- **Partes interessadas**

Pessoas que possibilitam a realização do projeto e para as quais esse projeto vai produzir o benefício acordado, o que justifica sua produção.

Também são chamadas de usuários, *stackholders*, clientes, entre outros.

- **PDR**

Preliminary Design Review

Revisão de Projeto Preliminar

- **Processo**

Coleção de padrões que definem um conjunto de atividades, ações, tarefas de trabalho, produtos de trabalho e/ou comportamentos relacionados necessários ao desenvolvimento de um *software* de computador.

Trata-se de uma sequência de passos realizados para um dado propósito. Simplificando, processo é aquilo que você faz. “Processo é aquilo que as pessoas fazem, usando procedimentos, métodos, ferramentas, e equipamentos, para transformar matéria prima (entradas) em produto (saída) que tenha valor para o cliente.

- **Processo Cíclico**

O processo é denominado cíclico, pois cada componente do sistema embarcado a ser desenvolvido segundo suas especificações deve realizar todas as suas fases e atividades, formando assim um ciclo de desenvolvimento de componentes de sistema embarcado.

- **Processo de desenvolvimento de *software***

Conjunto de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter o *software* e seus produtos relacionados.

- **Processo flexível**

Processo que pode ser adaptado para atender às particularidades do projeto, isto é, suas tarefas, atividades, entradas e saídas. Pode ser modificado de acordo com as características do componente a ser produzido.

- **Processo individualizado**

Processo que direciona os desenvolvedores a realizar as atividades de desenvolvimento das partes *software* e hardware do componente separadamente, uma vez que, por terem conceitos diferentes, essas partes precisam ser conduzidas de forma independente.

- **Produto espacial**

Tipo de produto que pode variar de um único equipamento ou até mesmo um produto complexo, como satélite ou um lançador de um sistema espacial completo, com requisitos de funcionamento para as condições ambientais do espaço.

- **Projeto Apollo**

Esse projeto foi criado pela Agência Espacial Americana, com um conjunto de missões espaciais, para levar o homem até a Lua. Esse objetivo foi atingido em 20 de julho de 1969, com o pouso da Apollo 11 no solo lunar. A missão completa teve sete voos não tripulados e onze voos tripulados. Em 1972, o Projeto Apollo foi abandonado por falta de verba e de interesse dos estadunidenses após o voo da Apollo 17.

- **Projeto SQuaRE**

Ver ISO 25000.

- **PRR**

Preliminary Requirement Review

Revisão de Qualificação

- **PSP**

- **Personal Software Process**

- Processo de Software Pessoal

- O PSP é um processo definido para ajudar o desenvolvedor a fazer melhor seu trabalho e favorece a capacitação de indivíduos.

- **QR**

- **Qualification Review**

- **Repetibilidade**

- Condição de se repetir, isto é, de conquistar a mesma qualidade em diferentes projetos.

- **Rosetta**

- Lançada em 2 de março de 2004 da base de Kourou, na Guiana Francesa, pelo foguete Ariane 5, a sonda atingiu seu alvo em meados de 2014. A nave é formada por duas partes: a sonda espacial Rosetta, que carrega onze instrumentos, e o pousador-robótico, que transporta mais dez. A sonda recebeu esse nome em homenagem à pedra da roseta, que, após sua descoberta em 1799, auxiliou no entendimento dos hieróglifos egípcios.

- **RUP**

- **Rational Unified Process**

- Plataforma de processo de *software* desenvolvida pela Rational Software Corporation, a partir do processo unificado (UP), para auxiliar no desenvolvimento de sistemas, em especial dos orientados a objetos, uma vez que tal plataforma oferece ferramentas sistemáticas para a elaboração dos diagramas da UML. Atualmente, pertence à IBM.

- **Satélites de sensoriamento remoto**

- Conjunto de técnicas que possibilitam a obtenção de informações a respeito de alvos na superfície terrestre (objetos, áreas, fenômenos), por meio do registro da interação da radiação eletromagnética com a superfície, realizado por sensores distantes ou remotos. Geralmente, esses sensores estão presentes em plataformas orbitais ou satélites, aviões e a nível de campo. A Nasa é uma das maiores captadoras de imagens recebidas por seus satélites. No Brasil, o principal órgão que atua nessa área é o Instituto Nacional de Pesquisas Espaciais (INPE).

- **Satélites de telecomunicações**

- Satélites artificiais para fins de telecomunicações. Os satélites de comunicação modernos usam órbitas geoestacionárias, órbitas Molniya ou baixas órbitas polares. Também são chamados de comsat.

- **SEI**

- **Software Engineering Institute**

- Centro de pesquisa e desenvolvimento patrocinado pelo Departamento de Defesa dos Estados Unidos da América que provê uma prática avançada de engenharia de *software*, qualificando graus de qualidade de *software*. É patrocinado pelo Departamento de Defesa dos EUA (DoD) e localizado na Carnegie Mellon University.

- **Sistema de uso global**

- Sistemas ou partes de sistemas com grande número de utilizações que, algumas vezes, tornam as pessoas dependentes.

- **Sistema embarcado**

- Produto formado por um conjunto de componentes encapsulados ou dedicados ao dispositivo ou sistema que ele controla, tendo em sua composição componentes de *hardware* e componentes de *software*.

- Também é chamado de sistema embutido, *software* embarcado, *software* embutido.

- **Sistema crítico**
Característica e condições de um produto ou sistema cujos parâmetros, requisitos ou itens exigem atenção especial para que seus objetivos possam ser cumpridos.
- **SOFTEX**
Associação para Promoção da Excelência do Software Brasileiro
Organização da Sociedade Civil de Interesse Público que visa aumentar a competitividade da indústria de *software* brasileira, por meio de ações em três áreas-fim: Capacitação e Inovação; Mercado; Qualidade e Competitividade.
- **Software embarcado**
Parte desenvolvida com recurso de programação computacional de um componente embarcado. Também é chamado de sistema embutido, sistema embarcado, *software* embutido.
- **SOHO**
Solar and Heliospheric Observatory
Observatório Solar e Heliosférico
- **Sondas interplanetárias**
Nave espacial não tripulada utilizada para a exploração remota de outros planetas, satélites, asteroides ou cometas. Geralmente, as sondas têm recursos de telemetria, que permitem estudar à distância suas características físico-químicas e, por vezes, também seu meio ambiente, tirando fotografias.
- **SPICE**
Ver ISO 15504.
- **SRR**
System Requirement Review
Revisão de Requisitos de Sistema
- **TCAS**
Traffic Alert and Collision Avoidance System - Traffic Collision Avoidance System
Sistema Anticolisão de Tráfego
Pequeno conjunto de equipamentos eletrônicos de bordo que constituem um sistema de segurança de voo. É incorporado às aeronaves com o objetivo de evitar colisões aéreas com outras aeronaves.
- **TVM430**
Transmissão Voie-Machine
Transmissão da via para o comboio

Forma de sinalização na cabine originalmente implantado na França e usado em linhas ferroviárias de alta velocidade. TVM-300 foi a primeira versão, seguido de TVM-430. Em altas velocidades, um motorista não consegue ver de maneira preciosa os sinais ferroviários com base na cor da luz na via. Informações relativas à sinalização são transmitidas ao comboio e exibidas como parte dos controles de trem. Ao motorista é mostrada a velocidade de operação segura, exibida em quilômetros por hora.

O sistema de TVM-430 1980 transmite também perfis de gradiente e informações a respeito do estado de sinalização, alguns blocos adiante. O alto grau de automação não tira do motorista a operação do trem, embora alguns controles possam fazer o comboio parar com segurança em caso de falha humana.
- **Therac-25**
Foi uma moderna máquina de radioterapia controlada por computador, e durante dois anos provocou ao menos 6 acidentes nos quais os pacientes receberam overdose de radiação, que resultou em ao menos cinco mortes causados por erros no software que controlava a máquina.

- **TQM**

- **Total Quality Management**

- Gestão da qualidade total

- Conjunto de esforços feitos por uma organização para melhorar continuamente sua capacidade de fornecer melhores produtos e serviços aos seus clientes.

- TQM é uma forma de gerenciar uma empresa ou outra organização, que concentra seus esforços de forma sistemática e disciplinada na melhoria contínua da qualidade de tudo que faz.

- **UML**

- **Unified Modeling Language**

- Linguagem de Modelagem Unificada

- Linguagem de modelagem que permite representar um sistema de forma padronizada. Por não ser uma metodologia de desenvolvimento, não determina a sequência das atividades para projetar seu sistema. Ajuda a visualizar os desenhos do projeto e a comunicação entre os objetos.

- **Voyager 1**

- Sonda espacial norte-americana lançada ao espaço em 5 de setembro de 1977 para estudar o sistema solar exterior e, posteriormente, o espaço interestelar. Embora esteja em operação há mais de 36 anos, até hoje essa sonda recebe comandos de rotina e transmite dados para a Terra. Segundo oficialmente confirmado pela Nasa no dia 12 de setembro de 2013, está fora do sistema solar.

- **Voyager 2**

- Nave robótica norte-americana lançada pela Nasa a 20 de agosto de 1977, da Estação da Força Aérea de Cabo Canaveral, na Flórida, aproximou-se dos quatro planetas gigantes do sistema solar, produzindo valiosíssimos resultados científicos e as melhores fotografias daqueles corpos e de seus satélites obtidas até então. Tornou-se o quarto artefato humano a ultrapassar a órbita de Plutão, em 1989.

- **Voyager programa**

- Desenvolvimento de duas sondas de exploração interplanetária (Voyager 1 e 2), com o objetivo de realizar um *grand tour* espacial.

- **XMM-Newton X-ray**

- Missão Multiespelho de Raios X

- Observatório orbital de raios X, lançado pela ESA em dezembro de 1999, pelo foguete Ariane 5. Seu nome é uma homenagem ao cientista Isaac Newton.

Apêndice B – Processo de desenvolvimento – descrição

Processo para o Desenvolvimento de Sistema Embarcado

**CMMI Nível 2
Versão 1.0**

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Histórico da Revisão

Data	Versão	Descrição	Autor

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Lista de Figuras

Figura 1.1 - Organização das disciplinas do padrão ECSS	11
Figura 2.1 - Representação do uso da simbologia.....	12
Figura 2.2 - Simbologia do Processo.....	13
Figura 2.3 - Representação das partes de um sistema	14
Figura 2.4 - Processo cíclico.....	15
Figura 2.5 - Componentes do processo cíclico	16
Figura 3.1 - Atividades comuns - Fase	18
Figura 3.2 - Engenharia - Ciclo.....	19
Figura 3.3 - Engenharia - Fase	20
Figura 3.4 - Requisitos - Fase	22
Figura 3.5 - Requisitos do produto - Ciclo	23
Figura 3.6 - Análise e projeto - Fase	25
Figura 3.7 - Análise e projeto - Ciclo	26
Figura 3.8 - Implementação - Fase.....	27
Figura 3.9 - Integração - Fase	28
Figura 3.10 - Implementação e integração - Ciclo	29
Figura 3.11 - Validação e verificação - Fase	30
Figura 3.12 - Verificação e validação - Ciclo	31
Figura 3.13 - Avaliação do ciclo - Fase.....	32
Figura 3.14 - Avaliação do ciclo - Ciclo.....	33
Figura 4.1 - Representação gráfica do fluxograma	34
Figura 4.2 - Fluxograma geral de realização do processo - Parte-1	35
Figura 4.3 - Fluxograma geral de realização do processo - Parte-2	36
Figura 4.4 - Notação para o fluxograma das fases	37
Figura 4.5 - Fluxograma da fase de engenharia	38
Figura 4.6 - Atividade controle de projeto.....	39
Figura 4.7 - Controle configuração	40
Figura 4.8 - Definir escopo-3	41
Figura 4.9 - Planejamento-4.....	42
Figura 4.10 - Analisar riscos-5.....	43
Figura 4.11 - Planejar métricas-6	44
Figura 4.12 - Análise de requisitos-F2	45
Figura 4.13 - Especificação de requisito-7	46
Figura 4.14 - Definição de componente-8	47
Figura 4.15 - Planejamento do sistema-9.....	48
Figura 4.16 - Projeto de componente-10	49
Figura 4.17 - Definir itens de controle-11	50

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Figura 4.18 - Fluxograma da fase de análise e projeto	51
Figura 4.19 - Detalhar componente-12.....	52
Figura 4.20 - Projeto de <i>hardware</i> -13	53
Figura 4.21 - Projeto de <i>software</i> -14.....	54
Figura 4.22 - Protótipo-15	55
Figura 4.23 - Validar modelo-16.....	56
Figura 4.24 - Implementação e integração-F4	57
Figura 4.25 - Implementação de SW-17.....	58
Figura 4.26 - Implementação de HW-18	59
Figura 4.27 - Teste de unidade-19.....	60
Figura 4.28 - Integração-20.....	61
Figura 4.29 - Teste-21	62
Figura 4.30 - Fluxograma da fase de verificação e validação.....	63
Figura 4.31 - Teste de SW-22	64
Figura 4.32 -Teste de HW-23	65
Figura 4.34 - Fechamento-24.....	67
Figura 4.35 - Avaliação-25	68
Figura 5.1 - Propriedades do documento Word (<i>templates</i>)	69
Figura 5.2 - Representação do recurso do <i>template</i> Word.....	70
Figura 5.3 - Representação do <i>template</i> Word.....	71
Figura 5.4 - Cabeçalho e rodapé do <i>template</i>	71

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Lista de Tabelas

Tabela 6.1 - Práticas genéricas	74
Tabela 6.2 - Gerência de requisitos	75
Tabela 6.3 - Medição e análise	76
Tabela 6.4 - Acompanhamento e controle de projeto	77
Tabela 6.5 - Gerência de acordos com fornecedores	78
Tabela 6.6 - Planejamento de projeto	79
Tabela 6.7 - Gerência de configuração	81
Tabela 6.8 - Garantia da qualidade de processo e produto	82
Tabela 6.9 - Desenvolvimento de requisitos	83
Tabela 6.10 - Solução técnica	84
Tabela 6.11 - Integração de produto	85
Tabela 6.12 - Verificação	86
Tabela 6.13 - Validação	89
Tabela 6.14 - Foco no processo da organização	90
Tabela 6.15 - Definição do processo da organização	90
Tabela 6.16 - Treinamento organizacional	91
Tabela 6.17 - Gerência integrada de projeto	92
Tabela 6.18 - Gerência de risco	93
Tabela 6.19 - Análise de decisão e resolução	94
Tabela 6.20 - Desempenho do processo organizacional	95
Tabela 6.21 - Gerência quantitativa de projeto	95
Tabela 6.22 - Análise causal e resolução	96
Tabela 6.23 - Gestão do processo organizacional	97

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

ÍNDICE

Lista de Figuras	3
Lista de Tabelas	5
1. Introdução	9
1.1. Finalidade.....	9
1.2. Escopo do documento.....	9
1.3. Envolvidos	10
1.4. Posicionamento do processo proposto no padrão ECSS.....	10
1.4.1. Organização das disciplinas do padrão ECSS.....	11
2. Representação	12
2.1. Como entender a disposição dos componentes do processo	12
2.2. Simbologia utilizada	13
2.3. Processo cíclico.....	13
3. Descrição das fases.....	16
3.1. Descrição dos modelos de documentos resultados das atividades das fases do processo.....	16
3.2. Atividades comuns(FC)	17
3.3. Engenharia(F1).....	19
3.4. Análise de requisitos (F2)	21
3.5. Análise e projeto (F3)	24
3.6. Implementação e integração (F4).	26
3.7. Verificação e validação do Sistema (F5)	29
3.8. Avaliação do ciclo (F6).....	31
4. Fluxogramas.....	34
4.1. Orientações para o entendimento do fluxograma	34
4.2. Fluxograma geral do processo.....	35
4.3. Fluxograma de cada fase do processo.....	37
4.3.1. Fluxograma da fase de engenharia – F1	38
4.3.1.1. Fluxograma da atividade de controle de projeto - 1	39
4.3.1.2. Fluxograma da atividade controle de configuração - 2.....	40
4.3.1.3. Fluxograma da atividade definir escopo - 3.....	41
4.3.1.4. Fluxograma da atividade planejamento - 4.....	42
4.3.1.5. Fluxograma da atividade analisar riscos - 5.....	43
4.3.1.6. Fluxograma da atividade planejar métricas - 6	44
4.3.2. Fluxograma da fase de análise de requisito - F2	45
4.3.2.1. Fluxograma da atividade especificação de requisito - 7	46
4.3.2.2. Fluxograma da atividade definição de componente - 8	47
4.3.2.3. Fluxograma da atividade planejamento do sistema - 9.....	48
4.3.2.4. Fluxograma da atividade projeto de componente - 10.....	49
4.3.2.5. Fluxograma da atividade definir itens de controle - 11	50

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.6. Fluxograma da fase da análise e projeto - F3.....	51
4.3.2.7. Fluxograma da atividade detalhar componente - 12.....	52
4.3.2.8. Fluxograma da atividade projeto de hardware - 13	53
4.3.2.9. Fluxograma da atividade projeto de software - 14	54
4.3.2.10. Fluxograma da atividade protótipo - 15.....	55
4.3.2.11. Fluxograma da atividade validar modelo - 16.....	56
4.3.2.12. Fluxograma da fase de implementação e integração - F4	57
4.3.2.13. Fluxograma da atividade implementação de SW - 17	58
4.3.2.14. Fluxograma da atividade implementação de HW - 18.....	59
4.3.2.15. Fluxograma da atividade teste de unidade - 19	60
4.3.2.16. Fluxograma da atividade integração - 20.....	61
4.3.2.17. Fluxograma da atividade teste - 21.....	62
4.3.3. Fluxograma da fase de verificação e validação.....	63
4.3.3.1. Fluxograma da atividade teste de SW - 22.....	64
4.3.3.2. Fluxograma da atividade teste de HW - 23.....	65
4.3.4. Fluxograma da fase de avaliação do ciclo.....	66
4.3.4.1. Fluxograma da atividade fechamento - 24.....	67
4.3.4.2. Fluxograma da atividade avaliação - 25	68
5. Padrão de uso dos documentos.....	69
5.1. Propriedade dos documentos.....	69
5.2. Preenchimento automático do documento	69
5.3. Digitação no interior do documento	71
6. Maturidade do processo proposto.....	73
6.1. Atendimento às práticas do CMMI	73
6.1.1. Genéricas	73
6.1.2. Específicas de nível 2	75
6.1.2.1. Gerência de requisitos.....	75
6.1.2.2. Medição e análise	76
6.1.2.3. Acompanhamento e controle de projeto	77
6.1.2.4. Gerência de acordo com fornecedores.....	78
6.1.2.5. Planejamento de projeto.....	79
6.1.2.6. Gerência da configuração	80
6.1.2.7. Garantia da qualidade	81
6.1.3. Específicas de nível 3	82
6.1.3.1. Desenvolvimento de requisitos	82
6.1.3.2. Solução técnica.....	84
6.1.3.3. Integração de produto	85
6.1.3.4. Verificação	86
6.1.3.5. Validação	88
6.1.3.6. Foco no processo da organização	89

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

6.1.3.7. Definição do processo da organização.....	90
6.1.3.8. Treinamento organizacional.....	91
6.1.3.9. Gerência integrada de projeto	92
6.1.3.10. Gerência de risco	93
6.1.3.11. Análise de decisão e resolução.....	94
6.1.4. Específicas de nível 4	94
6.1.4.1. Desempenho do processo organizacional	94
6.1.4.2. Gerência quantitativa de projeto.....	95
6.1.5. Específicas de nível 5	96
6.1.5.1. Análise causal e resolução	96
6.1.5.2. Gestão do processo organizacional.....	96
7. Referencias.....	97

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

1. Introdução

1.1. Finalidade

O documento processo de desenvolvimento de sistemas embarcados tem a finalidade de definir todas o fases, atividades e artefatos proposto para o desenvolvimento um sistema embarcado.

Este processo é resultado de um trabalho de pesquisa para a conquista do título de doutor em Engenharia e Tecnologia Espaciais / Gerenciamento de Sistemas Espaciais realizado no Instituto Nacional de Pesquisa Espacial – INPE. Com o título " UMA ABORDAGEM PARA PROCESSO DE DESENVOLVIMENTO DE SISTEMA EMBARCADO QUE ATENDE O NIVEL 2 DE MATURIDADE DO CMMI-DEV." Com data de defesa marcada para o dia 10 de Fevereiro de 2015 as 10:00 horas da manhã pela doutoranda Magda Aparecida Silvério Miyashiro a ser realizada na Sala A - Prédio Rotunda do INPE, tendo como componentes da banca examinadora os seguintes doutores:

Prof. Dr. Ronaldo Arias Presidente da banca examinadora - INPE;

Prof. Dr. Mauricio Gonçalves Vieira Ferreira Orientador - INPE;

Prof. Dr. Nilson Sant'Anna Orientador - INPE;

Prof. Dr. Vanderlei Cunha Parro Convidado - Inst. Mauá Tecnologia;

Prof. Dr. Mauro de Mesquita Spinola - Convidado USP.

1.2. Escopo do documento

Este documento apresenta um processo de desenvolvimento para sistemas embarcados que contemplem as práticas requeridas pelo Nível 2 do CMMI onde:

Para viabilizar a interpretação do processo, a seção 2 apresenta o significado de cada componente e a sua representação gráfica e sua simbologia e apresenta também o processo cíclico. A seção 3 apresenta o detalhamento e a descrição das fases e descrição de todos os componentes do processo cíclico, a seção 4 apresenta o fluxograma geral sugerido de utilização do processo, a seção 5 apresenta a estrutura automatizada para o uso dos *templates* com os recursos do word e a seção 6 apresenta como o processo cíclico atende a cada prática do modelo.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

1.3. Envolvidos

Elaboração

Ms: Magda Aparecida Silvério Miyashiro

Estudo de Caso

INPE – Laboratório de *hardware* e *software* da Divisão de Desenvolvimento de Sistemas de Solo (DSS) da coordenação de Engenharia e Tecnologia Espaciais (ETE) do INPE (Em fase inicial de utilização).

IMT- Instituto Mauá de Tecnologia, Núcleo de Sistemas Eletrônicos Embarcados – NSEE (Em fase inicial de utilização).

Orientação

Profº. Dr. Mauricio G.V. Ferreira

Profº. Dr. Nilson Sant’anna.

1.4. Posicionamento do processo proposto no padrão ECSS

O Padrão utilizado pelo Instituto Nacional de Pesquisas Espaciais – INPE, na realização de uma missão espacial é o European Cooperation for Space Standardization (ECSS), portanto para o desenvolvimento deste processo foi definido como requisito especial que ele esteja de acordo com tal padrão.

O Padrão ECSS em cada ramo suas disciplinas e requisitos são cobertos por normas específica e apoiada por manuais e memorandos Técnica quando necessário:

- **Normas** - Documento especificamente para uso em acordos entre empresas para a execução de atividades relacionadas à área espacial.
- **Manual** - Documento não normativo que fornece informações mais detalhadas como orientação, diretrizes, dados técnicos, pareceres ou recomendações sobre como implementar as atividades relacionadas à área espacial, onde existem dois tipos de manual que são as orientações e boas práticas e o tratamento de dados.
- **Memorandos Técnicos** - Documento não normativo que fornece informações úteis para a comunidade espacial sobre um tema específico.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

1.4.1. Organização das disciplinas do padrão ECSS

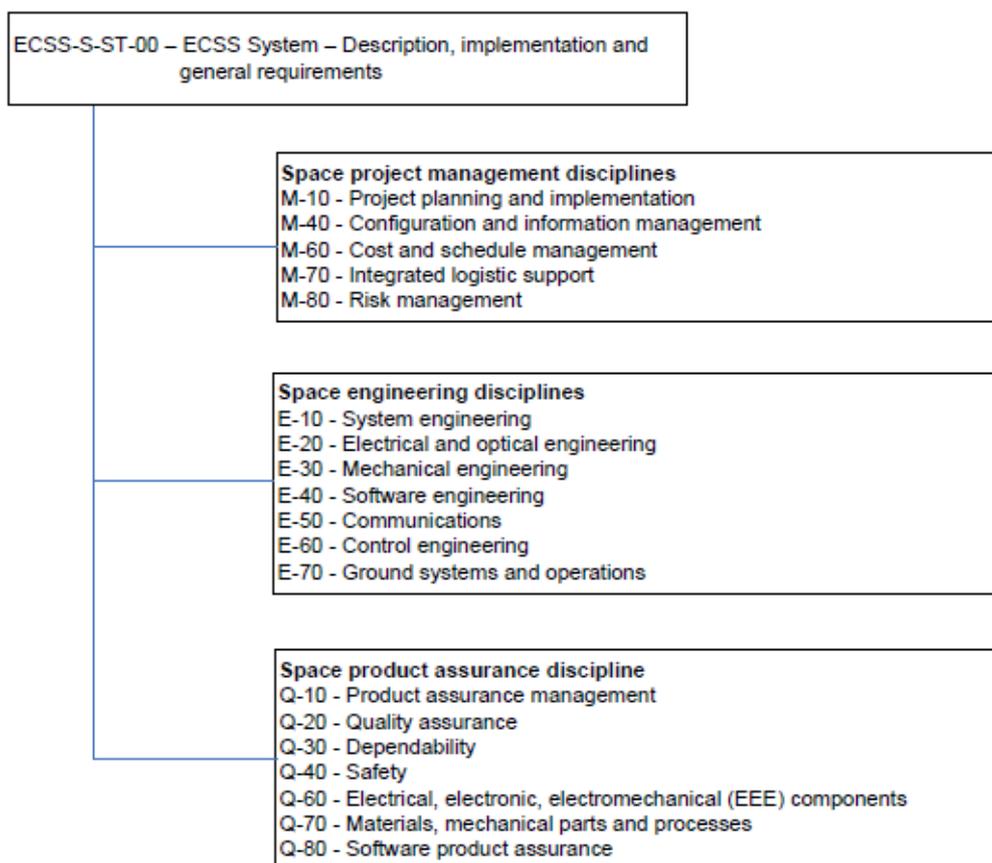


Figura 1.1 - Organização das disciplinas do padrão ECSS

As normas que tratam de engenharia de *software* e qualidade de *software* são ECSS-E-ST-40 - Space Engineering Software, ECSS-Q-ST-80 - Space Product Assurance - Software Product, respectivamente.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

2. Representação

2.1. Como entender a disposição dos componentes do processo

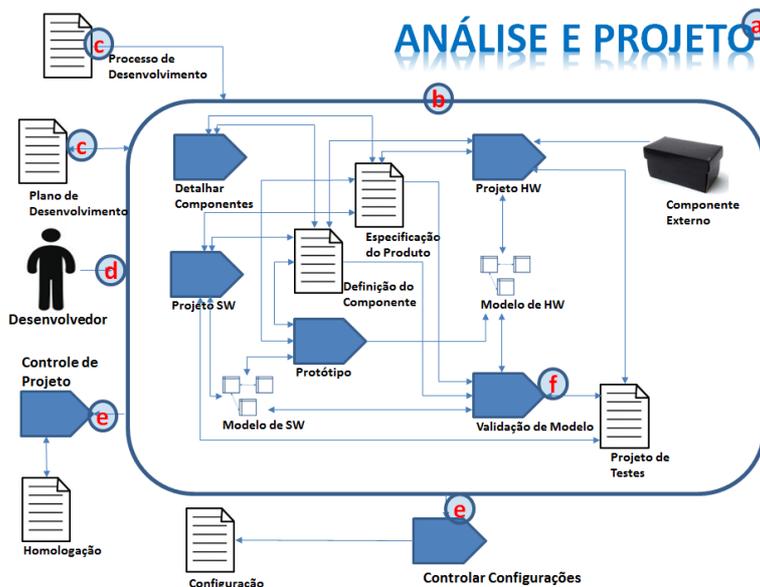


Figura 2.1 - Representação do uso da simbologia

- a) Nome da fase: utilizado para identificação.
- b) Retângulo arredondado: utilizado para definir o escopo das atividades realizadas naquela fase, todos os componentes do processo que estiver dentro do retângulo representam atividades ou artefatos produzidos ou manipulados naquela fase, todos os componentes do processo que estão fora do retângulo são referentes a atividades e artefatos que são utilizados ou realizados em todas as fases. Obs.: Os componentes do processo que estão localizados fora do retângulo e não tem associação com o retângulo, são produzidos pela atividade que está associada ao componente do processo. Os componentes do processo que estão associados ao retângulo devem ser entendidos como associados a todos os artefatos que podem ter associação.
- c) Documento associado diretamente ao retângulo: utilizado como referência durante a fase, observar o sentido da seta, se seta única saindo do documento em direção ao retângulo documento de leitura, sentido bidirecional, o documento pode ser atualizado durante a fase.
- d) Ator associado ao retângulo: todas as atividades internas ao retângulo são realizadas por ele.
- e) Atividade associada diretamente ao retângulo: são realizadas em todas as fases.
- f) Atividade associada a documento: a realização da atividade resulta no documento associado.
- g) Considerações: As associações devem existir entre ator-atividade, atividade-documento,

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

pois toda atividade tem um resultado e todo documento é resultado de alguma atividade e toda atividade é realizada por algum ator.

h) Não existem associações entre ator-documento, documento-documento, atividade-atividade, por exemplo, o ator sempre atualiza um documento através de uma atividade.

2.2. Simbologia utilizada

A simbologia utilizada para padronizar a representação dos componentes do processo está apresentada a figura 2.2.



Figura 2.2 - Simbologia do Processo

2.3. Processo cíclico

Ciclo de desenvolvimento é a realização sequencial de todas as fases e atividades do ciclo de vida. Cada ciclo de desenvolvimento terá como resultado um componente de um sistema embarcado e todos os documentos produzidos durante seu desenvolvimento. Quando um sistema embarcado for composto por diversos componentes, deverá ter diferentes ciclos de desenvolvimento, sendo um para cada componente de sistemas embarcado, conforme apresentado na Figura 2.3.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

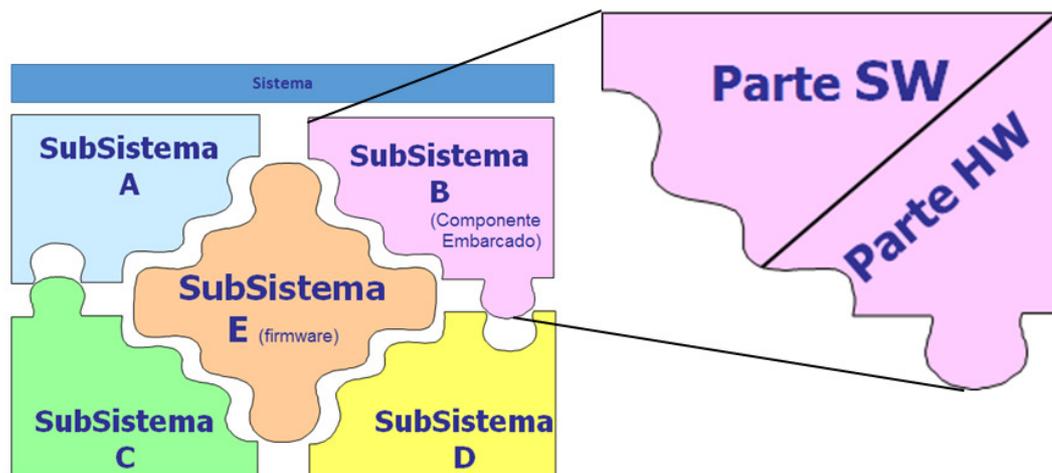


Figura 2.3 - Representação das partes de um sistema

A estrutura do processo de desenvolvimento é composta por várias fases que devem ser realizadas de forma cíclica e sequenciada.

Cíclica: o processo é denominado cíclico, pois cada componente (*hardware + software*) do sistema embarcado a ser desenvolvido segundo suas especificações deve realizar todas as suas fases e atividades, formando assim um ciclo de desenvolvimento de componentes de sistema embarcado e para cada componente será elaborado um plano de acordo com o processo e serão realizadas todas as fases e atividades planejadas.

Sequenciais: as fases e atividades devem ser realizadas sequencialmente, onde uma fase inicia após a anterior ser concluída e homologada, utilizando como referência o modelo clássico.

Nessa abordagem está sendo proposto um processo para o desenvolvimento de sistemas embarcados com foco na gestão do projeto e nas boas práticas de desenvolvimento de sistema, passando por um ciclo de desenvolvimento completo, isto é, todas as fases do ciclo de vida apresentadas na Figura 2.4.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

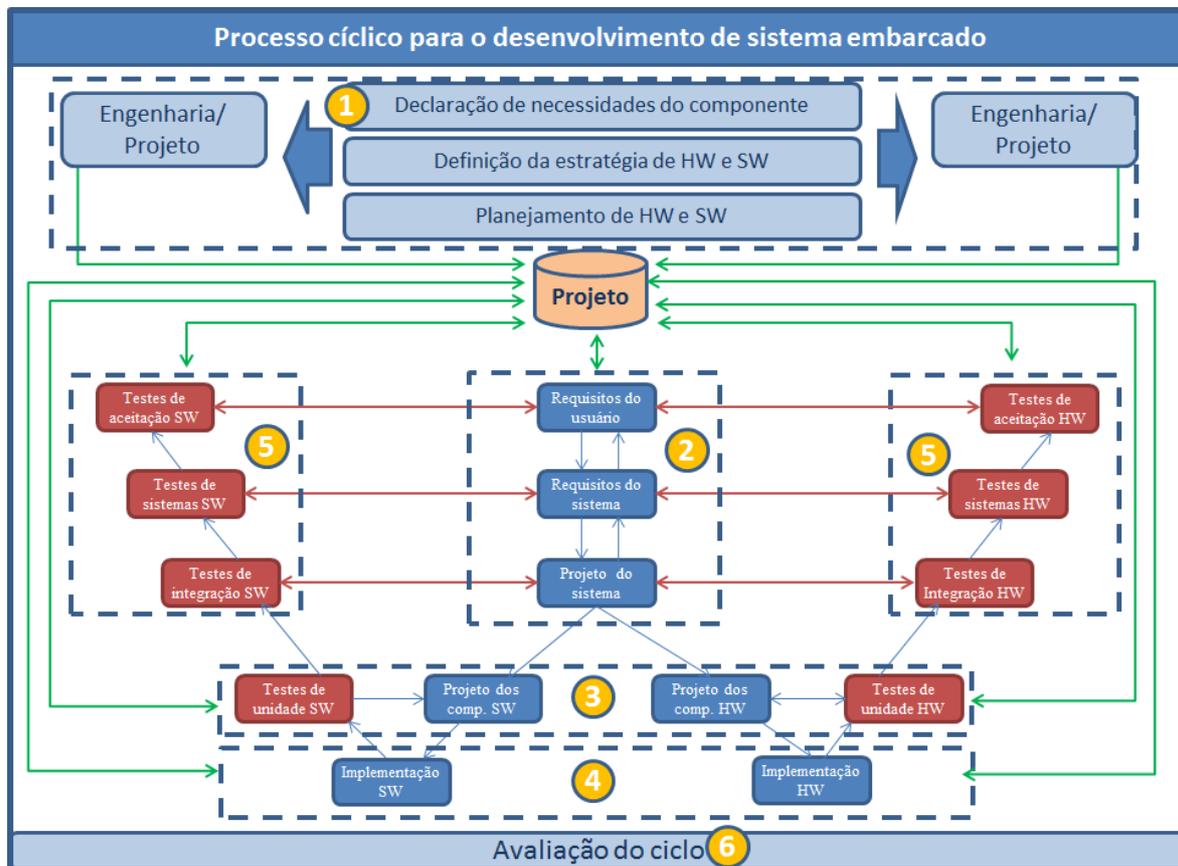


Figura 2.4 - Processo cíclico.

Cada ciclo deve estabelecer as relações de trabalho, definir e distribuir os papéis aos membros da equipe, bem como definir metas, estratégias e planos de trabalho para todas as fases do processo. Quando oportuno, deve realizar modificações no processo, de acordo com o resultado da avaliação do ciclo.

Na estrutura apresentada na Figura 2.5, as fases são o agrupamento de atividades no ciclo de vida; as atividades são procedimentos que devem ser realizados em cada fase; e as ações são detalhamentos da realização de cada atividade especificando a sequência a ser seguida, os desvios quando pertinentes, as saídas esperadas e os artefatos produzidos.

O processo é organizado em fases, compostas por atividades e realizadas por meio de ações governadas por procedimentos e documentos (modelos) que resultam em artefatos comuns, os quais compõem o produto de sistema embarcado (programas e sua documentação).

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

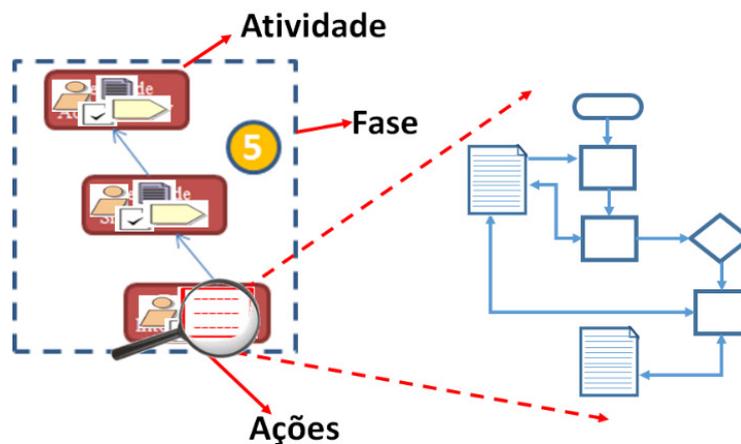


Figura 2.5 - Componentes do processo cíclico

3. Descrição das fases

3.1. Descrição dos modelos de documentos resultados das atividades das fases do processo

- **Processo de desenvolvimento:** documento contendo um conjunto de atividades, organizadas e formalizadas que produzam saídas (artefatos) predefinidos com o objetivo de auxiliar na construção de componentes com qualidade prevista.
- **Homologação:** documento resultante das atividades de acompanhamento e validação de cada fase do projeto, que registra a checagem de todos os artefatos produzidos em cada fase, de acordo com as especificações e as expectativas das saídas esperadas, bem como o comprometimento e os resultados dos envolvidos. Esse documento é produzido na fase de engenharia e alimentado nas demais fases, podendo ser desconsiderado quando suas informações forem inseridas no plano de desenvolvimento. Esse documento é adaptado do artefato do rup rup_pacpln.
- **Configuração:** documento resultante do planejamento, do controle e dos registros dos itens de configuração, seja para o ambiente de desenvolvimento e uso, linguagens, seja para todos os artefatos produzidos durante o processo, incluindo o próprio documento de configuração. Esse documento é produzido na fase de engenharia e alimentado nas demais fases, de acordo com as mudanças. Esse documento é adaptado do artefato do rup rup_cmpln.
- **Plano de desenvolvimento:** documento contendo o resultado do conjunto de atividades organizadas e formalizadas com atribuição de sequências de tempo, recursos e dependências entre as tarefas para o desenvolvimento do *hardware* e do *software* do componente embarcado. Esse documento é adaptado do artefato do rup rup_sdpln.
- **Especificação do produto:** o documento é a saída de diferentes atividades realizadas ao

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

longo do processo, relacionadas à descrição e à definição do produto. Esse documento reúne o conjunto de informações, especificações e necessidades do produto completo a ser construído (sistema embarcado). É de grande importância, mas opcional, pois o processo cíclico direciona o desenvolvimento de componente de forma individual. Entende-se que a especificação do produto é um documento existente em instância superior de desenvolvimento. Esse documento é adaptado do artefato do rup *rup_vision*.

- **Especificação do componente:** esse documento é a saída de diferentes atividades realizadas ao longo do processo, atividades relacionadas à descrição, à definição e ao detalhamento do componente (*hardware* e *software*). Reúne o conjunto de informações, especificações e necessidades do componente a ser produzido e inserido no sistema embarcado. Esse documento é adaptado do artefato do rup *rup_cmpln*.
- **Projeto de testes:** documento resultante da elaboração de procedimentos sistemáticos para realizar os testes do componente (*hardware* e *software*). Planejado detalhadamente de acordo com o fluxo de trabalho e com a evolução do produto em todas as fases pertinentes. Esse documento é adaptado do artefato do rup *rup_tstpln*.
- **Componentes externos:** documentação completa dos componentes externos ao projeto que poderão ser utilizados. São considerados externos os componentes construídos em outros projetos ou adquiridos do mercado, como documentos institucionais de outros projetos, especificações do sistema e documentação de componentes externos.
- **Modelo de SW:** representação gráfica resultante da atividade de construção do modelo de *software* que explique as características e o comportamento da parte *software* do componente a ser desenvolvido, bem como sua integração com o *hardware* e sua inserção no sistema embarcado. O conteúdo é inserido no documento de especificação de produto e de componente.
- **Modelo de HW:** representação gráfica resultante da atividade de construção do modelo de *hardware* que explique as características e o comportamento da parte *hardware* do componente a ser desenvolvido, bem como sua integração com o *software* e sua inserção no sistema embarcado. O conteúdo é inserido no documento de especificação de produto e de componente.
- **Software:** programa ou conjunto dos programas construídos de acordo com as especificações e os documentos resultantes das atividades anteriores.
- **Hardware:** conjunto de componentes eletrônicos construídos de acordo com as especificações e os documentos resultantes das atividades anteriores.
- **Produto final:** componente embarcado produzido por meio da integração do *hardware* e do *software*.

3.2. **Atividades comuns(FC):** são as atividades de controle realizadas em todas as fases do processo, sempre iniciadas na fase de engenharia. O item 1 da Figura 2.3 e individualizado na figura 3.2 apresenta seu posicionamento na estrutura do processo. As atividades comuns nascem na fase de engenharia e evoluem nas demais fases, e sua

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

representação gráfica no ciclo de vida do processo está atrelada à atividade de engenharia.

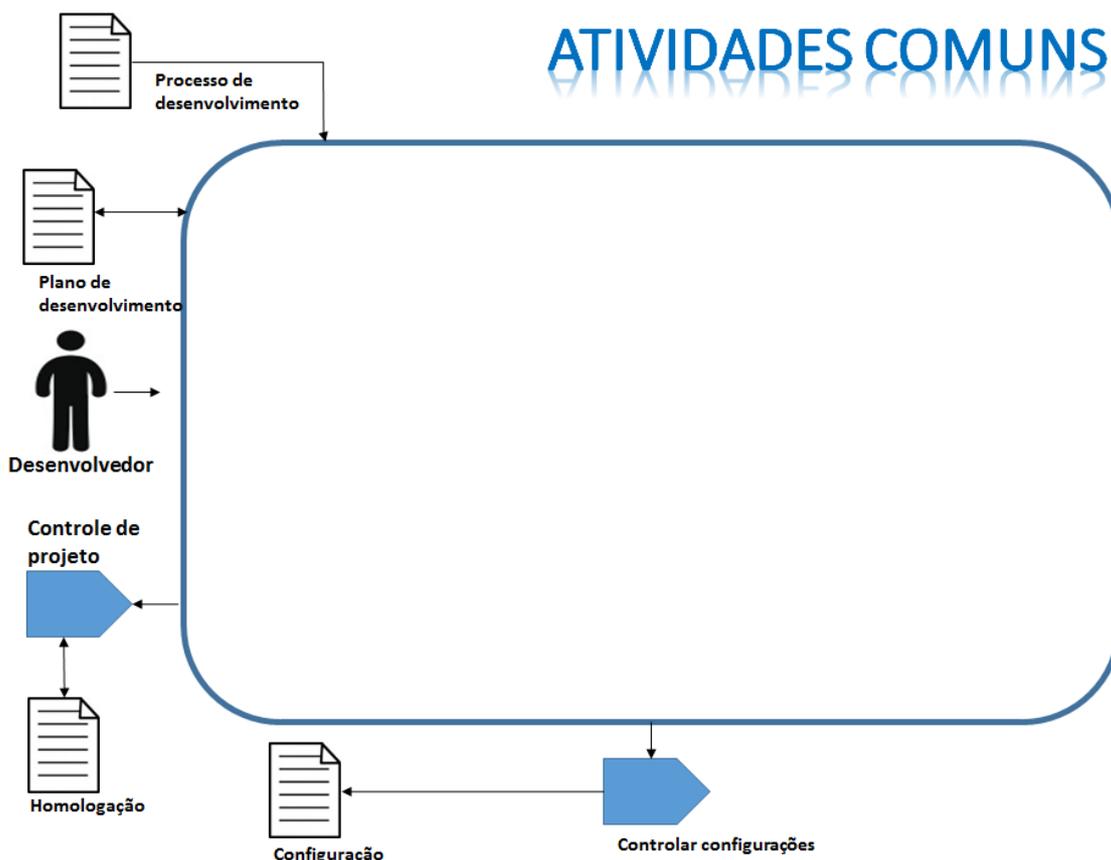


Figura 3.1 - Atividades comuns - Fase

As atividades comuns ao processo são aquelas que devem ser realizadas em todas as fases do desenvolvimento dos componentes. Consistem em atividades de controle do projeto e de qualidade, devendo se iniciar no mesmo momento em que nasce o projeto. Tais atividades devem ser descritas de maneira formal e cumpridas de acordo com o que foi definido, conforme mostrado na Figura 3.1.

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Atividades:

- **Controle do projeto – 1:** nesta atividade devem-se realizar ações para o monitoramento do projeto, o acompanhamento das ações realizadas, a avaliação do desempenho do projeto de acordo com o planejado, a realização de mudanças necessárias de acordo com o contexto do projeto.
- **Controlar configurações – 2:** nesta atividade devem-se definir a estrutura e a configuração necessárias para o desenvolvimento do projeto, identificar e descrever todos os itens que devem ser controlados, todas as mudanças que os artefatos produzidos podem sofrer, bem como todos os responsáveis por elas.

Artefatos Envolvidos:

- Processo de desenvolvimento.
- Plano de homologação.
- Configuração.
- Plano de desenvolvimento.

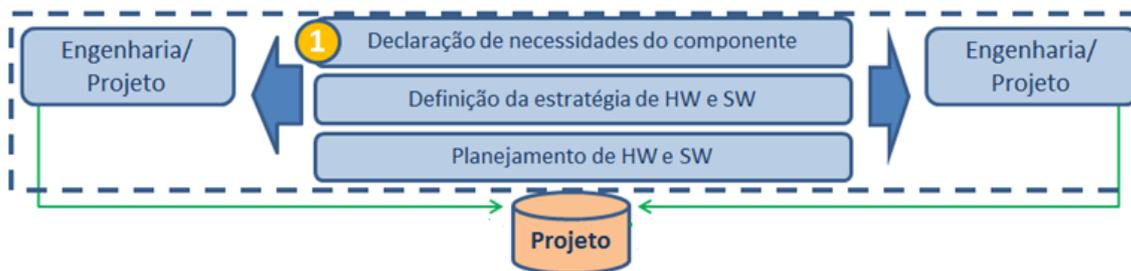


Figura 3.2 - Engenharia - Ciclo

3.3. **Engenharia(F1):** realização de atividades para o entendimento do produto completo, o estudo e o planejamento do projeto, a viabilidade e a preparação do ambiente. O item 1 da Figura 2.3 e individualizado na figura 3.2 apresenta seu posicionamento na estrutura do processo.

Na fase de engenharia, o componente é descrito de forma suficiente para permitir o planejamento de todas as atividades que serão realizadas. É definido o ambiente a ser utilizado para todo o desenvolvimento, além dos mecanismos para medição e controle do projeto (ver Figura 3.3).

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

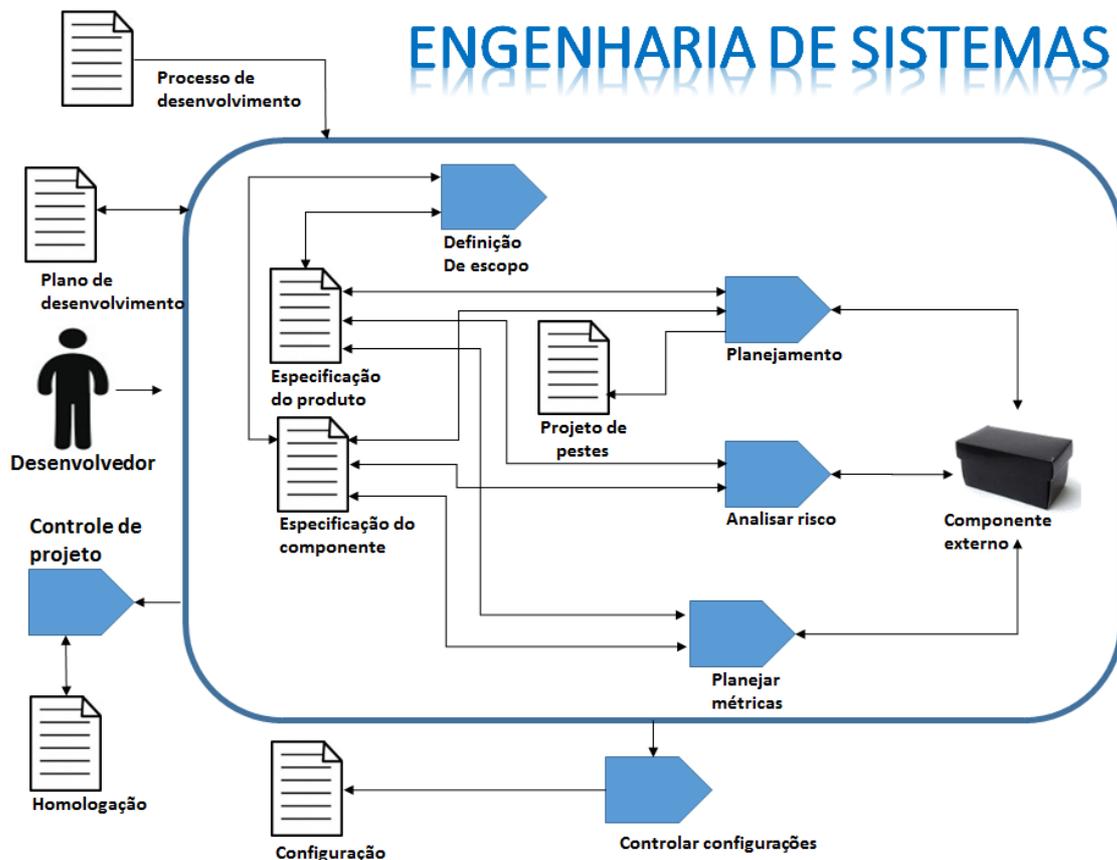


Figura 3.3 - Engenharia - Fase

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades:

- **Definição de escopo – 3:** atividade realizada para delimitar a fronteira das funcionalidades a serem realizadas pelos componentes e sua inserção no contexto. Podem ser apresentadas também as funcionalidades do produto completo (sistema). O documento de especificação do produto é opcional, uma vez que o processo é aplicado a cada componente.
- **Planejamento – 4:** atividade de adaptação, preparação, organização e estruturação do processo para o desenvolvimento do componente, de acordo com as diretrizes do processo. Oferece parâmetros para acompanhamento, medição e tomadas de decisão.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **Analisar risco – 5:** verificação dos pontos críticos que possam vir a apresentar imperfeição ou falha durante a realização do projeto, a qual pode comprometer o desenvolvimento do componente. É o estudo de cada fase de desenvolvimento feito para identificar possíveis riscos e evitá-los. Entre suas ações estão incluídas: identificação, análise, classificação, planejamento, monitoramento, rastreamento e controle de riscos.
- **Planejar métricas – 6:** atividade realizada para a identificação e a compreensão das medidas que devem ser contempladas como mecanismos de controle e avaliação. Nela devem ser definidas as métricas de qualidade do processo de desenvolvimento do componente, porém de forma especial, para os componentes de sistemas embarcados. Deve ser planejada com muita precisão a medição da qualidade do produto a ser desenvolvido. Nesta atividade devem-se identificar todos os requisitos do componente a ser desenvolvido e medido ao longo de sua construção.

Artefatos Envolvidos:

- Especificação do produto.
- Definição do componente.
- Projeto de testes.
- Componentes externos.

3.4. **Análise de requisitos (F2):** detalhamento dos requisitos do produto e dos componentes a serem desenvolvidos. O item 2 da Figura 2.3 e individualizado na Figura 3.5 apresenta seu posicionamento na estrutura do processo.

Na fase de requisitos o componente é descrito de maneira completa e com especificações de regras detalhadas, em que se devem destacar, de forma clara, os requisitos do projeto, suas restrições, entre outras informações que não podem ser omitidas, pois estabelecem critérios importantes durante todas as etapas futuras (Figura 3.4).

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

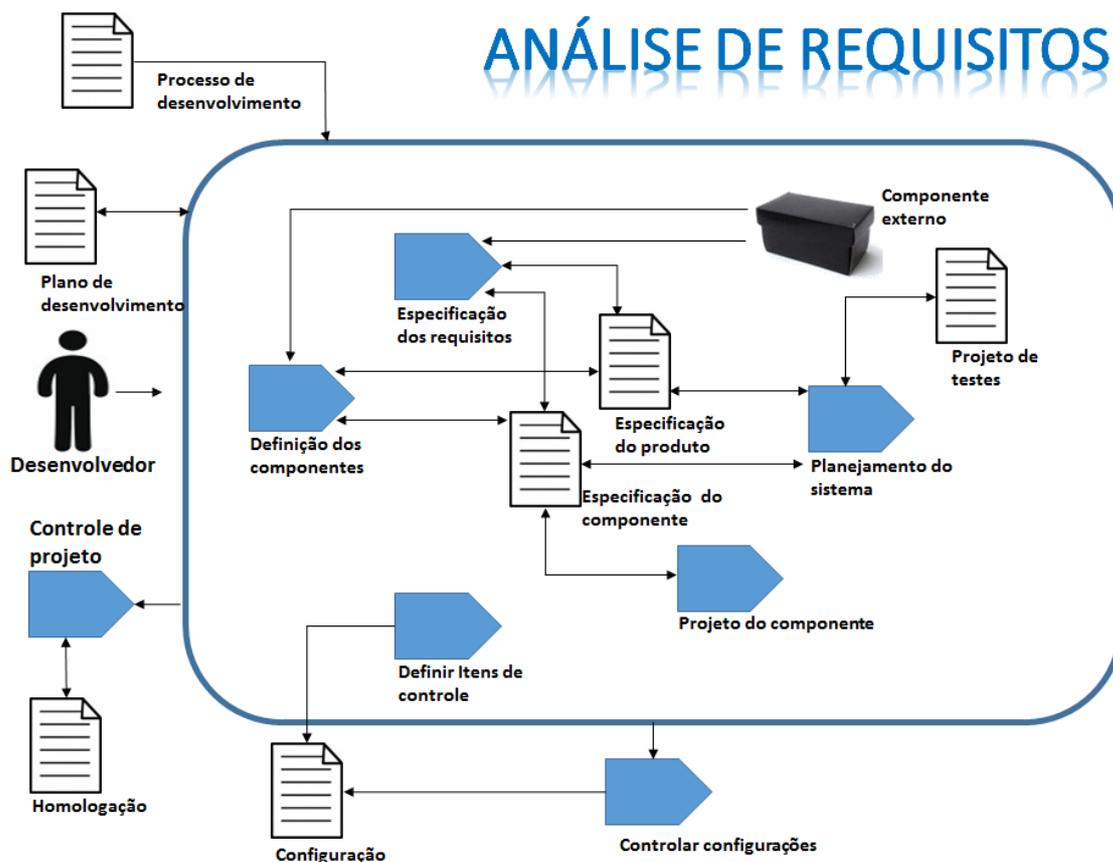


Figura 3.4 - Requisitos - Fase

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades:

- **Especificação dos requisitos – 7:** definição detalhada dos requisitos do produto a ser construído.
- **Definição dos componentes – 8:** detalhamento do componente que será desenvolvido.
- **Planejamento do sistema – 9:** identificação, individualização e especificação das tarefas atribuídas ao *software* e ao *hardware* do componente a ser construído.
- **Projeto do componente – 10:** elaboração do projeto do componente

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

(*hardware e software*) a ser construído.

- **Definir itens de controle – 11:** de acordo com os requisitos, devem-se identificar todos os itens do produto/componente e itens de estrutura do ambiente que deverão ser controlados pela gerência da configuração, além de indicar a forma de controle e as ferramentas a serem utilizadas.

Artefatos Envolvidos:

- Especificação do produto.
- Definição do componente.
- Projeto de testes.
- Componentes externos.
- Configuração.



Figura 3.5 - Requisitos do produto - Ciclo

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

3.5. **Análise e projeto (F3):** definição dos componentes e divisão das funcionalidades a serem implementadas nas partes de *hardware* e de *software* do componente, estudo de reutilização de componentes ou utilização de componentes externos, elaboração e validação de modelos e protótipos. O item 3 da Figura 2.3 e individualizado na Figura 3.7 apresenta seu posicionamento na estrutura do processo.

É nessa fase que se identificam as funcionalidades do sistema que serão implementadas em *hardware* e as que serão implementadas em *software*. Nela também se avalia a utilização de componentes externos ao projeto (Figura 3.6).

Ainda nessa fase são produzidos os modelos individuais de *hardware* e de *software*. É importante o uso de técnicas e ferramentas para prototipação e simulação de arquitetura de *software* e de *hardware*, pois, por meio dos modelos produzidos, poderão ser estudados os resultados obtidos. Se necessário, pode-se rever o critério de separação das funcionalidades, realizando simulações até que se produza um modelo que atenda aos requisitos estabelecidos.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

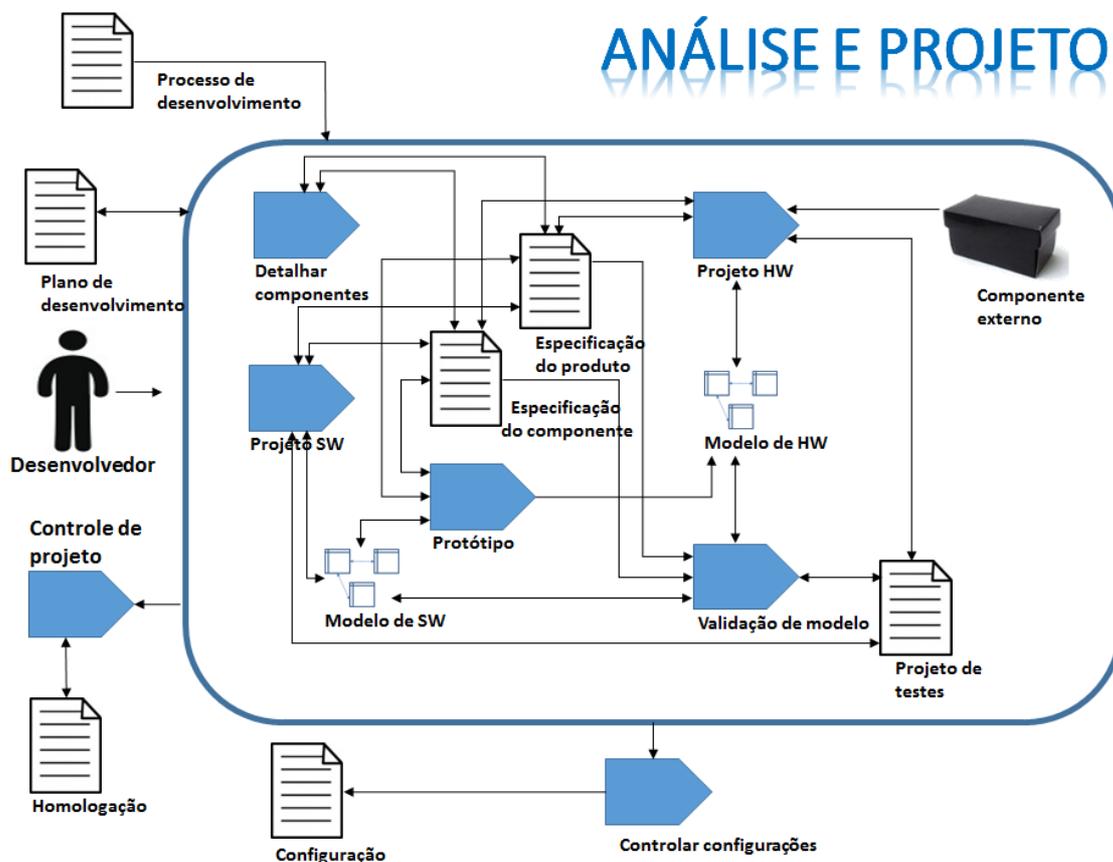


Figura 3.6 - Análise e projeto - Fase

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades

- **Detalhar componentes – 12:** detalhamento dos itens que serão utilizados pelos componentes de *hardware* e de *software* de acordo com a individualização das tarefas. Esta atividade vai alimentar os documentos de especificação do produto e definição dos componentes.
- **Projeto HW – 13:** desenho da arquitetura do *hardware* para a realização da prototipação e posterior construção de acordo com a arquitetura do *software*. Nesta atividade são feitas a análise e a definição das ferramentas de modelagem e prototipação a serem utilizadas. Ela vai alimentar os documentos de especificação do produto e definição dos componentes.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **Projeto SW – 14:** desenho da arquitetura do *software* para a realização da prototipação e posterior construção de acordo com a arquitetura do *hardware*. Nesta atividade são feitas a análise e a definição das ferramentas de modelagem e prototipação a serem utilizadas. Ela vai alimentar os documentos de especificação do produto e definição dos componentes.
- **Protótipo – 15:** construção do protótipo dos modelos de *hardware* e de *software* elaborados a serem validados.
- **Validação de modelo – 16:** validação dos modelos e protótipos construídos em atendimento aos requisitos de acordo com os planos.

Artefatos Envolvidos:

- Especificação do produto
- Definição do componente
- Projeto de testes
- Componentes externos
- Modelo de SW
- Modelo de HW



Figura 3.7 - Análise e projeto - Ciclo

3.6. **Implementação e integração (F4):** nesta fase são realizadas atividades independentes de implementação do *hardware* e do *software* do componente. Em seguida, conclui-se sua integração e são realizados testes independentes e testes integrados. O item 4 da Figura 2.3 e individualizado na Figura 3.10 apresenta seu posicionamento na estrutura do processo.

A fase de implementação deve ser realizada especialmente quando já se atingiu o(s) modelo(s) ideal(is) que atenda(m) aos requisitos do componente. Nesta fase cada um dos componentes (*hardware* e *software*) deve ser produzido segundo os modelos individuais que foram validados nas fases de modelagem e design. Também nesse momento devem ocorrer testes preliminares

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

das partes do componente. Após a validação das partes dos componentes, eles devem ser integrados para formar o componente especificado e validado (Figuras 3.8 e 3.9).

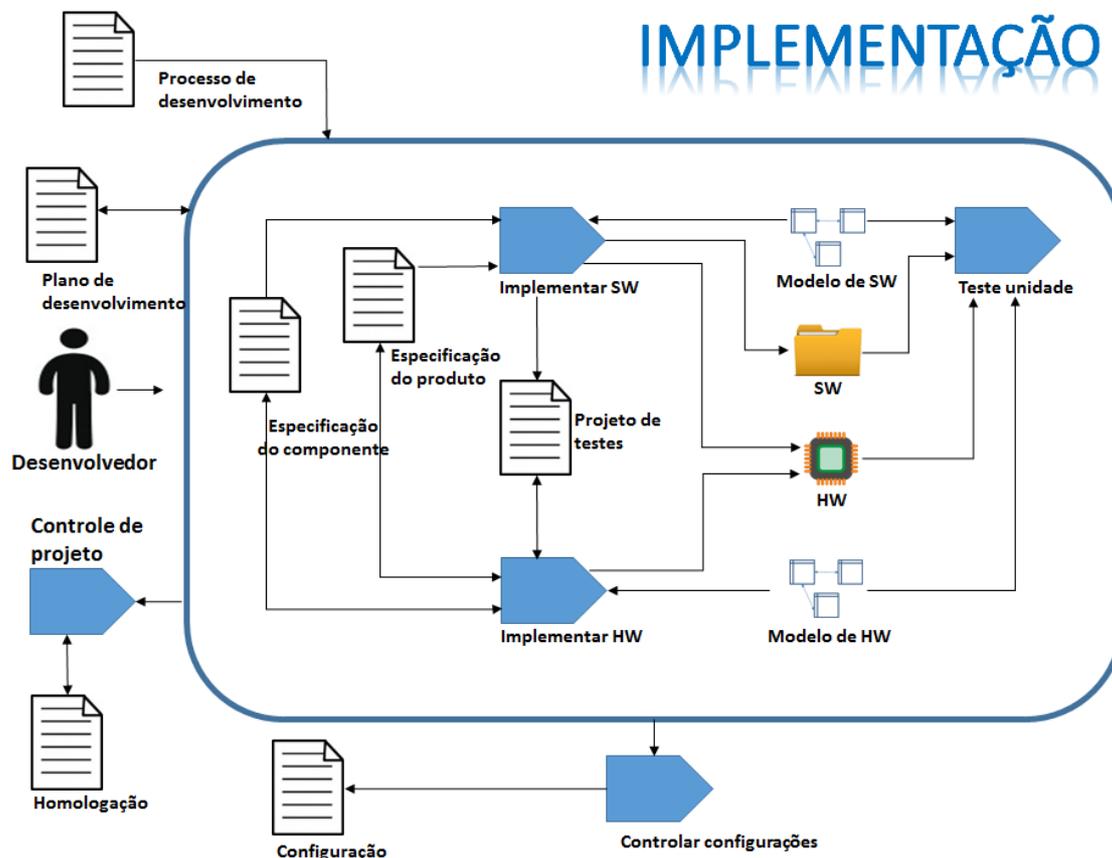


Figura 3.8 - Implementação - Fase

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades:

- **Implementar SW– 17** – Construção do *software* de acordo com a especificação e modelos validados.
- **Implementar HW– 18** – Produção ou aquisição do *hardware* que atenda as especificações validadas.
- **Teste de Unidade– 19** – Atividades de teste e conferência de cada parte do

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

componente (*hardware e software*) de acordo com o plano de testes.

Artefatos Envolvidos:

- Especificação do produto
- Definição do componente
- Projeto de testes
- Componentes externos
- Modelo de SW
- Modelo de HW
- *Software*
- *Hardware*

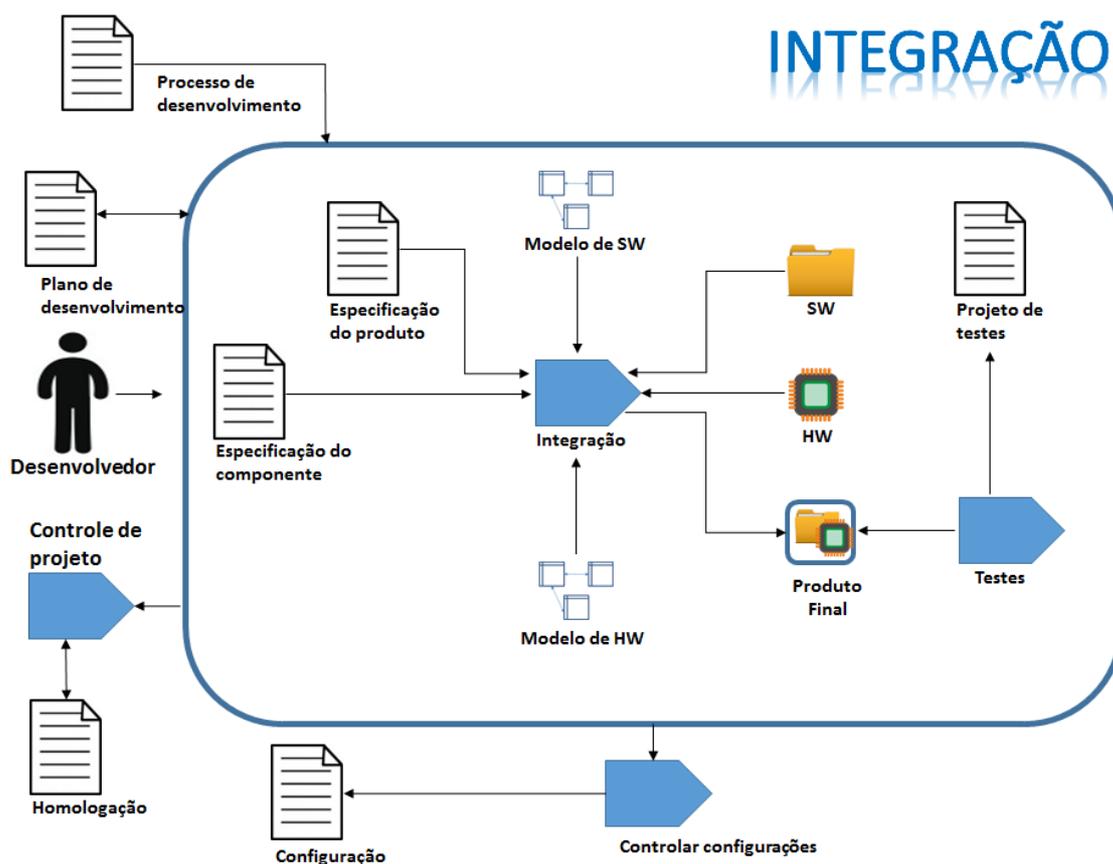


Figura 3.9 - Integração - Fase

Atores:

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades:

- **Integração-SW- HW- 20** – Integração (Conexão) das funcionalidades da parte *hardware* e da parte *software* do componente;
- **Teste - 21** – Atividades de teste e conferência de cada parte do componente e do componente integrado bem como a simulação do seu ambiente de uso de acordo com o plano de testes.

Artefatos Envolvidos:

- Especificação do produto
- Definição do componente
- Projeto de testes
- Modelo de SW
- Modelo de HW
- *Software*
- *Hardware*
- Produto-final



Figura 3.10 - Implementação e integração - Ciclo

3.7. **Verificação e validação do Sistema (F5)** - nesta fase executam-se os testes de verificação e validação do componente, bem como conferência, acompanhamento e homologação. Na Figura 2.3 - item 5 e individualizado na Figura 3.12, apresenta seu posicionamento na estrutura do processo.

A verificação e validação devem buscar de forma exaustiva e completa todas as condições ao qual o componente será submetido, seguindo os planos de teste que foram produzidos ao longo do desenvolvimento. Devem ser validados os comportamentos do componente integrado, das partes do componente e do seu comportamento inserido no sistema ou no ambiente em que ele

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

irá funcionar. É importante o uso de ferramentas e técnicas de testes bem como simulações. (Figura 3.11).

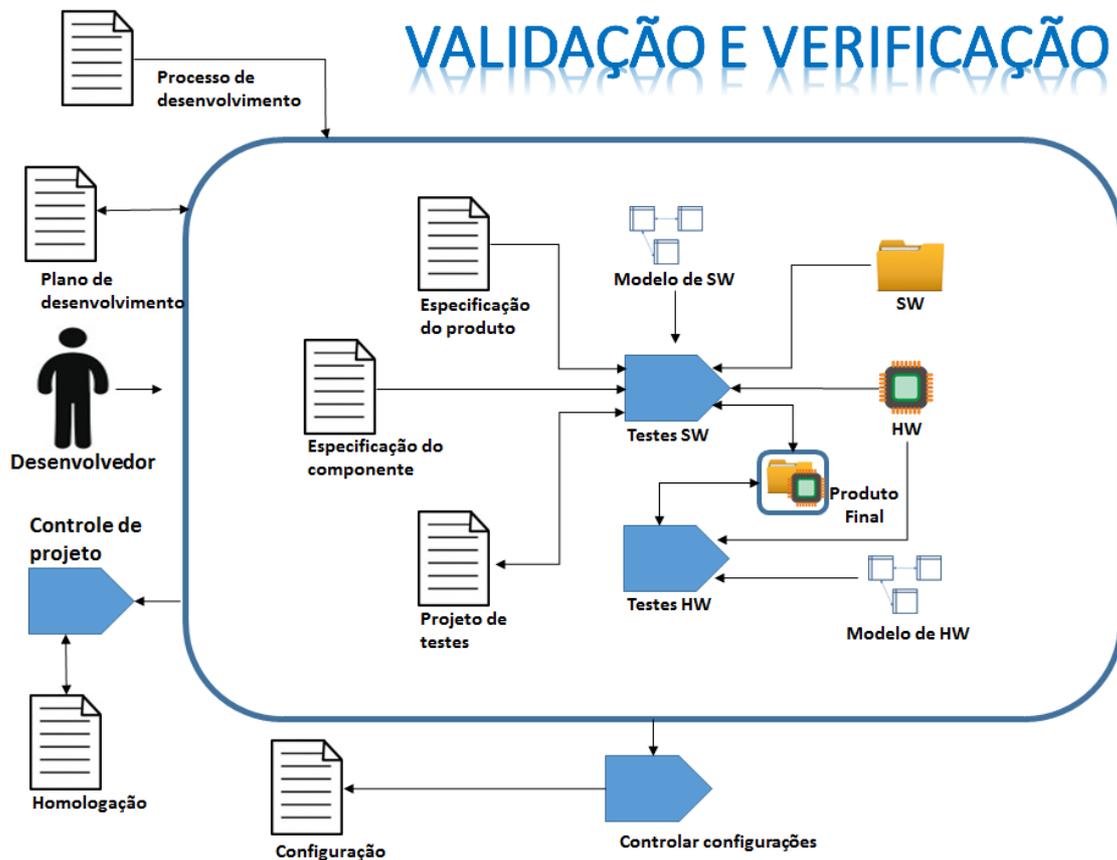


Figura 3.11 - Validação e verificação - Fase

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades:

- **Testes SW - 22** – Testes de verificação e validação do *software* do componente integrado e da sua integração e atendimento aos requisitos do *hardware*, de acordo com o plano de testes.
- **Testes HW - 23** – Testes de verificação e validação do *hardware* do componente integrado e da sua integração e atendimento aos requisitos do *software*, de acordo

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

com o plano de testes.

Artefatos Envolvidos:

- Especificação do produto
- Definição do componente
- Projeto de testes
- Modelo de SW
- Modelo de HW
- *Software*
- *Hardware*
- Produto-final

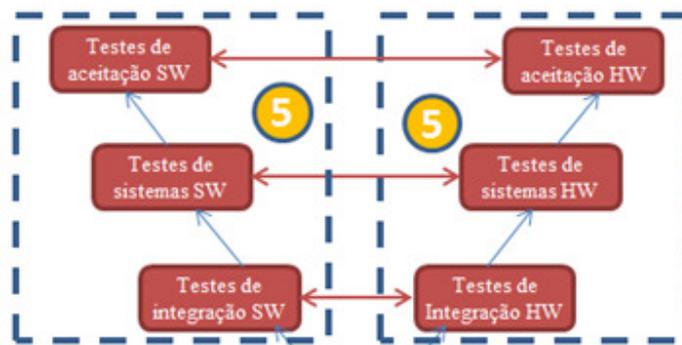


Figura 3.12 - Verificação e validação - Ciclo

3.8. **Avaliação do ciclo (F6):** análise das atividades desenvolvidas no ciclo, ajustes e adaptação para o próximo ciclo (Item 6 da Figura 2.3 e individualizado na Figura 3.14).

Na fase de avaliação do ciclo deve-se congelar e integrar (reunir) todos os artefatos produzidos durante o processo de desenvolvimento, elaborando um livro (pasta) único com todos os documentos do projeto, além de documentos externos ao projeto que sejam relevantes. O produto a ser entregue incluirá não apenas o componente produzido, mas também toda a documentação resultante de sua produção. Deve-se realizar também um documento de avaliação do ciclo de desenvolvimento no qual sejam apresentadas todas as impressões deixadas no andamento do projeto, com possíveis sugestões de soluções (Figura 3.13).

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

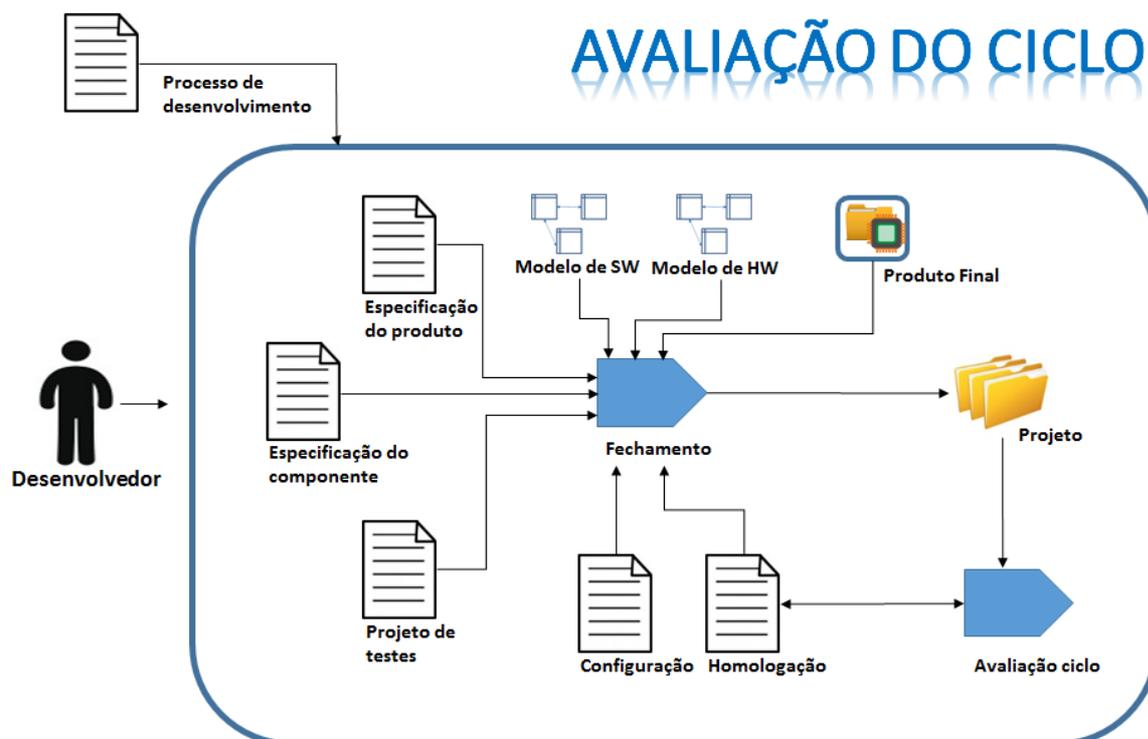


Figura 3.13 - Avaliação do ciclo - Fase

Atores:

- **Desenvolvedor** – É responsável por realizar as atividades definidas no processo. Ele pode assumir diferentes papéis durante o processo dependendo da atividade que será realizada.

Atividades:

- **Fechamento – 24** – conclusão e fechamento de todas as atividades do ciclo de desenvolvimento do componente, bem como a elaboração do livro do projeto.
- **Avaliação do ciclo – 25** – análise e avaliação de todas as atividades realizadas durante o ciclo, produtividade e qualidade do processo e do produto, e sugestões de adaptações e melhoria para os próximos ciclos.

Artefatos Envolvidos:

- Especificação do produto
- Definição do componente
- Projeto de testes

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- Configuração
- Modelo de SW
- Modelo de HW



Figura 3.14 - Avaliação do ciclo - Ciclo

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4. Fluxogramas

4.1. Orientações para o entendimento do fluxograma

A Figura 4.1 apresenta a notação utilizada para a elaboração do fluxograma geral do processo, representado nas Figuras 4.2 e 4.3. É preciso considerar que:

- O fluxo sequencial abaixo é uma sugestão para a realização do processo de desenvolvimento de componente embarcado.
- As ações descritas à direita da barra de sincronismo são realizadas sequencialmente.
- A ação descrita à esquerda da barra de sincronismo deve ser realizada em todas as atividades.
- As siglas F1, F2, F3, F4, F5, F6 e FC representam os códigos das fases em que as atividades são realizadas.

A notação utilizada é:

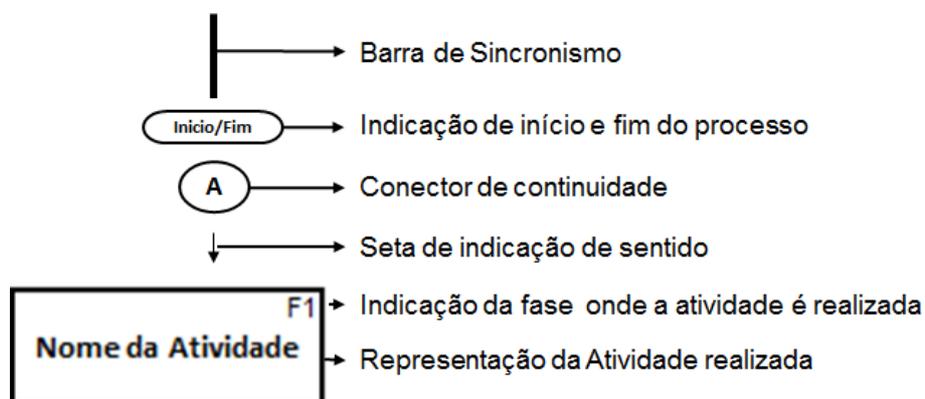


Figura 4.1 - Representação gráfica do fluxograma

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.2. Fluxograma geral do processo

Este fluxograma sugere de forma geral as atividades principais que devem ser realizadas contemplando todas as fases do processo (parte 1 e parte 2).

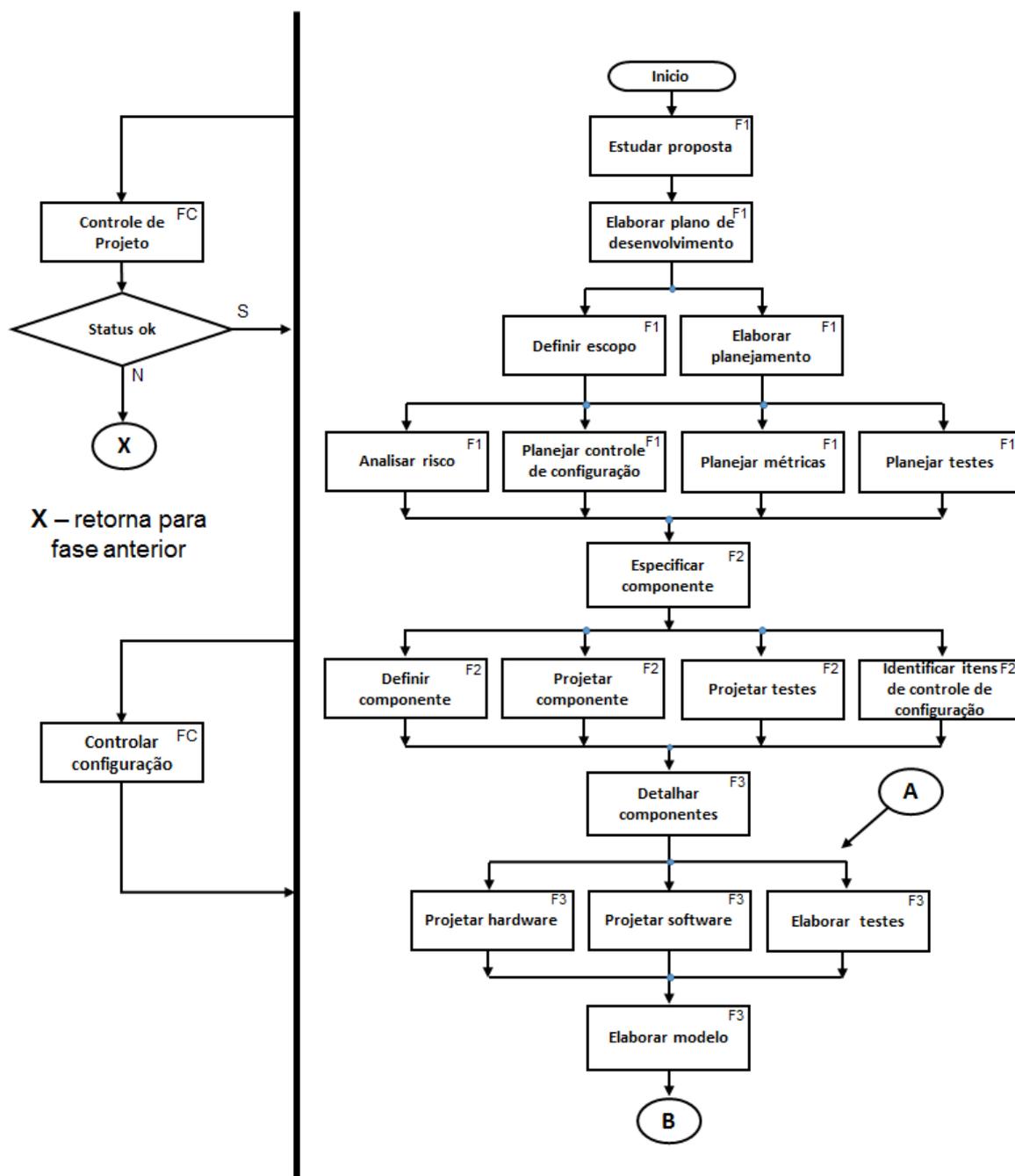


Figura 4.2 - Fluxograma geral de realização do processo - Parte-1

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

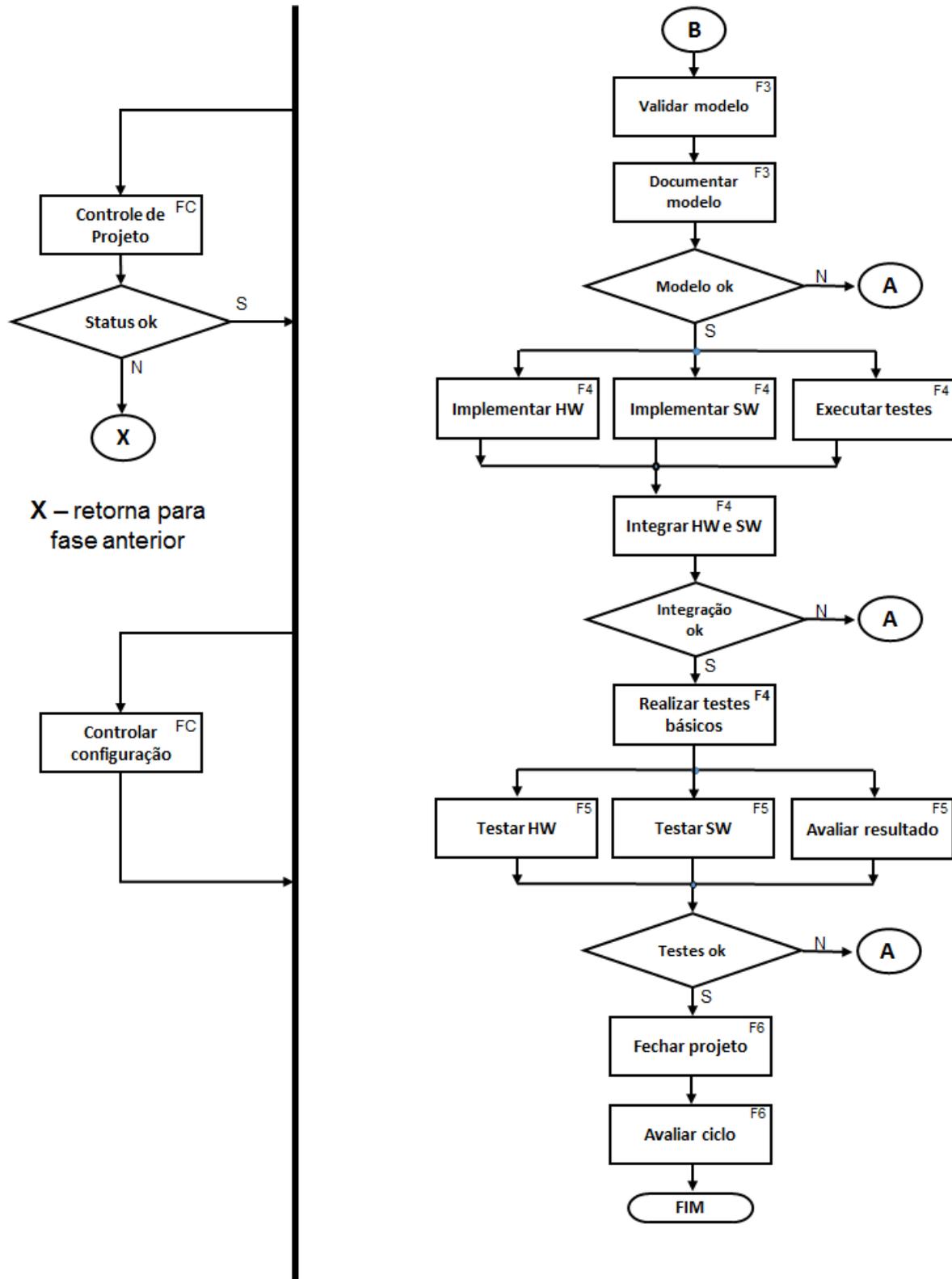


Figura 4.3 - Fluxograma geral de realização do processo - Parte-2

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3. Fluxograma de cada fase do processo

A Figura 4.4 apresenta a notação utilizada para a elaboração dos fluxos de fase, os quais seguem a numeração de 4.5, 4.12, 4.18, 4.24, 4.30 e 4.34. É preciso considerar os seguintes aspectos:

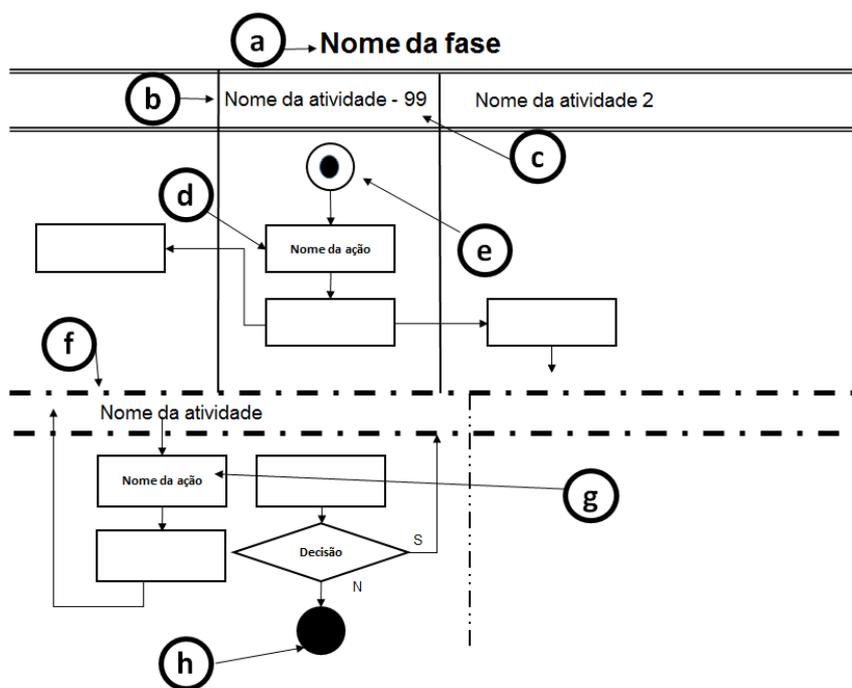


Figura 4.4 - Notação para o fluxograma das fases

- Nome da fase a que o fluxo se refere.
- Raia onde são realizadas as ações em cada atividade.
- Número sequencial da atividade no processo.
- Ação a ser realizada.
- As atividades localizadas nessas raias são consequência de todas as ações das atividades das raias acima.
- Início do fluxo.
- Ação a ser realizada em consequência de todas as outras ações das atividades das raias acima.
- Fim do fluxo.

Os Fluxogramas das atividades bem como a sua execução não necessitam de orientações especiais.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1. Fluxograma da fase de engenharia – F1

Este Fluxograma sugere de forma geral as principais atividades da fase.

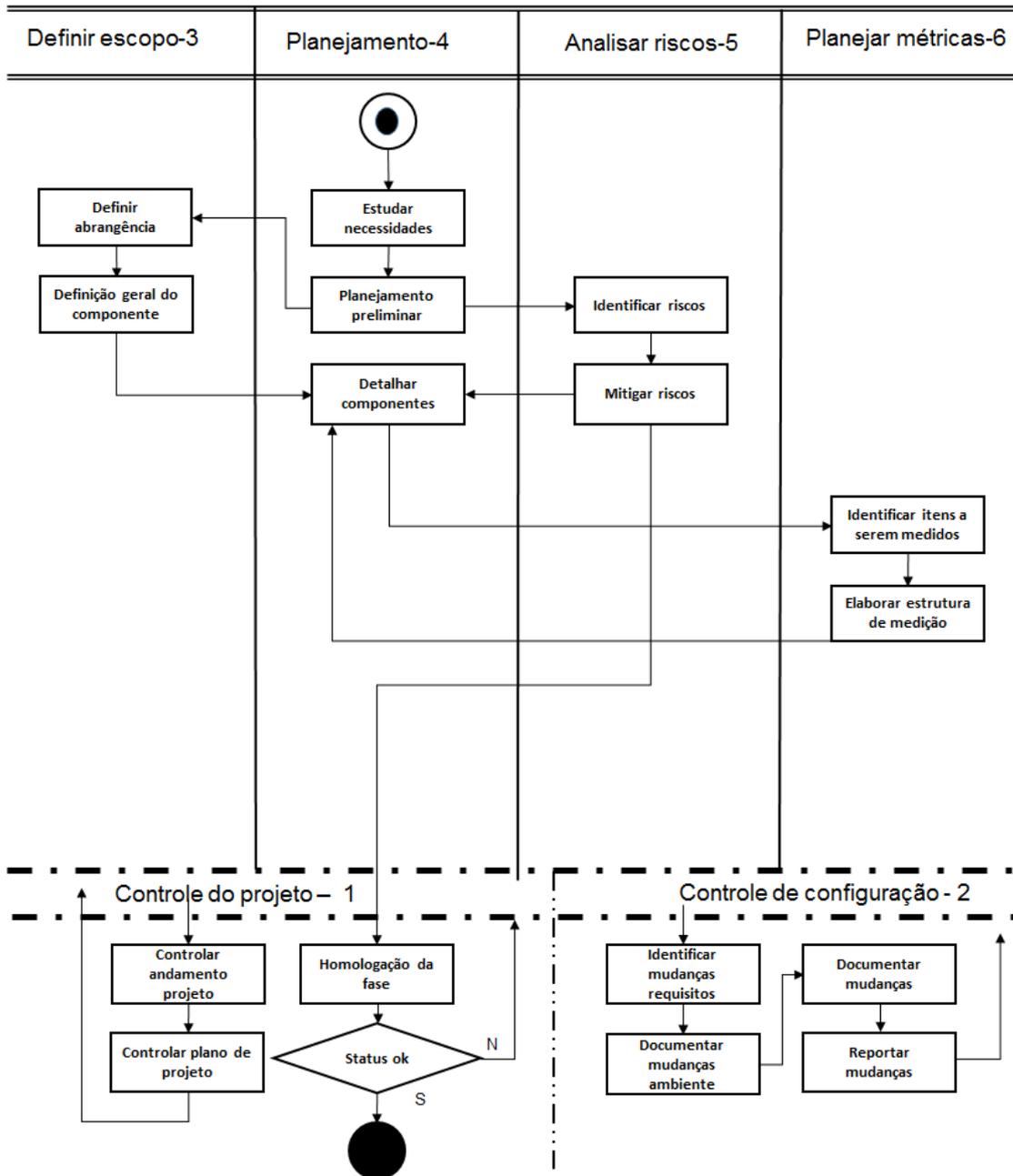


Figura 4.5 - Fluxograma da fase de engenharia

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1.1. Fluxograma da atividade de controle de projeto - 1

As atividades deste fluxograma têm o objetivo de realizar o controle do projeto em todas as fases.

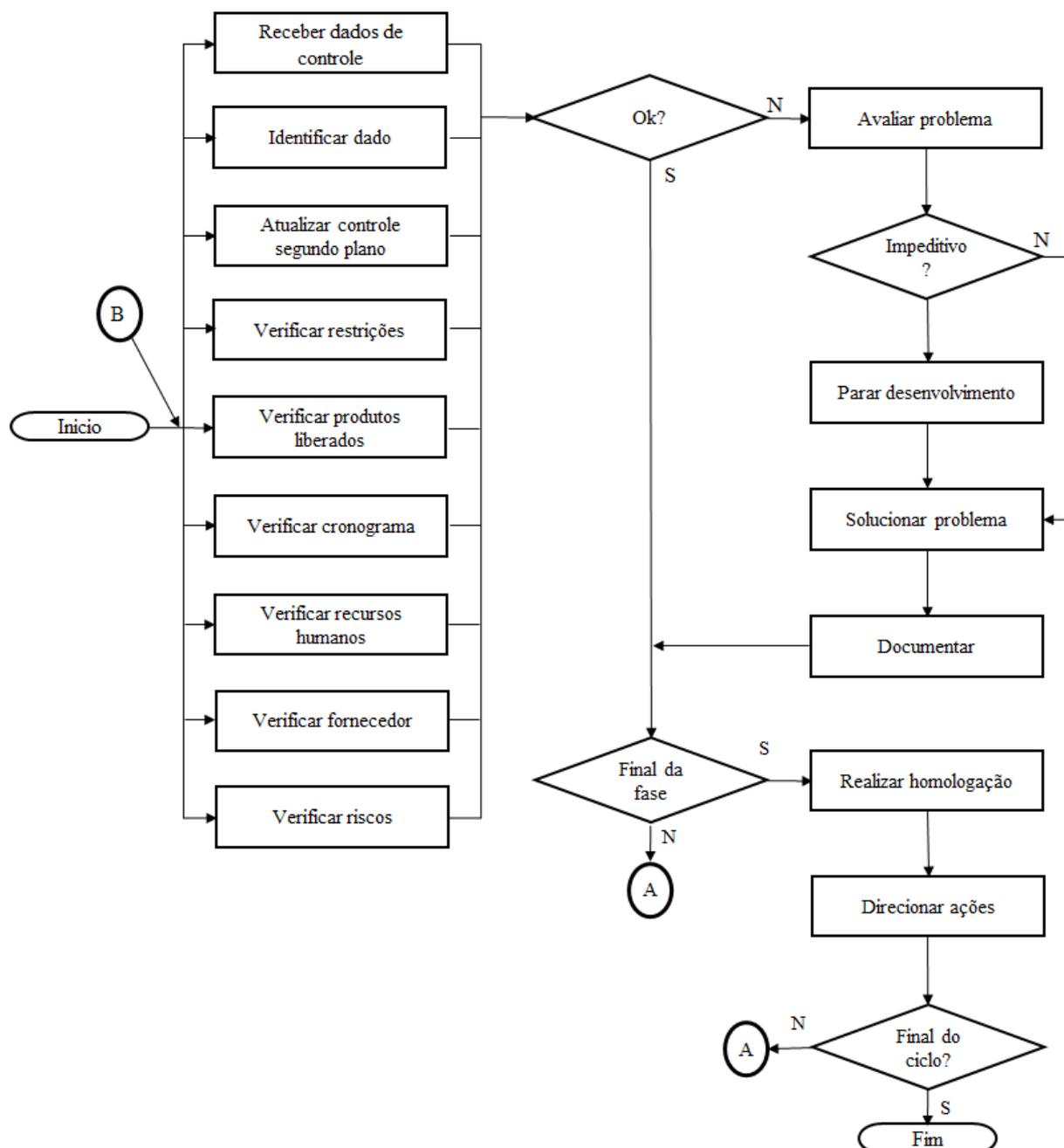


Figura 4.6 - Atividade controle de projeto

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1.2. Fluxograma da atividade controle de configuração - 2

As atividades deste fluxograma têm o objetivo de realizar o controle de configuração e mudanças de versão de todos os componentes do projeto sendo realizado em todas as fases.

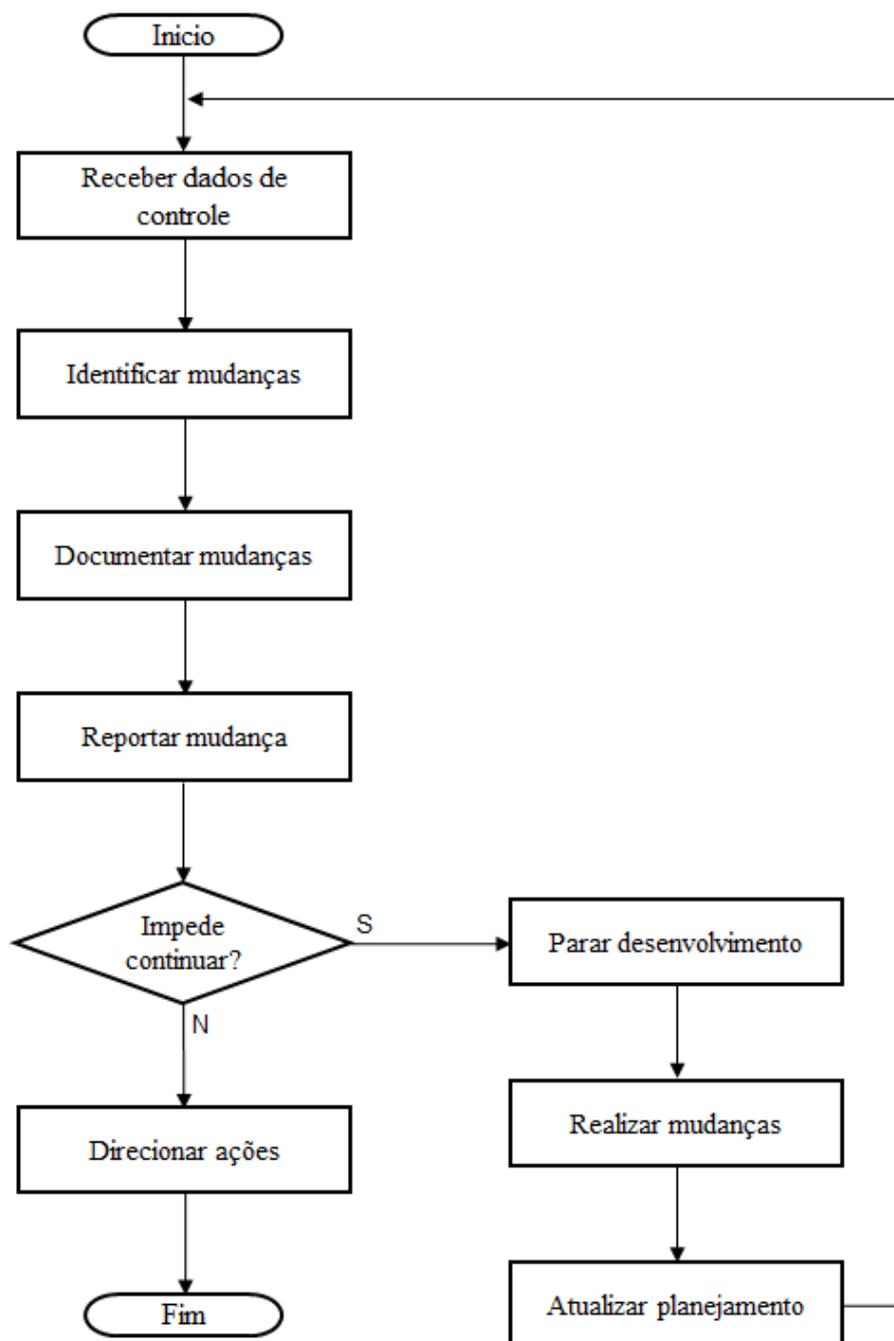


Figura 4.7 - Controle configuração

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1.3. Fluxograma da atividade definir escopo - 3

As atividades deste fluxograma têm o objetivo de realizar a definição do escopo do componente a ser produzido, e a sua inserção no produto completo.

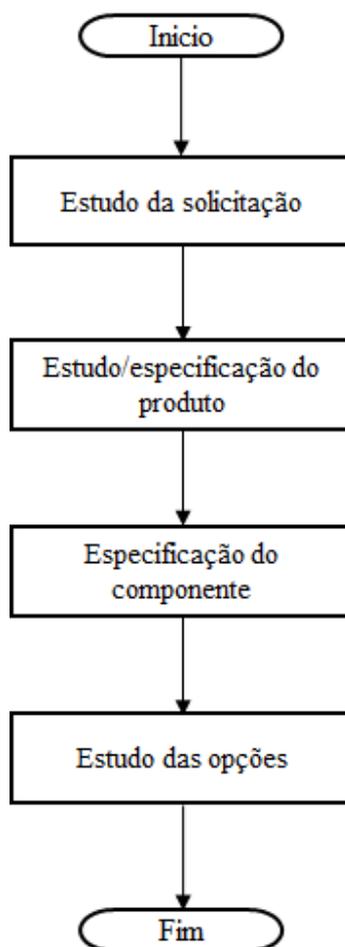


Figura 4.8 - Definir escopo-3

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1.4. Fluxograma da atividade planejamento - 4

As atividades deste fluxograma têm o objetivo de realizar o planejamento do projeto do componente a ser produzido.

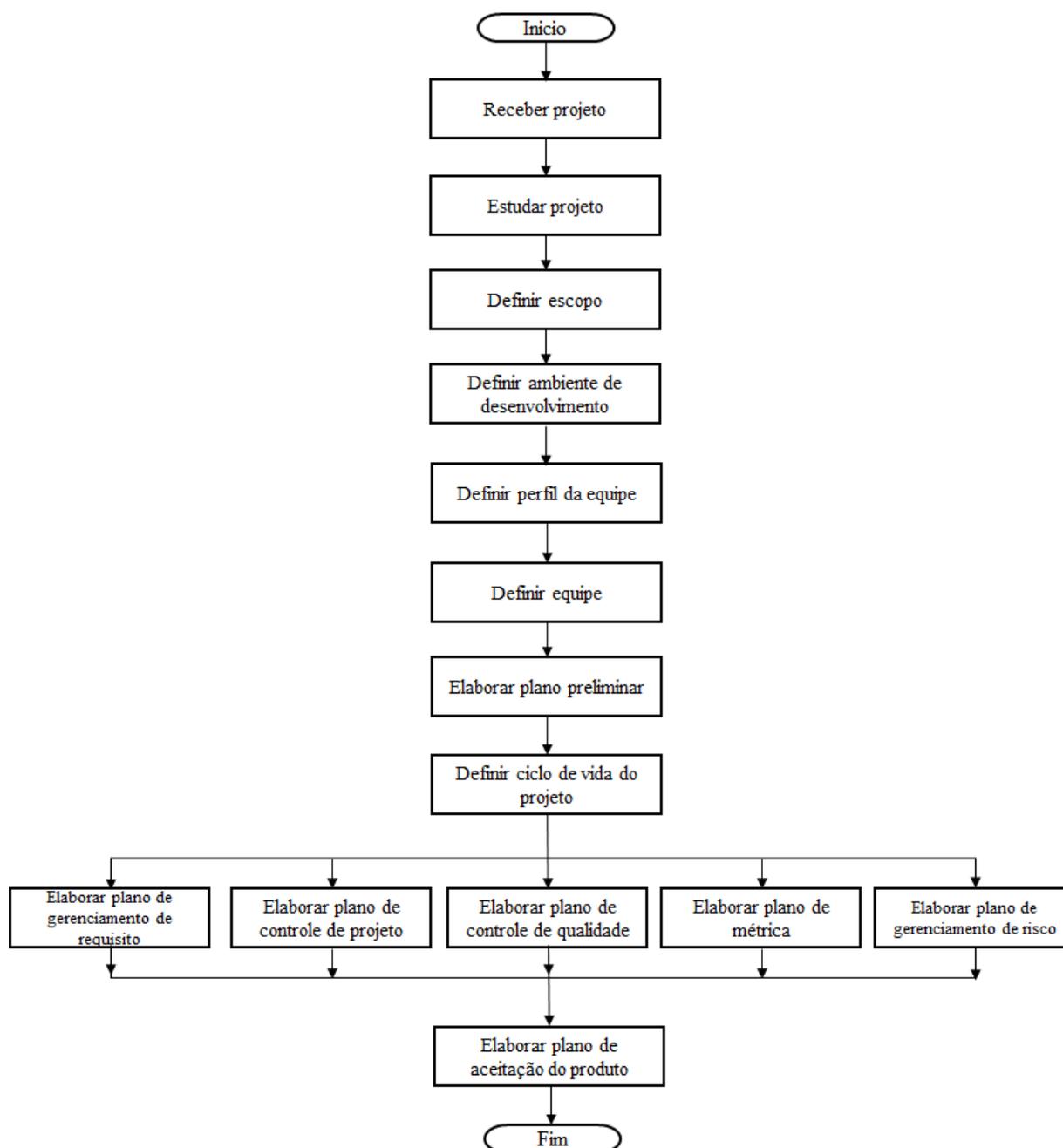


Figura 4.9 - Planejamento-4

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1.5. Fluxograma da atividade analisar riscos - 5

As atividades deste fluxograma têm o objetivo de identificar e prevenir os possíveis riscos de problemas que podem acontecer durante a realização do projeto.

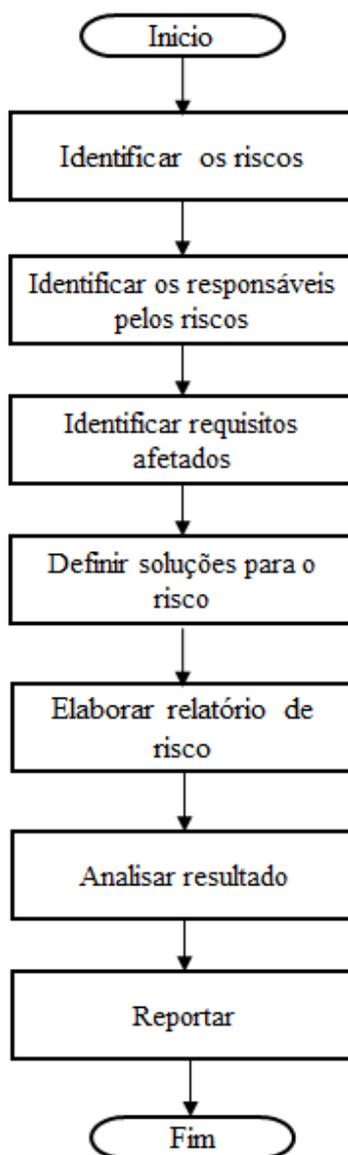


Figura 4.10 - Analisar riscos-5

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.1.6. Fluxograma da atividade planejar métricas - 6

As atividades deste fluxograma têm o objetivo de realizar a medição dos componentes a serem medidos e acompanhando durante a realização do projeto. Os itens de controle são planejados e podem ser itens do componente a ser produzido e também itens (artefatos) do processo.

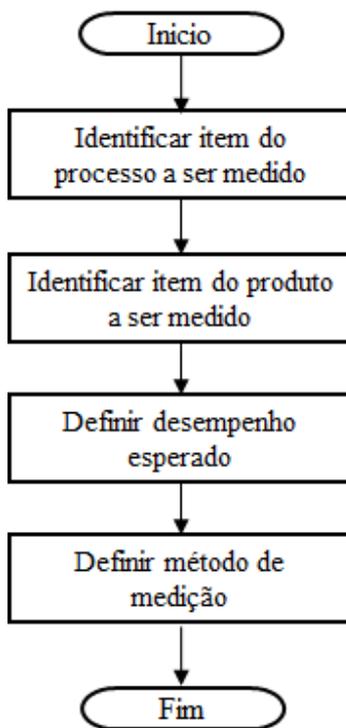


Figura 4.11 - Planejar métricas-6

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2. Fluxograma da fase de análise de requisito - F2

Este Fluxograma sugere de forma geral as principais atividades da fase.



Figura 4.12 - Análise de requisitos-F2

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.1. Fluxograma da atividade especificação de requisito - 7

As atividades deste fluxograma têm o objetivo de realizar a especificação do componente a ser produzido, e devem contemplar desde o entendimento do produto completo com seus requisitos e identificar os requisitos do componente e a sua influência no produto.

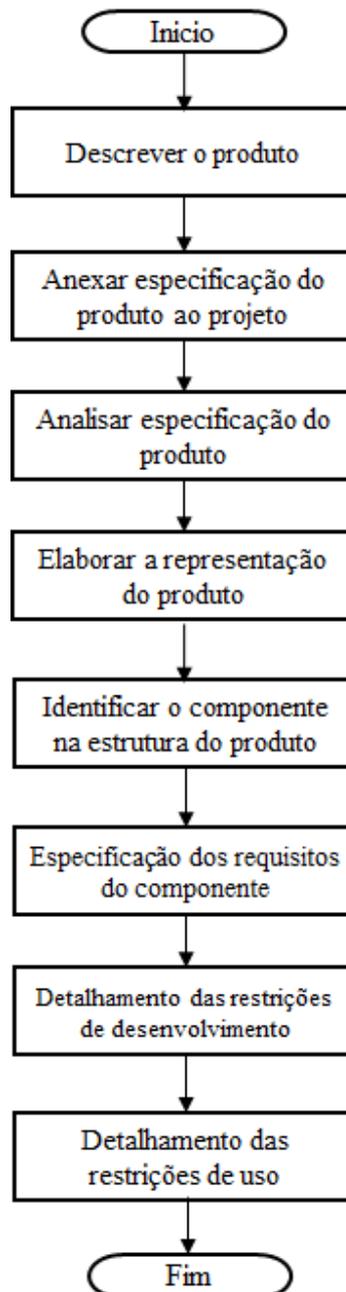


Figura 4.13 - Especificação de requisito-7

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.2. Fluxograma da atividade definição de componente - 8

As atividades deste fluxograma têm o objetivo de especificar detalhadamente o componente a ser produzido, incorporando suas restrições e atendendo seus requisitos especiais.

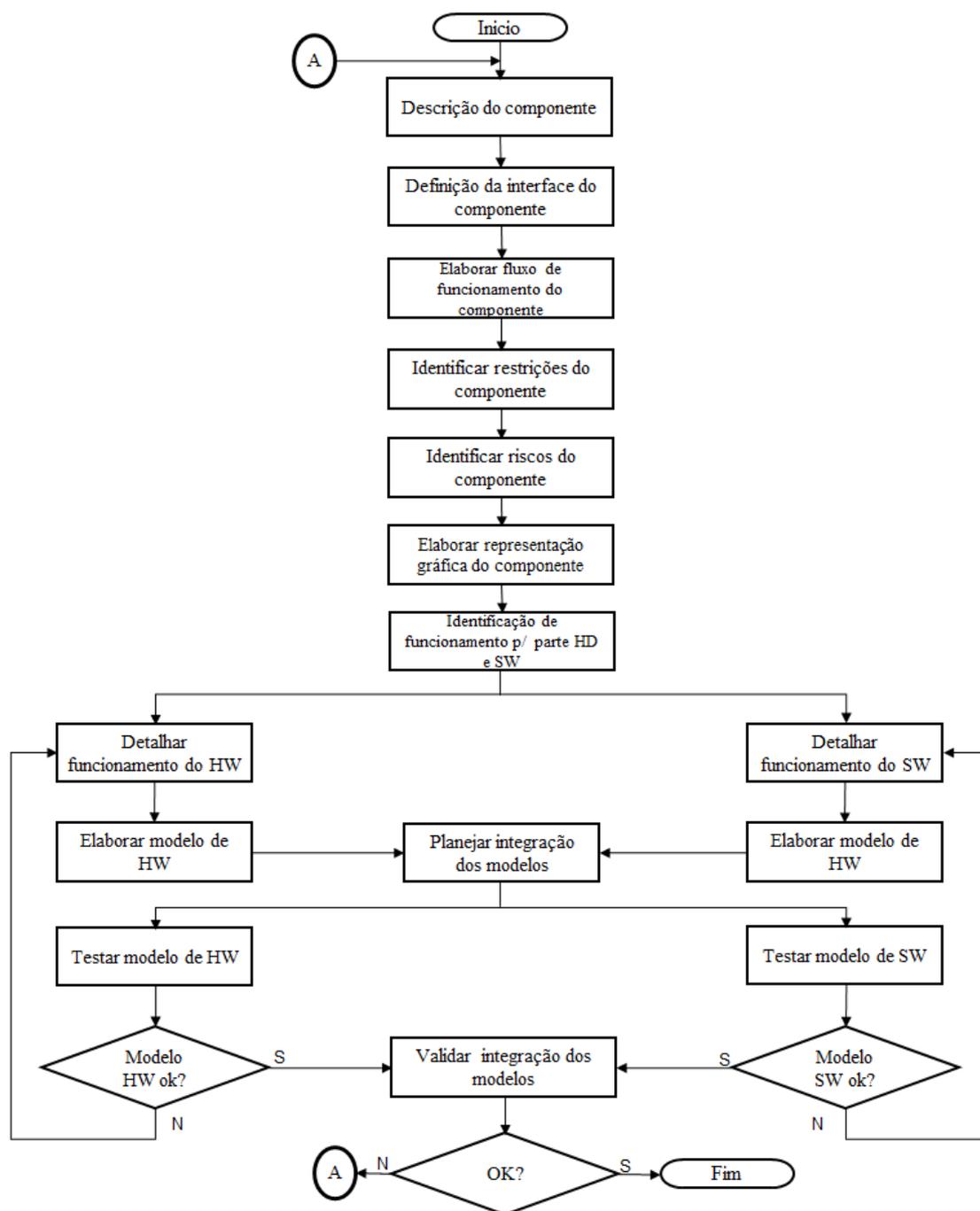


Figura 4.14 - Definição de componente-8

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.3. Fluxograma da atividade planejamento do sistema - 9

As atividades deste fluxograma têm o objetivo de realizar o planejamento do projeto de produção do componente e em todas as fases.

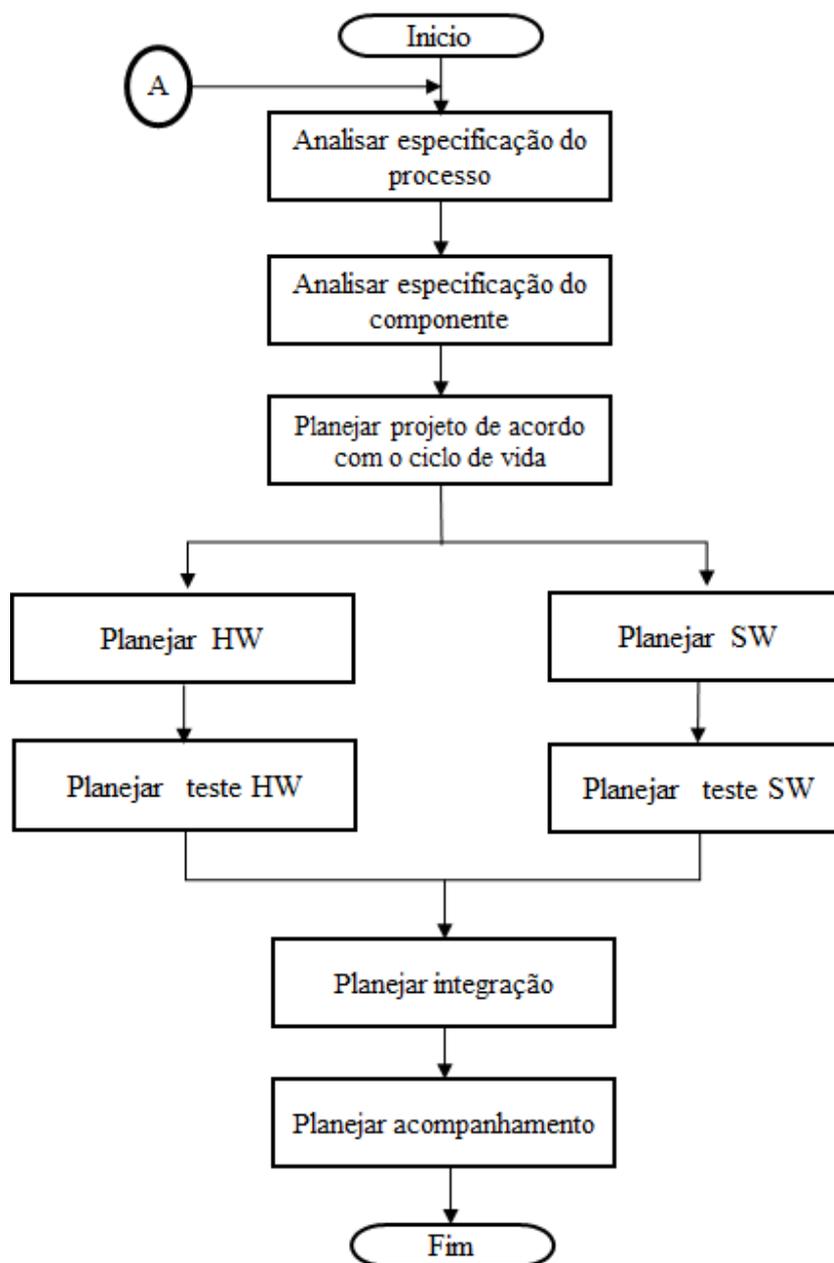


Figura 4.15 - Planejamento do sistema-9

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.4. Fluxograma da atividade projeto de componente - 10

As atividades deste fluxograma têm o objetivo elaborar o projeto do componente detalhamento a produção da parte SW e da parte HW.

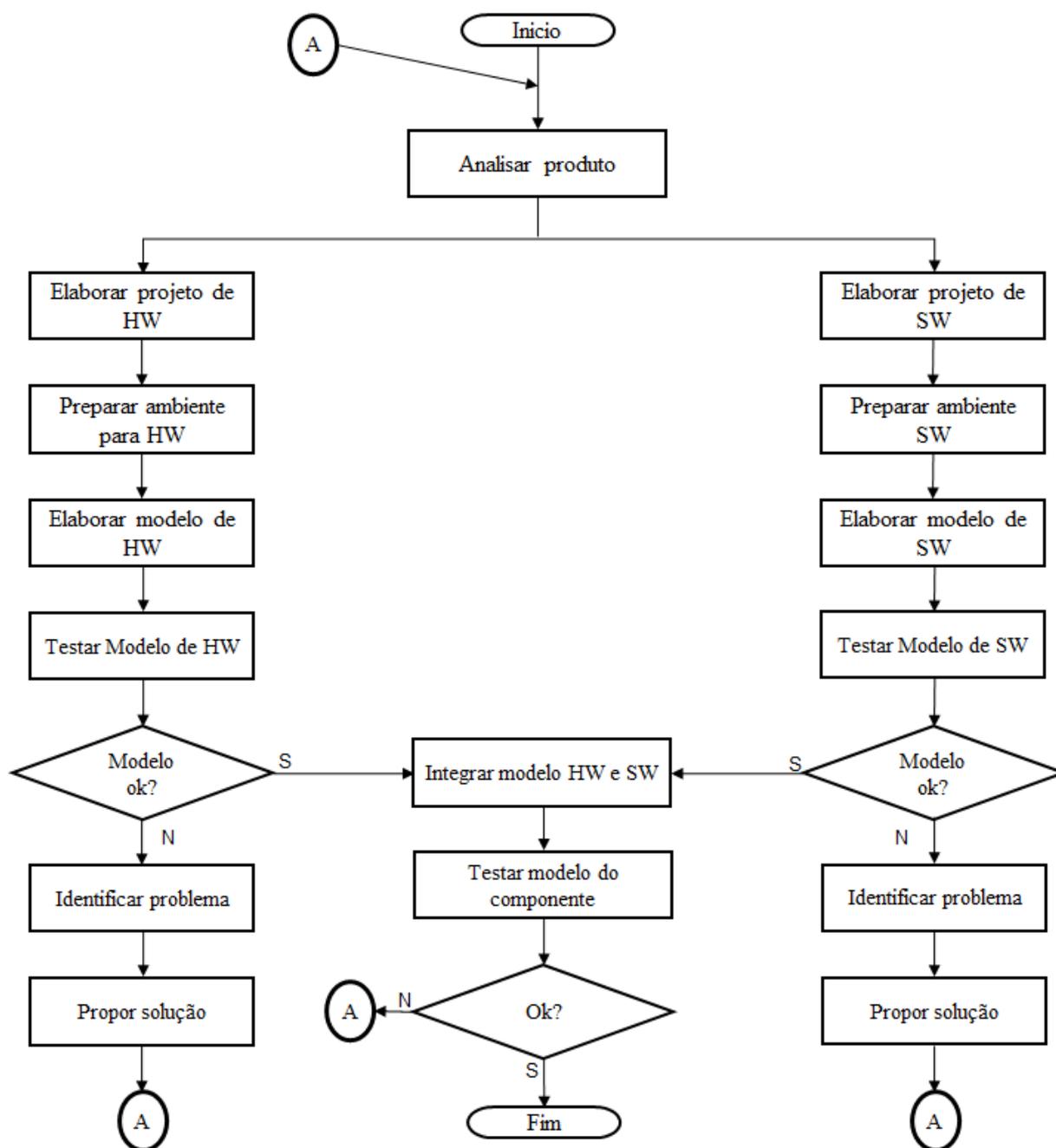


Figura 4.16 - Projeto de componente-10

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.5. Fluxograma da atividade definir itens de controle - 11

As atividades deste fluxograma têm o objetivo de identificar os itens que devem ser controlados durante a realização do projeto, e devem ser considerados os itens do componente e os artefatos produzidos como resultado do processo.

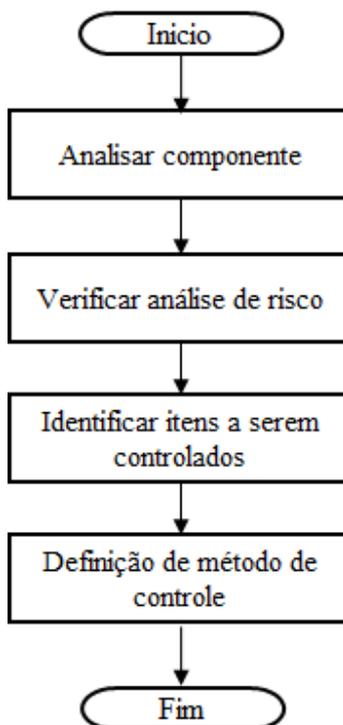


Figura 4.17 - Definir itens de controle-11

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.6. Fluxograma da fase da análise e projeto - F3

Este Fluxograma sugere de forma geral as principais atividades da fase.

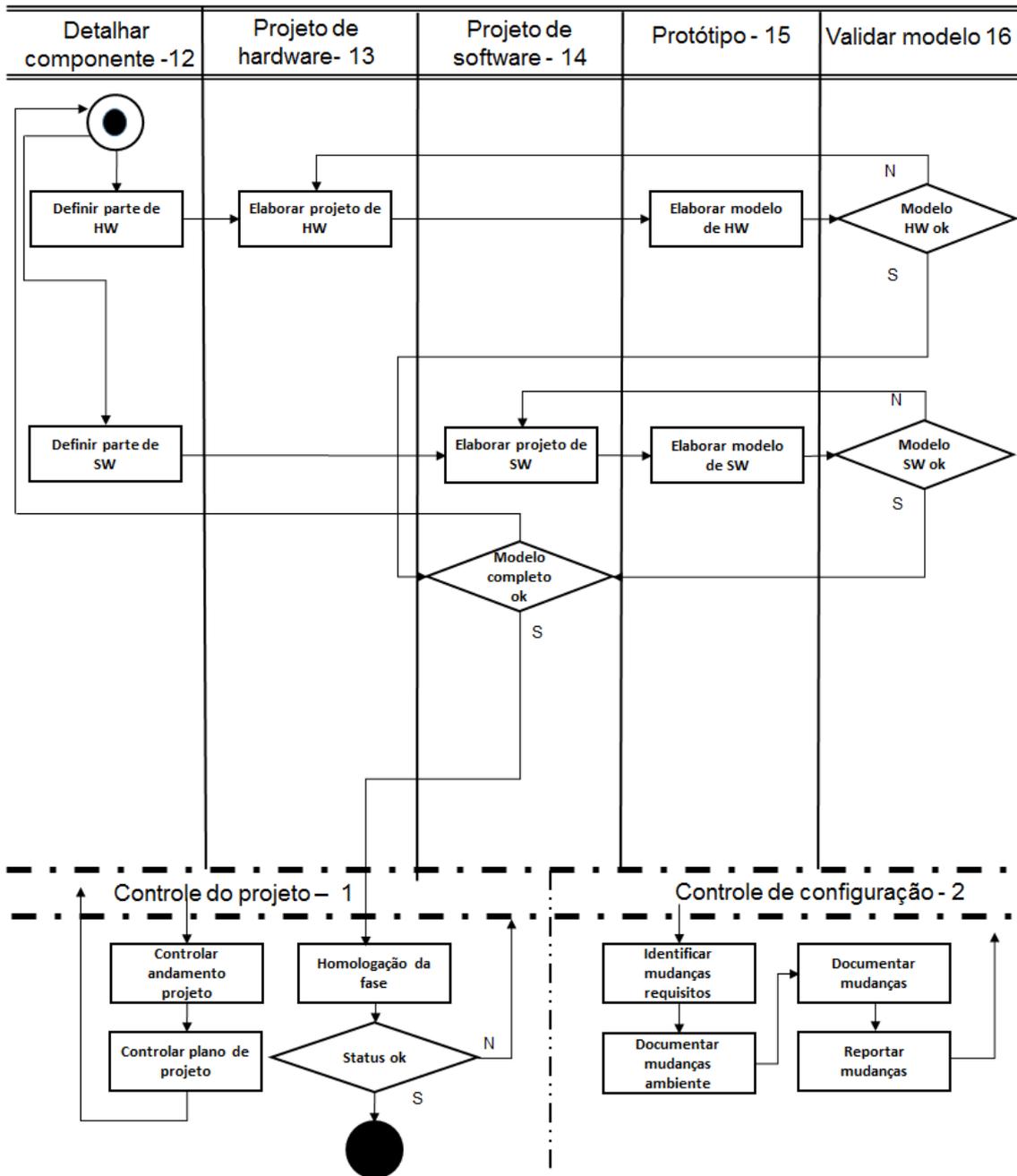


Figura 4.18 - Fluxograma da fase de análise e projeto

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.7. Fluxograma da atividade detalhar componente - 12

As atividades deste fluxograma têm o objetivo de detalhar a produção do componente de acordo com o modelo.

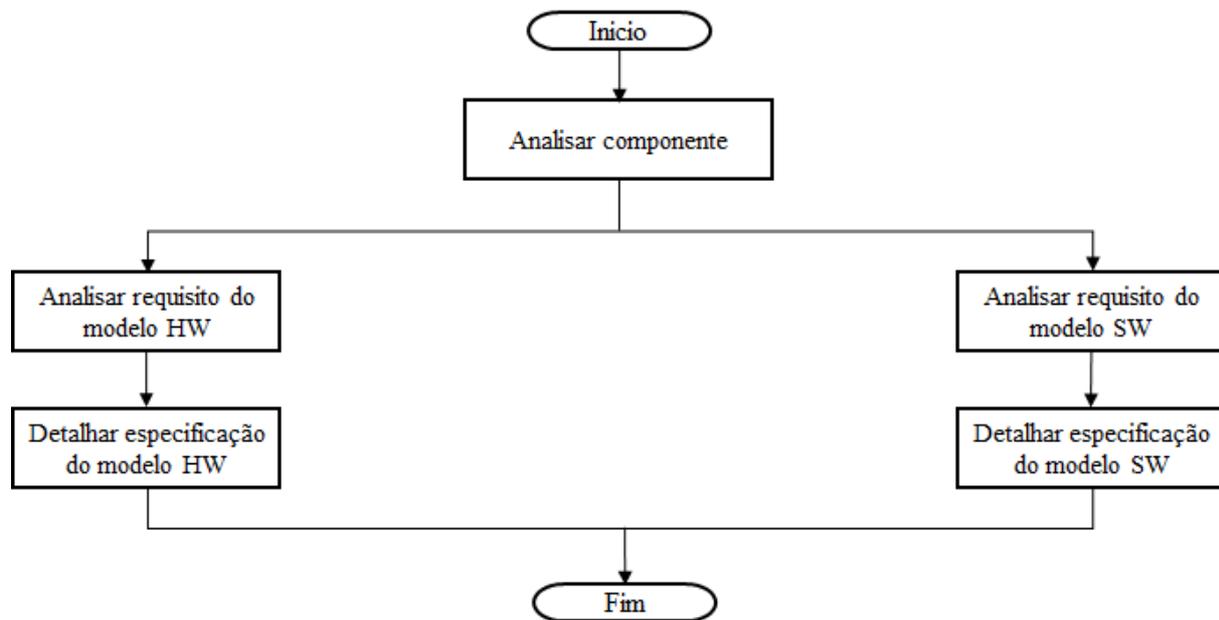


Figura 4.19 - Detalhar componente-12

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.8. Fluxograma da atividade projeto de hardware - 13

As atividades deste fluxograma têm o objetivo projetar modelos de *hardware* para serem validados em nível de projeto.

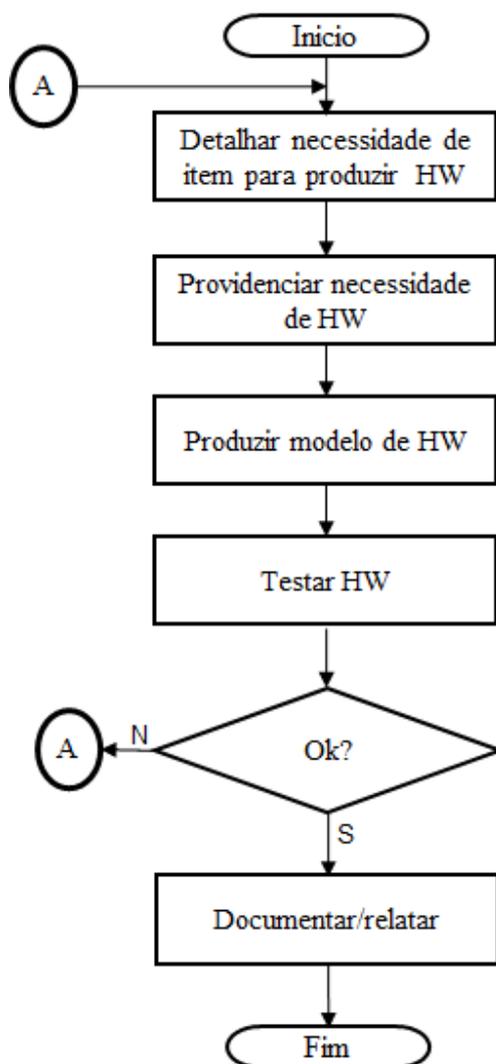


Figura 4.20 - Projeto de *hardware*-13

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.9. Fluxograma da atividade projeto de software - 14

As atividades deste fluxograma têm o objetivo projetar modelos de *software* para serem validados em nível de projeto.

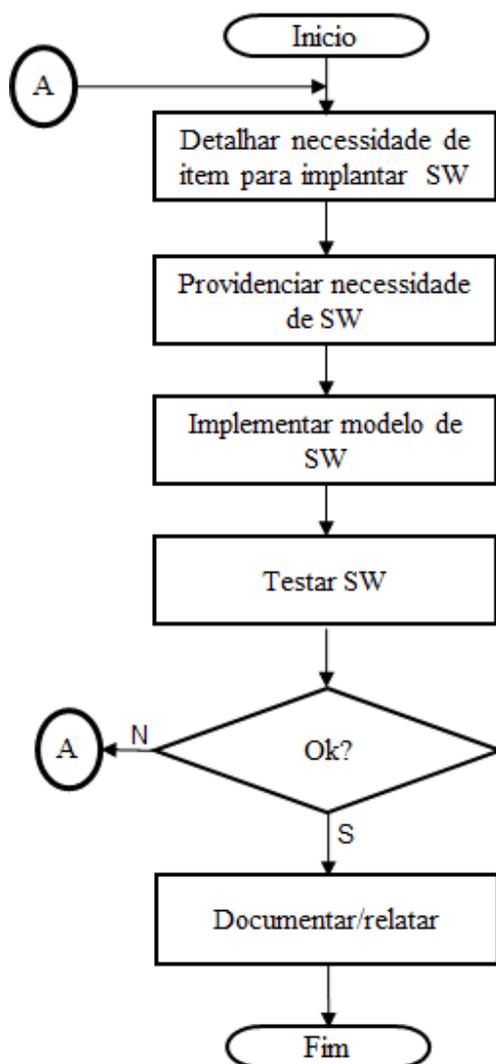


Figura 4.21 - Projeto de *software*-14

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.10. Fluxograma da atividade protótipo - 15

As atividades deste fluxograma têm o objetivo produzir protótipos do *hardware* e do *software* e integra-los em nível de projeto.

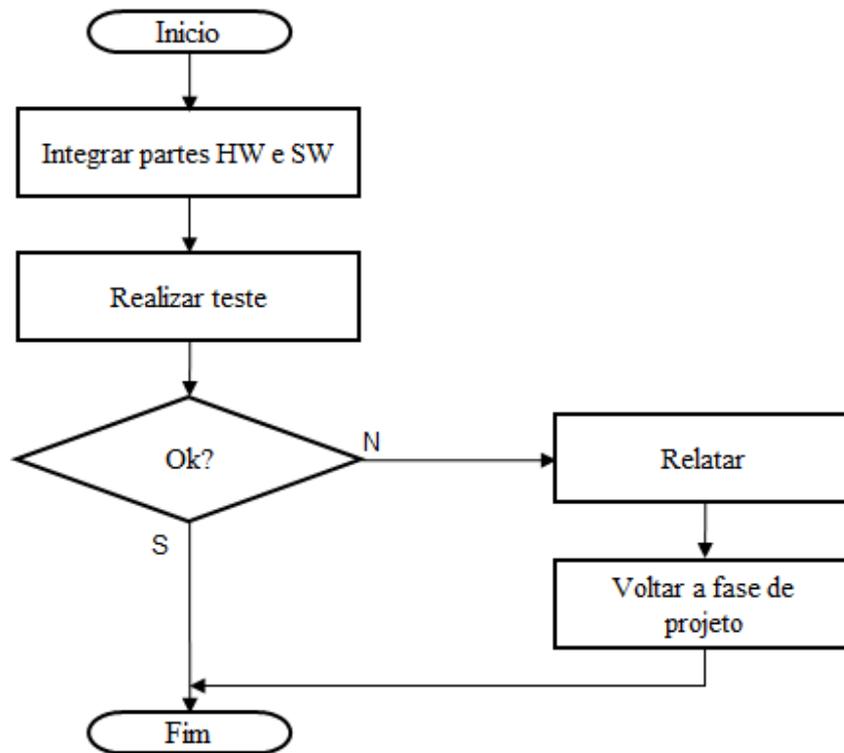


Figura 4.22 - Protótipo-15

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.11. Fluxograma da atividade validar modelo - 16

As atividades deste fluxograma têm o objetivo validar os protótipos em nível de projeto.

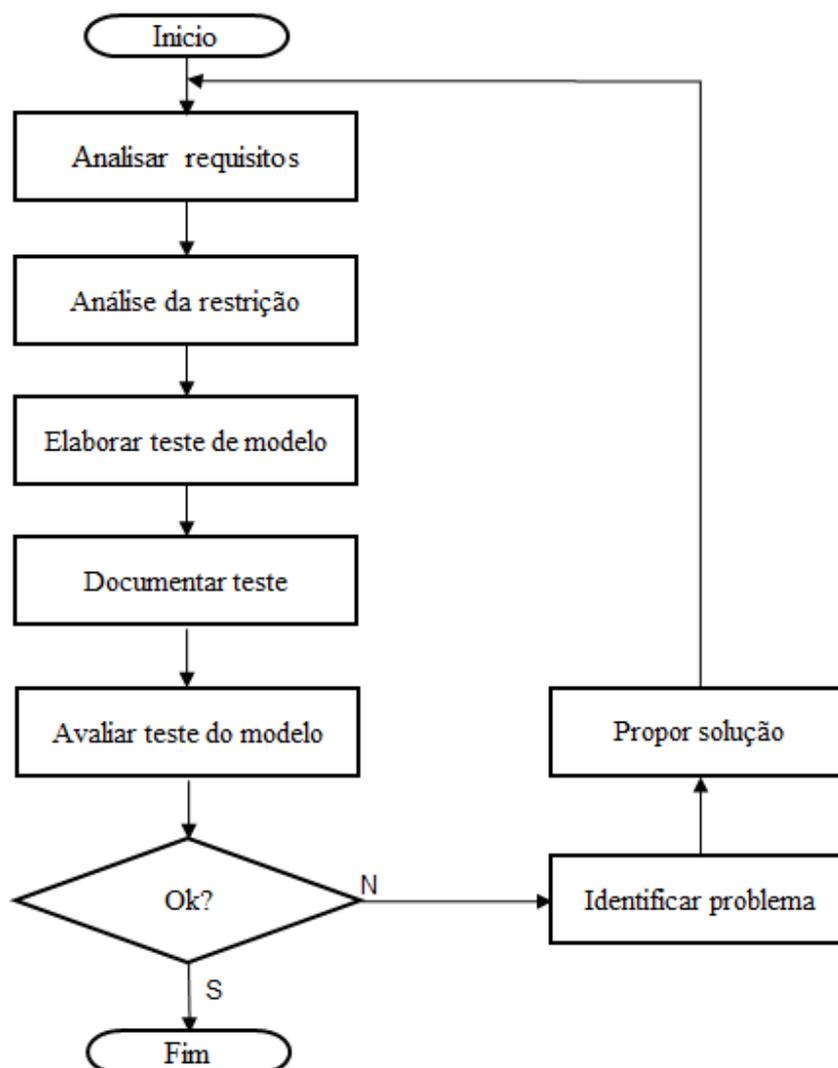


Figura 4.23 - Validar modelo-16

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.12. Fluxograma da fase de implementação e integração - F4

Este Fluxograma sugere de forma geral as principais atividades da fase.

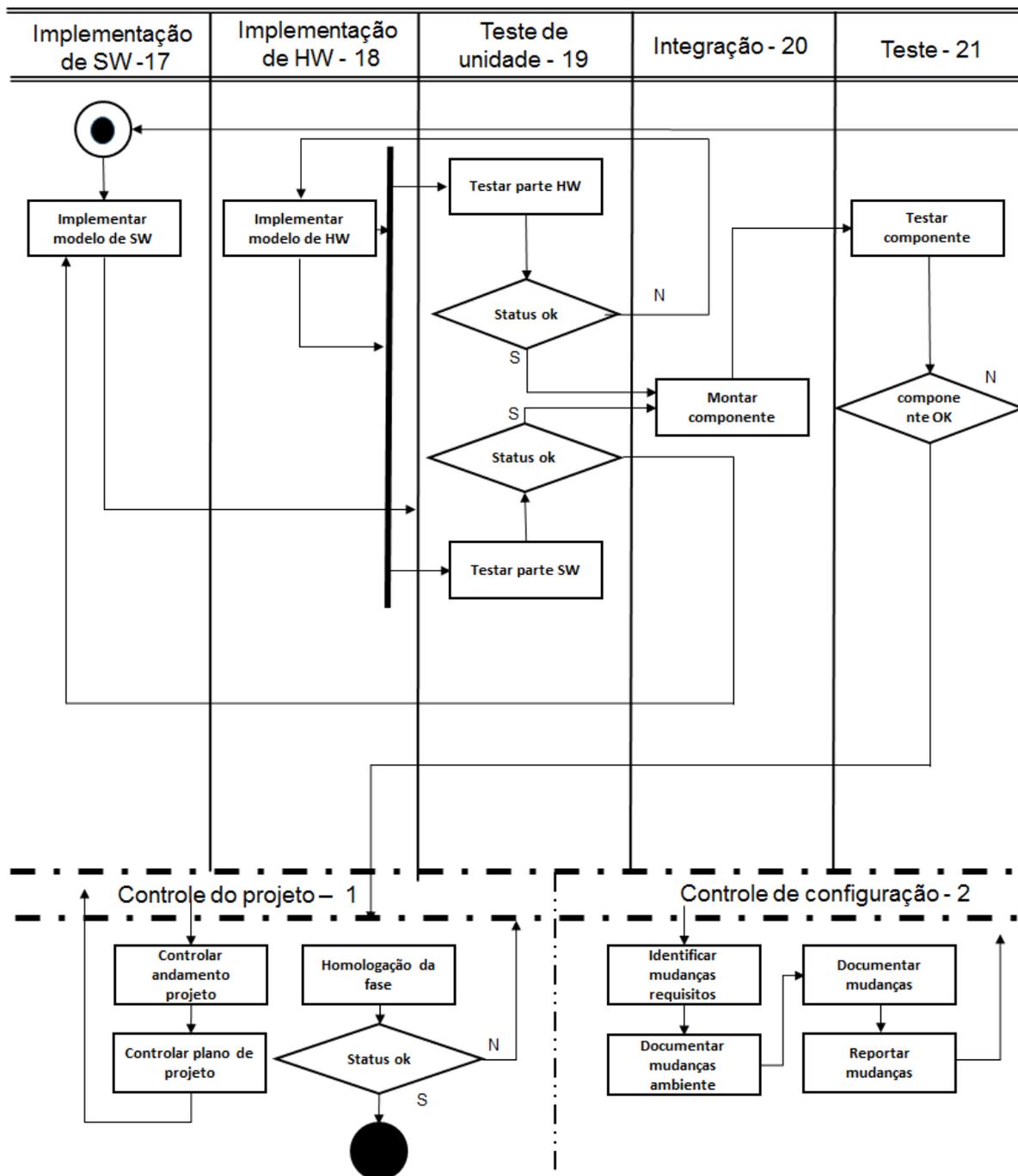


Figura 4.24 - Implementação e integração-F4

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.13. Fluxograma da atividade implementação de SW - 17

As atividades deste fluxograma têm o objetivo de implementar o *software* de acordo com o modelo e protótipo validado.

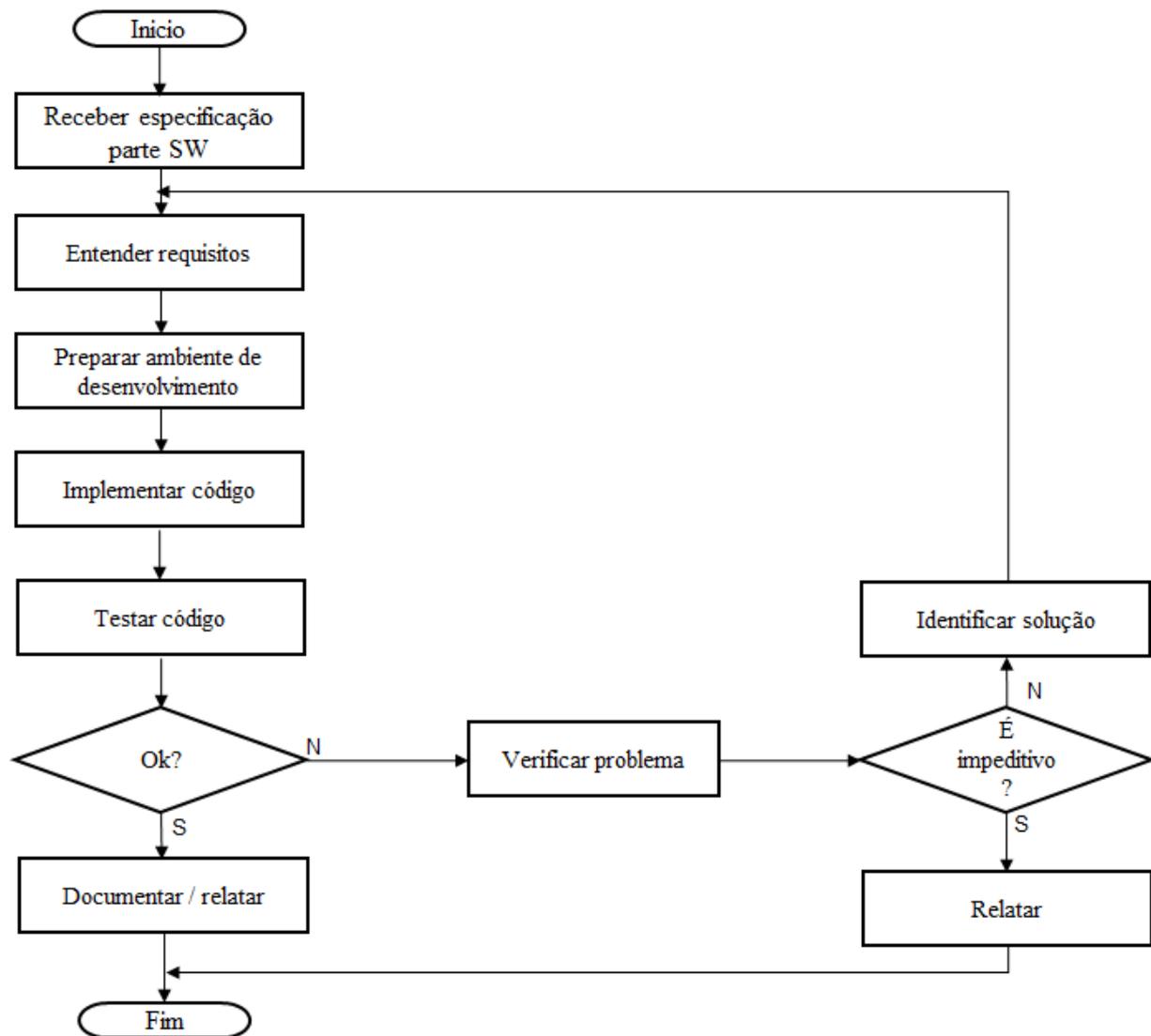


Figura 4.25 - Implementação de SW-17

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.14. Fluxograma da atividade implementação de HW - 18

As atividades deste fluxograma têm o objetivo de produzir o *hardware* de acordo com o modelo e protótipo validado.

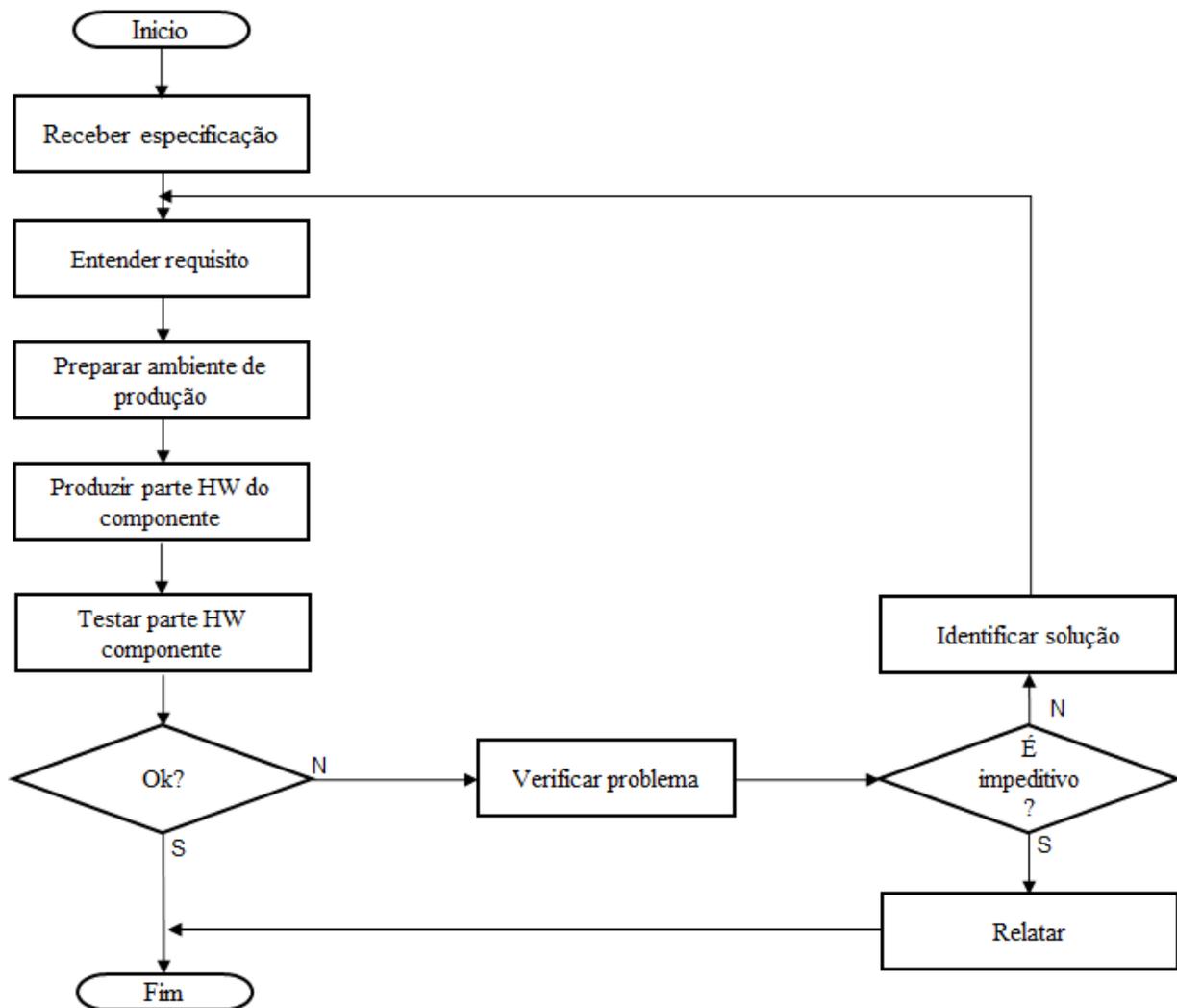


Figura 4.26 - Implementação de HW-18

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.15. Fluxograma da atividade teste de unidade - 19

As atividades deste fluxograma têm o objetivo de implementar o *software* de acordo com o modelo e protótipo validado

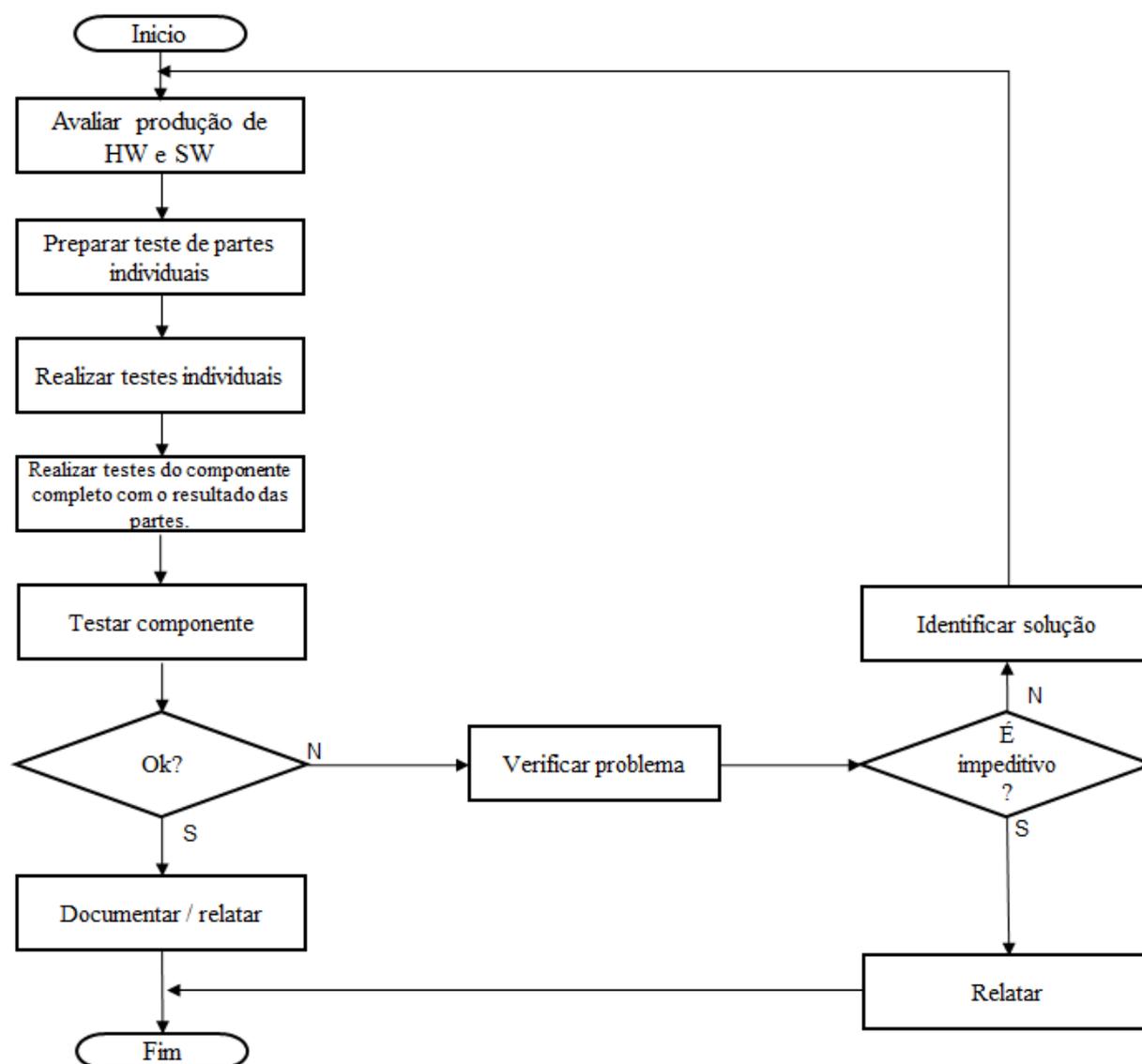


Figura 4.27 - Teste de unidade-19

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.16. Fluxograma da atividade integração - 20

As atividades deste fluxograma têm o objetivo conduzir a integração das partes *software* e *hardware* do componente. Os testes realizados nesta atividade são básicos a nível de integração.

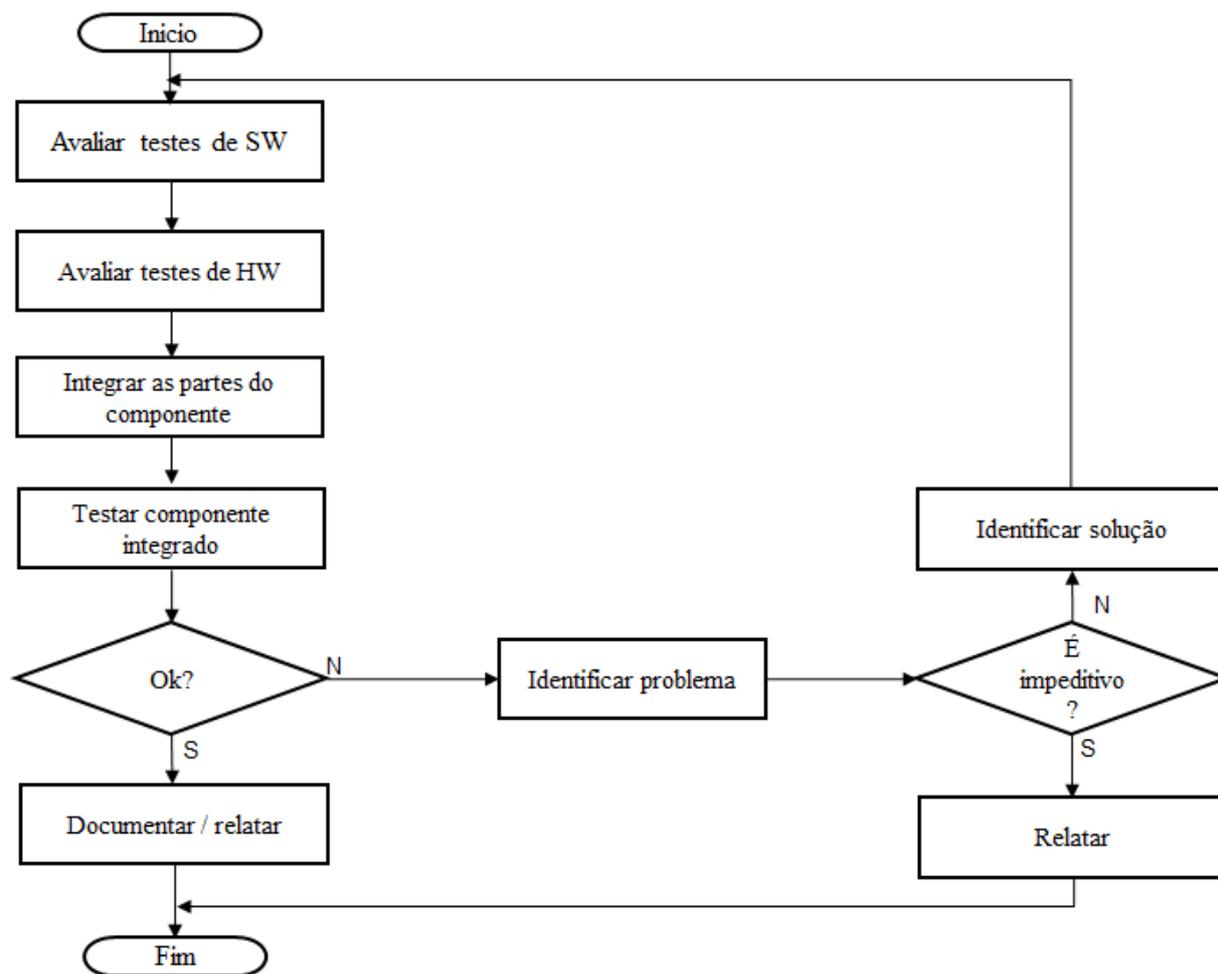


Figura 4.28 - Integração-20

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.2.17. Fluxograma da atividade teste - 21

As atividades deste fluxograma têm o objetivo de realizar testes das partes *software*, *hardware* e do componente completo a nível de comunicação entre as partes.

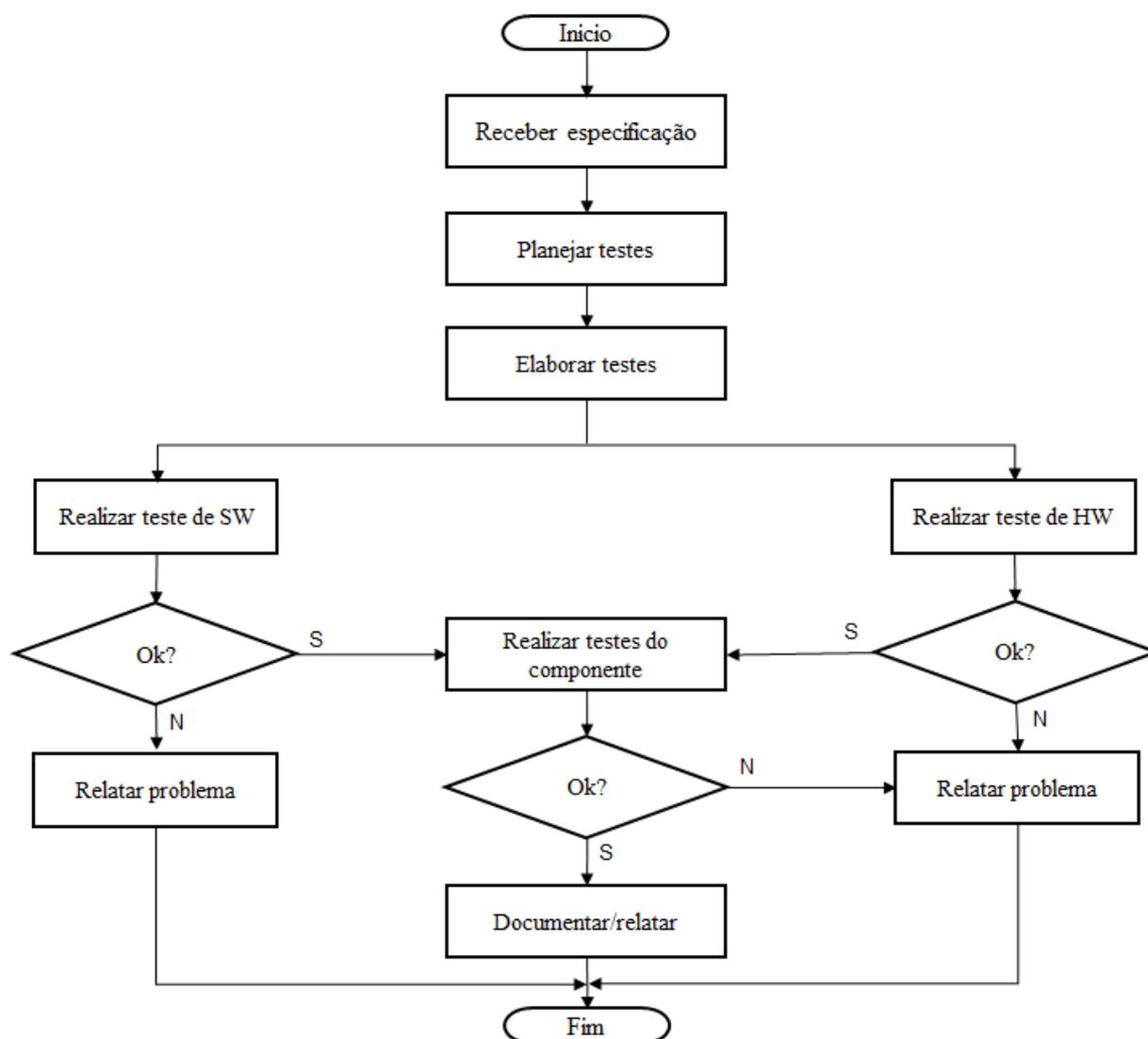


Figura 4.29 - Teste-21

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.3. Fluxograma da fase de verificação e validação

Este Fluxograma sugere de forma geral as principais atividades da fase.

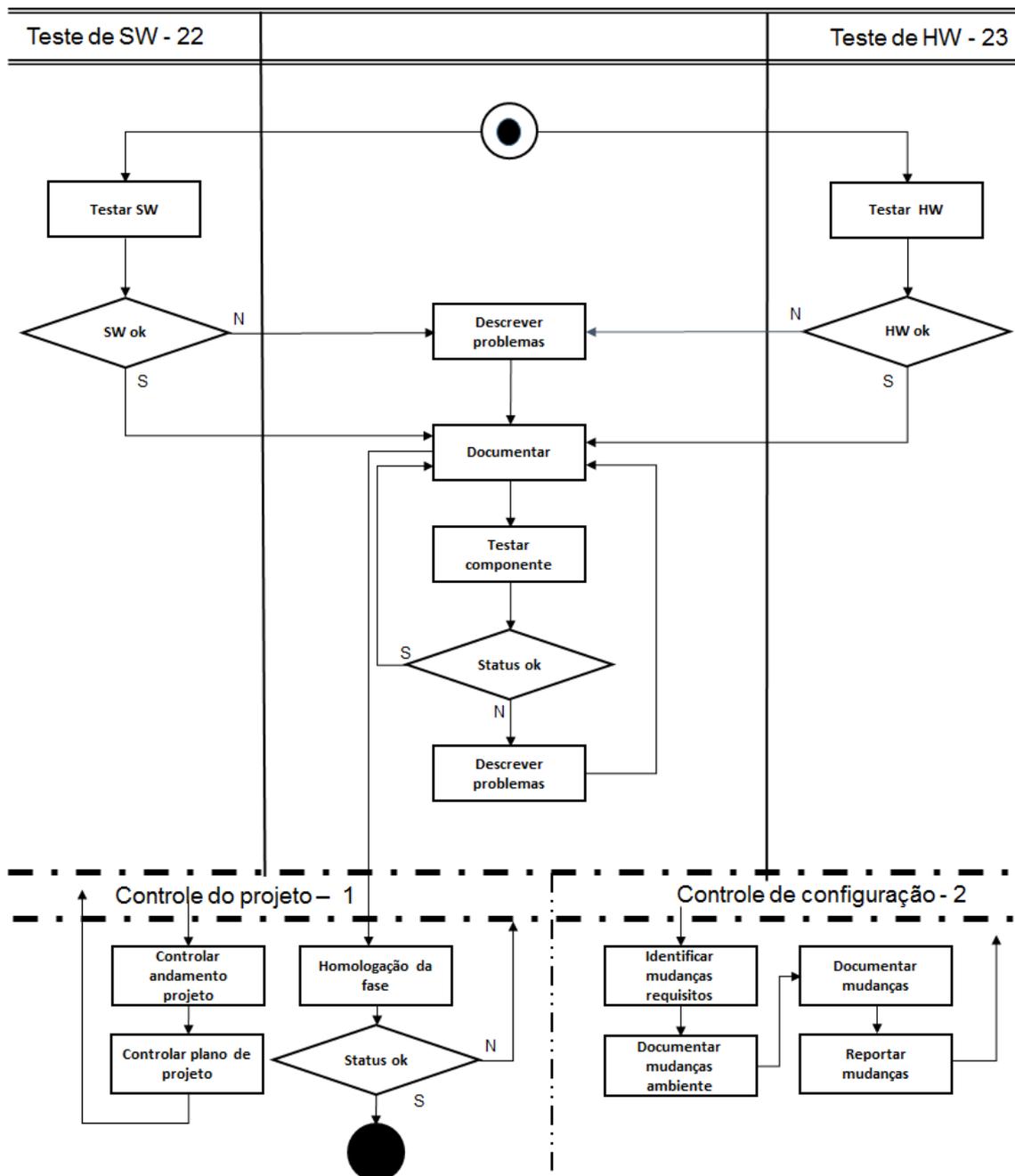


Figura 4.30 - Fluxograma da fase de verificação e validação

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.3.1. Fluxograma da atividade teste de SW - 22

As atividades deste fluxograma têm o objetivo de realizar o teste da parte *software* do componente a nível de componente e sua inserção no produto.

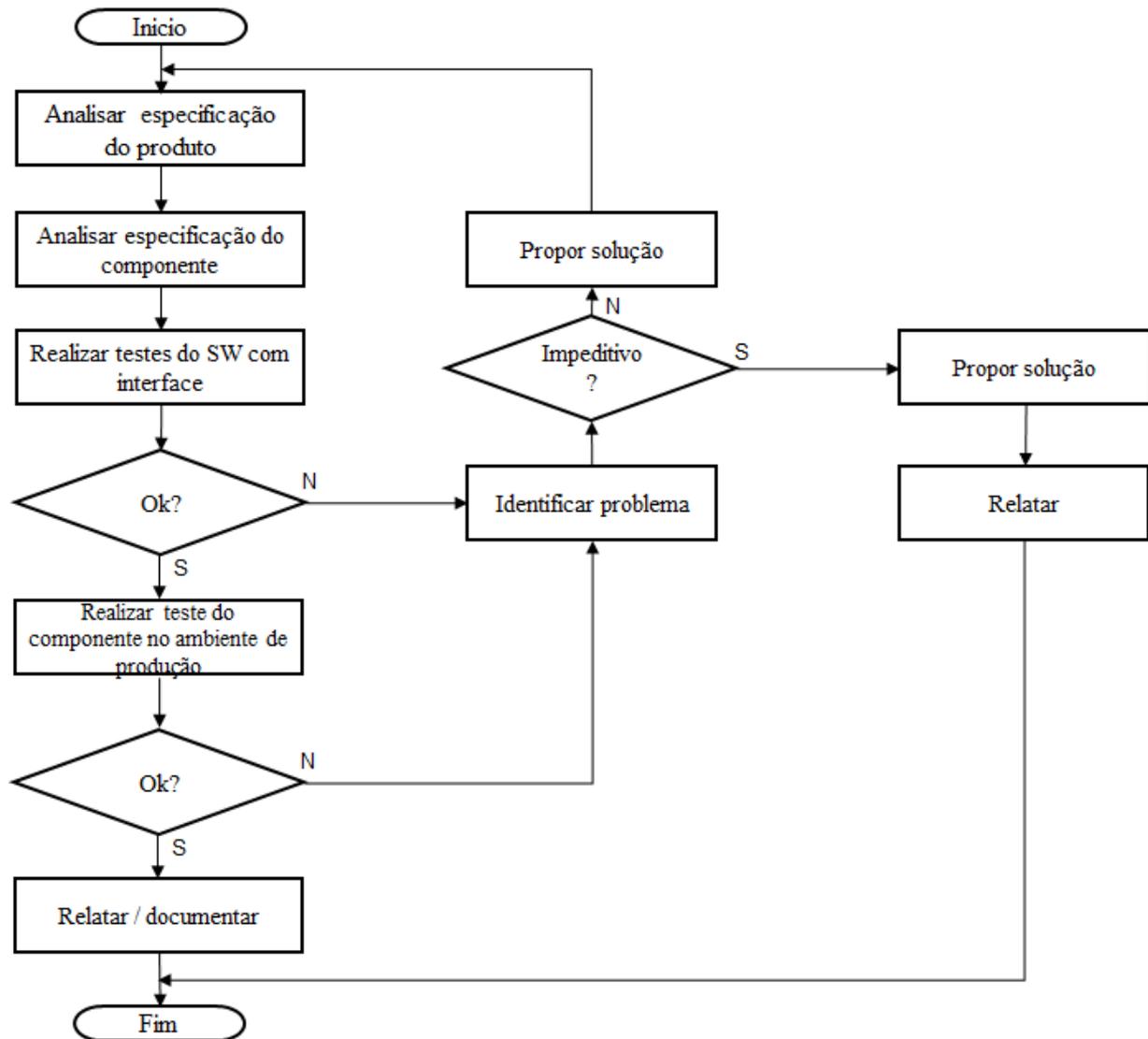


Figura 4.31 - Teste de SW-22

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.3.2. Fluxograma da atividade teste de HW - 23

As atividades deste fluxograma têm o objetivo de realizar o teste da parte *hardware* do componente a nível de componente e sua inserção no produto.

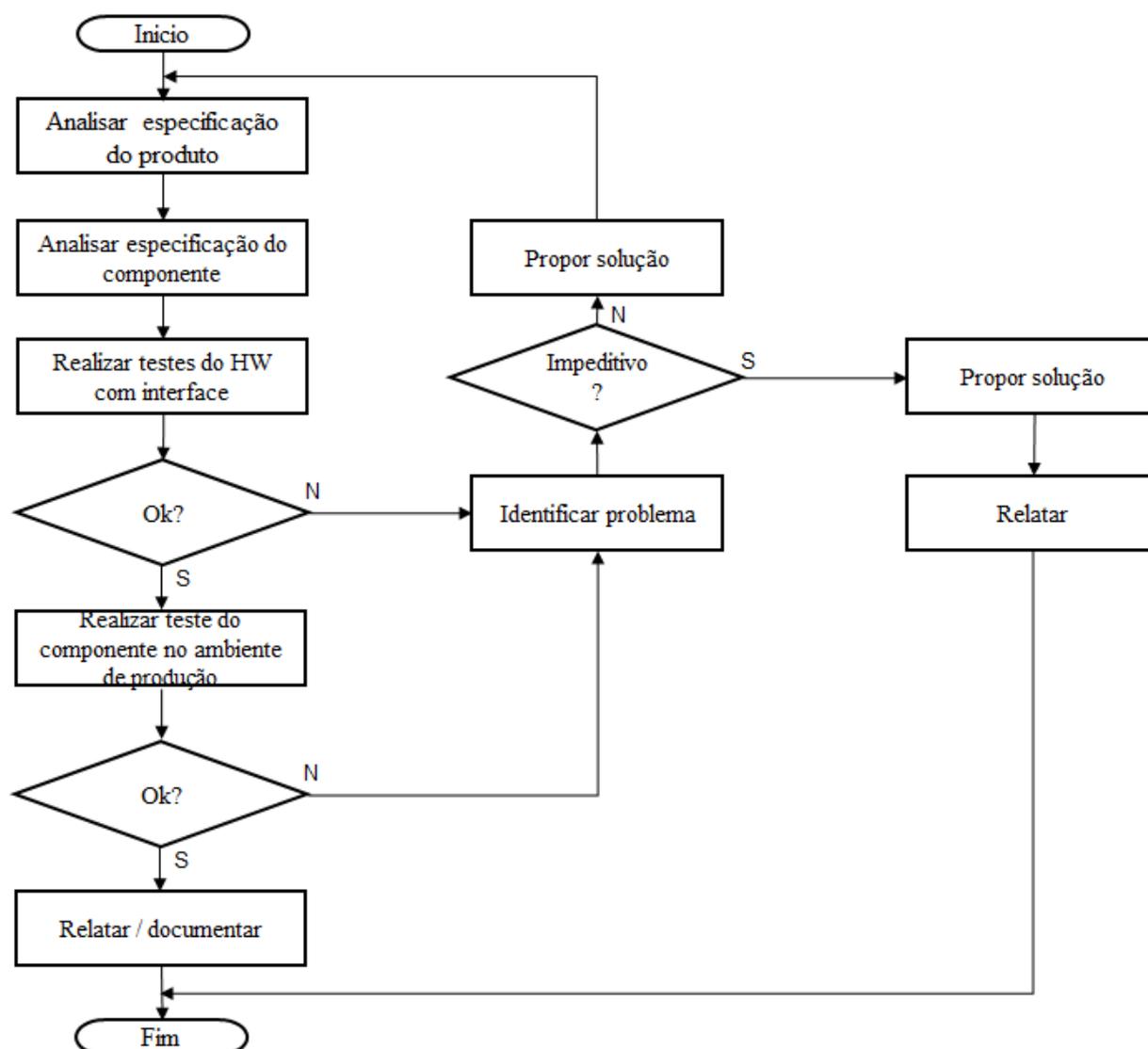


Figura 4.32 -Teste de HW-23

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.4. Fluxograma da fase de avaliação do ciclo

Este Fluxograma sugere de forma geral as principais atividades da fase.

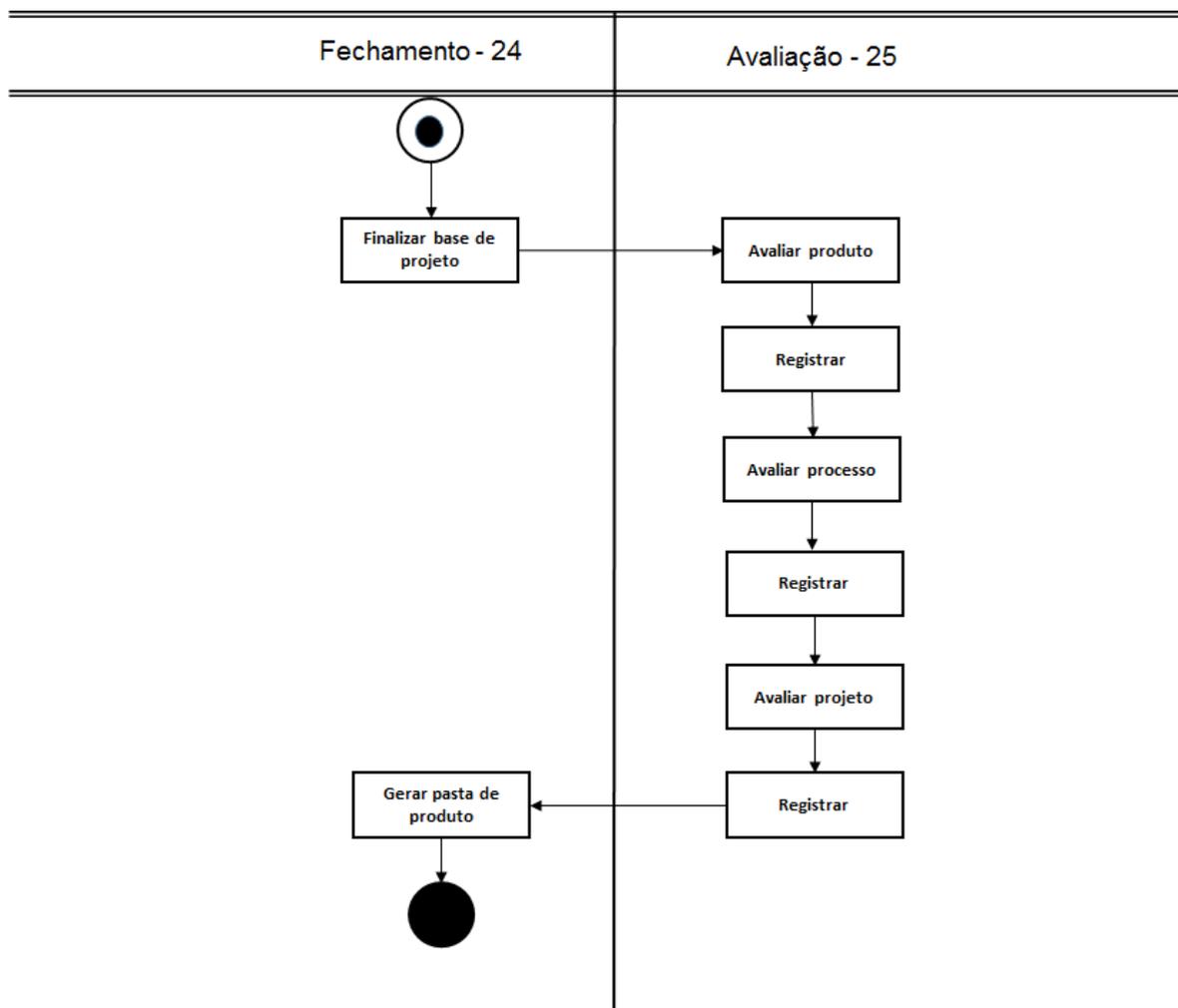


Figura 4.33 - Avaliação do ciclo

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.4.1. Fluxograma da atividade fechamento - 24

As atividades deste fluxograma têm o objetivo de encerrar o projeto e elaborar pacote do produto.

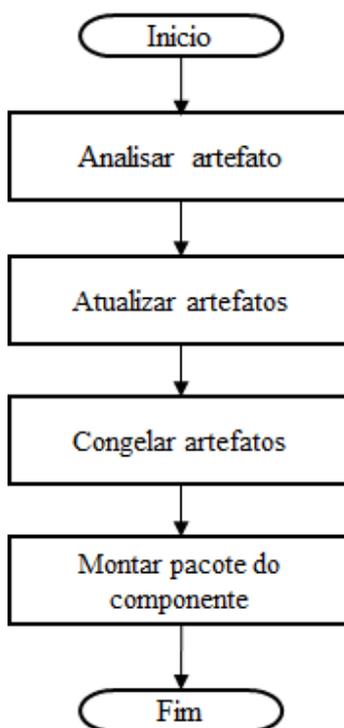


Figura 4.34 - Fechamento-24

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

4.3.4.2. Fluxograma da atividade avaliação - 25

As atividades deste fluxograma têm o objetivo de avaliar o projeto e produto produzido e registrar resultados e sugestões de melhorias para próximos ciclos.



Figura 4.35 - Avaliação-25

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

5. Padrão de uso dos documentos

5.1. Propriedade dos documentos

Deverá ser preenchido os campos da propriedade dos documentos para que ele possa ser carregado automaticamente no corpo do documento, conforme figura abaixo.

As informações contidas nestes campos serão transportadas para o documento Word.

The image shows a Windows-style dialog box titled "Propriedades de 0 - Homologação". It has several tabs: "Geral", "Resumo", "Estatísticas", "Conteúdo", and "Personalizar". The "Geral" tab is active. The fields are as follows:

- Título: <Nome do Componente>
- Assunto: <Nome do Projeto>
- Autor: <Nome do Autor>
- Gerente: CITAR - INPE
- Empresa: <Nome da Inst. Parceira>
- Categoria: Plano de Homologação
- Palavras-chave: Processo de Desenvolvimento de SE
- Comentários: Este documento é resultado da proposta de Doutorado de M. Magda Ap. Silvério Miyashiro, pelo INPE
- Base do hiperlink: (empty)
- Modelo: Normal
- Salvar Miniaturas de Todos os Documentos do Word

At the bottom, there are "OK" and "Cancelar" buttons.

As propriedades do documento devem ser preenchidas conforme descrição.

Figura 5.1 - Propriedades do documento Word (*templates*)

5.2. Preenchimento automático do documento

Os modelos (*templates*) utilizados neste processo foram criados com o formato de modelo do word.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

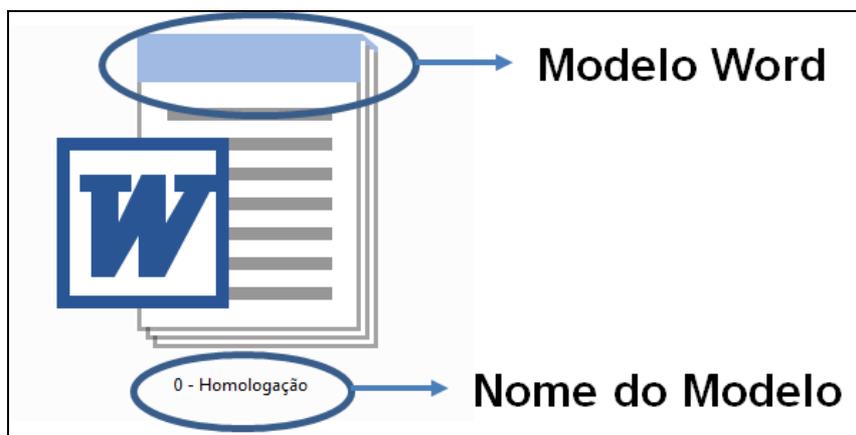
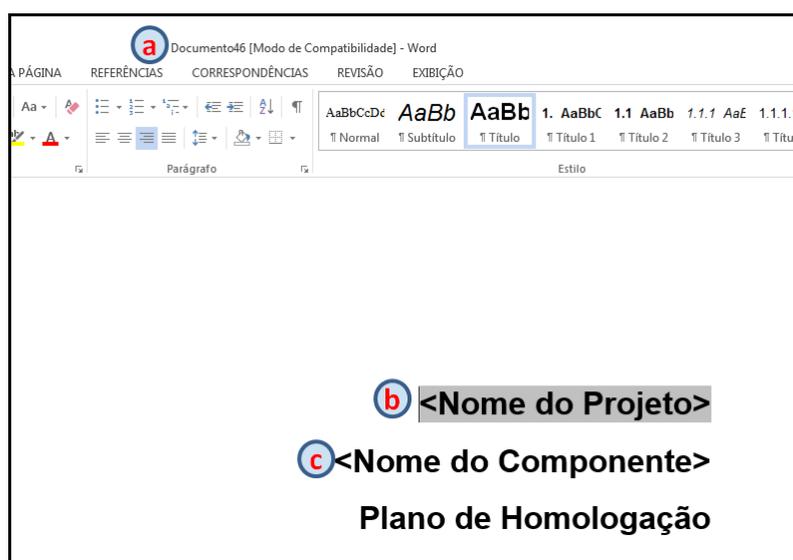


Figura 5.2 - Representação do recurso do *template* Word

Quando o arquivo modelo é editado abre-se um documento imagem ao documento editado (recurso word) que deverá ser salvo com o nome adequado aos documentos do ciclo, conforme item a da figura abaixo.

Os itens b e c do corpo do documento possuem informações que são carregadas das propriedades do documento.

Para puxar estas informações, após preencher os campos das propriedades do documento selecionar o documento todo e acessar F9, que todos os pontos do documento possuem campos automáticos das propriedades são carregados.



Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Figura 5.3 - Representação do *template* Word

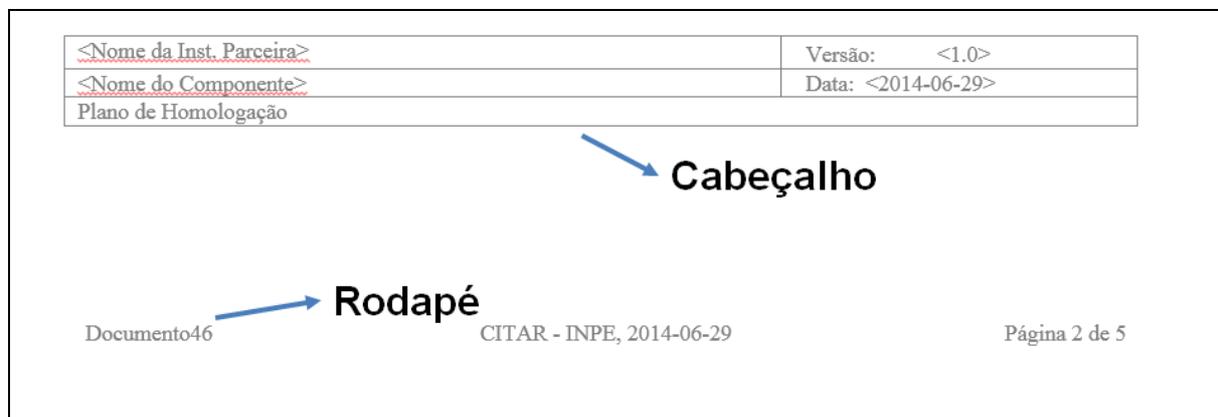


Figura 5.4 - Cabeçalho e rodapé do *template*

Para carregar informações das propriedades do documento no cabeçalho e rodapé, utiliza-se o mesmo conceito do corpo do documento, abre-se o cabeçalho seleciona tudo e F9, repetir o mesmo para o rodapé para atualizar os campos automáticos do rodapé.

5.3. Digitação no interior do documento

Os *templates* utilizados no processo utiliza o mesmo recurso utilizado pelos *templates* do RUP e devem ser utilizados conforme orientações a seguir e representado na figura 5.5.

- a) Todos os itens dos *templates* possuem orientação de como ser preenchido, estas orientações estão descritas em cor azul itálico.
- b) O conteúdo a ser digitado deverá ser digitado na cor automática padrão do documento e
 - Posicionar o curso ao final da linha de orientação sobre o item a ser descrito.
 - Teclar *enter* onde o cursor muda de linha.
 - Iniciar a digitação que já estará com a fonte e cor padronizada.
 - Deletar a linha azul da orientação de preenchimento do item.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

3.1 Componentes

Componente	Descrição	Responsável
<i>[Informe component e]</i>	<i>[Faça uma breve descrição dos do componente]</i>	<i>[Informe o grupo, setor ou organização responsável pelo desenvolvimento do componente]</i>

Obs: Deve-se incluir linhas na tabela acima quantos forem os componentes.

3.2 Ambiente
[Detalhe o ambiente onde o produto (ou os componentes) serão submetidos]

Estou digitando o conteúdo desejado

3.3 Ambiente de Desenvolvimento
[Detalhe o ambiente de desenvolvimento, bem como produtos e ferramentas a serem utilizadas]

3.4 Ambiente de Uso
[Identifique e detalhe todos os itens que devem ser considerados]

- *Tarefas a serem realizadas?*
- *Tempo de utilização?*

Figura 5.5 - Representação da digitação no documento

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

6. Maturidade do processo proposto

A documentação de um processo não deve ser vista como mais uma das atividades a serem realizadas no desenvolvimento do produto. Os documentos gerados em um bom processo são resultado de boas práticas realizadas em todas as suas fases, de forma natural e com a credibilidade daqueles que os utilizam.

Para que um processo seja classificado no nível 2 de maturidade, os projetos da organização devem ser planejados e executados de acordo com uma política. O *status* deve ser facilmente identificado e as pessoas envolvidas devem ser capacitadas de acordo com as necessidades identificadas para produzir as saídas esperadas. Todas as partes interessadas devem ser envolvidas, e os controles e as revisões devem ser realizados por meio de monitoramento e validações. Todas essas atividades devem estar de acordo com a descrição do processo.

6.1. Atendimento às práticas do CMMI

Indicação da cobertura das práticas do CMMI no processo.

6.1.1. Genéricas

A prática genérica auxilia na implementação efetiva do processo. Deve seguir uma estrutura de ação em um contexto de empresa, em que seja descrita e realizada. Em geral, esse tipo de atividade é direcionado para a captação do apoio dos grandes gestores para sua realização.

No processo as práticas genéricas são planejadas e oficializadas conforme tabela 6.1:

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.1 - Práticas genéricas

Código	Descrição	Atividade	Documento	Fase
GP 2.2	Planejar o processo.	Planejamento.	Plano de desenvolvimento - Apêndice C, processo de desenvolvimento - Apêndice B, documentos externos.	Engenharia, gerência de requisitos.
GP 2.3	Fornecer recursos.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
GP2.4	Designar responsabilidades.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
GP2.5	Treinar pessoas.	Planejamento.	Plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
GP2.6	Controlar produtos de trabalho.	Controle do projeto.	Plano de Desenvolvimento - Apêndice C, homologação - Apêndice H.	Comum.
GP2.7	Identificar e envolver as partes interessadas relevantes.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
GP2.8	Monitorar e controlar o processo.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
GP2.9	Avaliar objetivamente a adesão.	Controle do projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Comum.
GP2.10	Revisar status com a gerência de nível superior.	Controle do projeto.	Plano de desenvolvimento - Apêndice C, documentos externos.	Comum.

- **GP2.2:** a implementação dessa prática genérica é registrada no plano de desenvolvimento durante o planejamento. Deve ser legitimada conforme um plano institucional e/ou plano de desenvolvimento do produto que ofereça garantia de que ela sempre será realizada.
- **GP2.3:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com a elaboração do plano de especificação de equipe, e aprovada e viabilizada pelos níveis superiores.
- **GP2.4:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com a especificação dos papéis e responsabilidades, e aprovada e viabilizada pelos níveis superiores.
- **GP2.5:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com o planejamento de aquisição e treinamento da equipe segundo um plano institucional da organização, e aprovada e viabilizada pelos níveis superiores.
- **GP2.6:** a implementação dessa prática genérica é registrada na homologação, com o planejamento de aceitação do produto, e também no plano de gerenciamento da configuração, segundo critérios de controle e homologação institucional, e no plano de desenvolvimento do produto, sendo aprovada e viabilizada pelos níveis superiores.
- **GP2.7:** a implementação dessa prática genérica é registrada no plano de desenvolvimento, com a definição de todos os responsáveis pelo projeto, tanto por seu controle e sua execução quanto por sua viabilização, este último com autoridade para tomadas de decisões. Essa atividade deve ser previamente aprovada e viabilizada pelos níveis superiores de forma institucional.
- **GP2.8:** a implementação dessa prática genérica é registrada no plano de desenvolvimento com o acompanhamento do projeto, e deve ser aprovada e viabilizada pelos níveis superiores de acordo com diretrizes institucionais.
- **GP2.9:** a implementação dessa prática genérica é registrada no plano de homologação e realizada de acordo com o plano de desenvolvimento, com avaliação e aprovação dos

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

artefatos produzidos em cada fase do projeto. Deve ser aprovada e viabilizada pelos níveis superiores de forma institucional.

- **GP2.10:** a implementação dessa prática genérica é registrada no plano de homologação e no plano de desenvolvimento, ao final de cada fase, e realizada de acordo com o plano de desenvolvimento. Deve ser aprovada e viabilizada pelos níveis superiores de forma institucional.

6.1.2. Específicas de nível 2

6.1.2.1. Gerência de requisitos

A atividade de coleta de requisitos acontece em sua respectiva fase, mas as atividades de controle e alinhamento são realizadas em todas as fases. Serve de base estrutural para as demais atividades, conforme mostra a tabela 6.2.

Tabela 6.2 - Gerência de requisitos

Código	Descrição	Atividade	Documento/Apêndice	Fase
SP1.1	Entender requisitos.	Definição de escopo, definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice F, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP1.2	Obter comprometimento com os requisitos.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.3	Gerenciar mudanças nos requisitos.	Controlar configuração, definir itens de controle.	Configuração - Apêndice G.	Comum.
SP1.4	Manter rastreabilidade bidirecional dos requisitos.	Controlar configuração.	Configuração - Apêndice G (matrix de rastreabilidade).	Comum, gerência de requisitos.
SP1.5	Garantir alinhamento entre produtos de trabalho, planos de projeto e requisitos.	Controle de projeto.	Homologação - Apêndice H.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com o entendimento dos requisitos do componente, bem como seu posicionamento no produto final. Deve ser revisada e validada com os responsáveis, de acordo com o planejado.
- **SP1.2:** a implementação dessa prática específica se dá com a identificação das necessidades de mudança dos requisitos e a análise de suas consequências internas e externas ao componente, bem como com a identificação de mudanças de requisito externas ocorridas e seu impacto no componente.
- **SP1.3:** a implementação dessa prática específica se dá com o controle rígido das mudanças de requisitos que acontecerem, independentemente do estímulo a essa mudança, de acordo com o plano de desenvolvimento e o controle de configuração.
- **SP1.4:** a implementação dessa prática específica se dá com o mapeamento e o controle dos requisitos e seus desdobramentos até o último nível, em que um ponto do requisito consegue-se atingir seus pares, bem como planos de testes, configuração, documentação, entre outros. O nível de controle deve seguir o que foi definido no plano de desenvolvimento.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP1.5:** a implementação dessa prática específica se dá com a realização das atividades de acompanhamento e a homologação dos artefatos produzidos em cada fase do processo, de acordo com o plano de desenvolvimento.

6.1.2.2. Medição e análise

As atividades da área de processo de medição e análise nascem na fase de engenharia e repetem-se em todas as fases, por meio das atividades comuns. As distribuições da cobertura de suas práticas ocorrem conforme disposto na Tabela 6.3.

Tabela 6.3 - Medição e análise

Código	Descrição	Atividade	Documento	Fase
SP1.1	Estabelecer objetivos de medição.	Planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Especificar medidas.	Planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.3	Especificar procedimentos de coleta e armazenamento de dados.	Planejamento, planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.4	Especificar procedimento de análise.	Planejamento, planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Obter dados resultantes de medição.	Planejamento, planejar métricas.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Analisar dados resultantes de medição.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP2.3	Armazenar dados resultantes.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP2.4	Comunicar resultados.	Avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Avaliação do ciclo.

- **SP1.1** :a implementação dessa prática específica se dá com a realização das atividades de identificação da necessidade de medição dos itens a serem utilizados no projeto e no planejamento.
- **SP1.2:** a implementação dessa prática específica se dá com a realização das atividades de definição das medidas a serem feitas para o projeto, de acordo com a necessidade.
- **SP1.3:** a implementação dessa prática específica se dá com a realização das atividades de definição das medidas a serem feitas se a especificação de meios de armazenamento dessas medidas.
- **SP1.4:** a implementação dessa prática específica se dá com a definição da análise dos itens medidos e a comunicação aos níveis superiores e aos interessados, de acordo com o plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com a definição da estrutura de armazenamento e de análise dos dados, de acordo com o plano de desenvolvimento.
- **SP2.2:** implementação dessa prática específica se dá com a análise dos dados armazenados, de acordo com o plano de desenvolvimento e homologado com os níveis superiores.
- **SP2.3:** a implementação dessa prática específica se dá com a documentação dos resultados da análise, de acordo com o plano de desenvolvimento.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP2.4:** a implementação dessa prática específica se dá com a homologação dos resultados dos dados de medidos e analisados, de acordo com o plano de desenvolvimento ao final de cada fase e do projeto.

6.1.2.3. Acompanhamento e controle de projeto

As atividades da área de processo, de acompanhamento e controle de projeto, iniciam-se na fase de engenharia, conforme apresentado na Tabela 6.4, e estendem-se nas atividades comuns.

Tabela 6.4 - Acompanhamento e controle de projeto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Monitorar os parâmetros de planejamento do projeto.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.2	Monitorar compromissos.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.3	Monitorar riscos do projeto.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.4	Monitorar a gestão de dados.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.5	Monitorar o envolvimento das partes interessadas.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.6	Conduzir revisões de progresso.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP1.7	Conduzir revisões de macro.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP2.1	Analisar as questões.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP2.2	Implementar ações corretivas.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.
SP2.3	Gerenciar ações corretivas.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia.

- **SP1.1:** a implementação dessa prática específica se dá com o estudo e a documentação dos parâmetros controlados pelo projeto e seus resultados, de acordo com o plano de desenvolvimento.
- **SP1.2:** a implementação dessa prática específica se dá com a homologação do acompanhamento e o monitoramento das medições realizadas durante a realização do projeto em cada fase, de acordo com o plano de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com o acompanhamento dos riscos identificados e documentados no plano do projeto e revisados em cada fase do projeto, de acordo com o plano de desenvolvimento.
- **SP1.4:** a implementação dessa prática específica se dá com a realização da atividade de homologação feita em cada fase do projeto, de acordo com o plano de desenvolvimento.
- **SP1.5:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento.
- **SP1.6:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento, e também deve ser feita sempre que necessário.
- **SP1.7:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP2.1:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento, e também deve ser feita sempre que necessário.
- **SP2.2:** a implementação dessa prática específica se dá com a realização da atividade de homologação em cada fase do projeto, de acordo com o plano de desenvolvimento, e também deve ser feita sempre que necessário, com base nas questões identificadas.
- **SP2.3:** a implementação dessa prática específica se dá com a realização de atividades identificadas para solucionar questões críticas, devendo ser incluída no plano de desenvolvimento e recebera acompanhamento específico.

6.1.2.4. Gerência de acordo com fornecedores

As atividades da área de processo de gerência de acordo com fornecedores devem ter sua práticas voltadas para todos os tipos de fornecedores envolvidos no processo, entre eles fornecedores de componentes, de *softwares*, de ferramentas para o desenvolvimento, enfim, de todas as aquisições que acontecerão durante o projeto, conforme Tabela 6.5.

Tabela 6.5 - Gerência de acordos com fornecedores

Código	Descrição	Atividade	Documento	Fase
SP1.1	Determinar o tipo de aquisição.	Planejamento, planejamento do sistema, projeto do sistema.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Engenharia, gerência de requisitos.
SP1.2	Selecionar fornecedores.	Planejamento.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Comum, gerência de requisitos.
SP1.3	Estabelecer contratos com fornecedores.	Planejamento.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Comum, gerência de requisitos.
SP2.1	Executar contrato com fornecedores.	Planejamento.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D.	Comum, gerência de requisitos.
SP2.2	Aceitar o produto adquirido.	Controle de projeto.	Plano de desenvolvimento - Apêndice C, especificação do produto - Apêndice D, homologação - Apêndice H, projeto de teste - Apêndice F.	Comum.
SP2.3	Garantir a transição de produtos.	Controle de projeto.	Plano de desenvolvimento - Apêndice C.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com a identificação dos componentes que poderão ser adquiridos no mercado ou reutilizados de outros projetos. Deve ser incluída no plano de desenvolvimento, e sua documentação deve ser anexada à documentação produzida pelo projeto, entrando no processo normal de desenvolvimento, de acordo com o plano.
- **SP1.2:** a implementação dessa prática específica se dá com a identificação dos fornecedores dos componentes que serão adquiridos no mercado ou reutilizados de outros projetos, devendo seguir os critérios de seleção de fornecedores definidos no plano de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com a oficialização do fornecimento por meio do contrato e sua inclusão na documentação do projeto.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP2.1:** a implementação dessa prática específica se dá com o início e o acompanhamento das atividades desenvolvidas pelo fornecedor e avaliadas na homologação, ao final de cada fase do processo, sempre que for necessário.
- **SP2.2:** a implementação dessa prática específica se dá com o registro e o acompanhamento das atividades desenvolvidas pelo fornecedor e avaliadas na homologação, ao final de cada fase do processo, sempre que for necessário.
- **SP2.3:** a implementação dessa prática específica se dá com o acompanhamento dos produtos desenvolvidos pelo fornecedor e avaliados na homologação, ao final de cada fase do processo, sempre que for necessário.

6.1.2.5. Planejamento de projeto.

As atividades da área de processo de planejamento de projeto nascem na fase de engenharia e estendem-se por todo o ciclo de vida. Elas devem ser realizadas com muita propriedade, pois servem de base para a realização das demais atividades. O processo proposto cobre as práticas dessa área de processo, conforme mostrado na Tabela 6.6.

Tabela 6.6 - Planejamento de projeto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Estimar o escopo do projeto.	Definição de escopo.	Especificação do produto - Apêndice F, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia
SP1.2	Estabelecer estimativas para atributos de produtos de trabalho e de tarefas.	Especificação dos requisitos.	Especificação do produto - Apêndice F, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Gerência dos requisitos.
SP1.3	Definir fases do ciclo de vida do projeto.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia.
SP1.4	Estimativas de esforço e custo.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Estabelecer orçamento e cronograma.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Identificar riscos do projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.3	Plano de gerenciamento de dados.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.4	Planejar recursos do projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.5	Planejar habilidades e conhecimentos necessários.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.6	Planejar o envolvimento das partes interessadas.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.7	Estabelecer o plano do projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.1	Revisar planos que afetam o projeto.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C, homologação - Apêndice H.	Engenharia, comum.
SP3.2	Conciliar carga de trabalho e recursos.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.3	Obter comprometimento com o plano.	Planejamento, controle do projeto.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.

- **SP1.1:** a implementação dessa prática específica se dá com a identificação e a definição da estrutura do componente a ser desenvolvido para sua estimativa.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP1.2:** a implementação dessa prática específica se dá com a identificação dos atributos do componente a ser desenvolvido, que serão considerados para a estimativa do projeto.
- **SP1.3:** a implementação dessa prática específica se dá com a organização das fases e atividades que serão realizadas durante o projeto.
- **SP1.4:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, identificando a estimativa de custo e tempo levando em consideração as estimativas indicadas pelo produto.
- **SP2.1:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, de acordo com o plano de desenvolvimento.
- **SP2.2:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, avaliando o risco do projeto de acordo com o plano de desenvolvimento. É realizada com foco nos riscos do produto.
- **SP2.3:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, identificando os dados que devem ser medidos e acompanhados durante o projeto, de acordo com o plano de desenvolvimento. Esses dados são identificados de acordo com a criticidade do componente e atendem ao produto e ao processo.
- **SP2.4:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, identificando e alocando os recursos humanos necessários para seu desenvolvimento, bem como a especificação e a providência do ambiente necessário para o desenvolvimento do projeto.
- **SP2.5:** a implementação dessa prática específica se dá com a identificação das características técnicas e admirativas para a execução do projeto.
- **SP2.6:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, com a alocação dos recursos humanos para a execução das atividades.
- **SP2.7:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto, com a indicação da estimativa de tempo para a execução de cada atividade.
- **SP3.1:** a implementação dessa prática específica se dá com o acompanhamento do plano de projeto e homologado ao final de cada fase.
- **SP3.2:** a implementação dessa prática específica se dá com a realização das atividades de acompanhamento e o replanejamento do projeto para atender a alguma demanda ou planejamento não cumprido.

6.1.2.6. Gerência da configuração

As atividades da área de processo de Gerência da Configuração têm seu escopo definido na fase de Engenharia; nesta fase são controladas todas as configurações e todas as mudanças que

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

acontecem em todas as atividades de todas as fases e os controles não se resumem aos requisitos e versões de programas ou componentes, mas sim englobam todos os elementos que fazem parte do ambiente de desenvolvimento e implantação do projeto. A distribuição de suas atividades estão representadas na tabela 6.7.

Tabela 6.7 - Gerência de configuração

Código	Descrição	Atividade	Documento	Fase
SP1.1	Identificar itens de configuração.	Definir itens de controle.	Configuração - Apêndice G	Gerência dos requisitos.
SP1.2	Estabelecer um sistema de gestão de configuração.	Planejamento.	Plano de desenvolvimento - Apêndice C, configuração - Apêndice G, homologação - Apêndice H.	Engenharia.
SP1.3	Criar ou liberar baselines.	Controlar configuração.	Configuração - Apêndice G	Comum.
SP2.1	Acompanhar solicitações de mudanças.	Controlar configuração, controle de projeto.	Configuração - Apêndice G, plano de desenvolvimento - Apêndice C.	Comum.
SP2.2	Controlar itens de configurações.	Controlar configuração.	Configuração - Apêndice G.	Comum.
SP3.1	Estabelecer registros de gestão de configuração.	Controlar configuração.	Configuração - Apêndice G.	Comum.
SP3.2	Realizar auditorias de configuração.	Controle de projeto.	Homologação - Apêndice H.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com identificação e planejamento dos itens que deverão ser controlados durante o projeto de acordo com o plano de desenvolvimento.
- **SP1.2:** a implementação dessa prática específica se dá com o estabelecimento do mecanismo de controle de mudanças e configuração.
- **SP1.3:** a implementação dessa prática específica se dá com o registro das mudanças de acordo com o plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com o análise da solicitação da mudança e o registro do impacto no componente e no plano de desenvolvimento.
- **SP2.2:** a implementação dessa prática específica se dá com o registro das mudanças e o controle da configuração utilizada.
- **SP3.1:** a implementação dessa prática específica se dá com o registro das mudanças, o reporte e o controle da configuração utilizada.
- **SP3.2:** a implementação dessa prática específica se dá com a manutenção dos registros e avaliação dos dados da configuração.

6.1.2.7. Garantia da qualidade

No processo proposto, as atividades da área de processo de garantia da qualidade têm a responsabilidade de consolidar a execução e o resultado de todas as atividades do processo e a qualidade do produto construído. Suas atividades são apresentadas na Tabela 6.8.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.8 - Garantia da qualidade de processo e produto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Avaliar objetivamente os processos.	Avaliação do ciclo.	Projeto, homologação - Apêndice H.	Avaliação do ciclo.
SP1.2	Avaliar objetivamente produtos de trabalho.	Teste SW, teste HW.	Plano de desenvolvimento - Apêndice C, plano de teste - Apêndice F, projeto, homologação - Apêndice H.	Verificação e validação.
SP2.1	Comunicar e resolver problemas de descumprimento.	Fechamento, avaliação do ciclo.	Projeto, Homologação - Apêndice H.	Avaliação do ciclo.
SP2.2	Estabelecer registros.	Avaliação do ciclo.	Homologação - Apêndice H.	Avaliação do ciclo.

- **SP1.1:** a implementação dessa prática específica se dá com a realização das atividades de avaliação do andamento do projeto do componente, de acordo com o plano do projeto e com a descrição, bem como de avaliação dos artefatos gerados com relação à especificação do produto.
- **SP1.2:** a implementação dessa prática específica se dá com a realização das atividades dos artefatos produzidos com relação à especificação do produto.
- **SP2.1:** a implementação dessa prática específica se dá com a realização das atividades homologação em cada fase do projeto, registrando e comunicando as não conformidades.
- **SP2.2:** a implementação dessa prática específica se dá com a realização das atividades de planejamento do projeto com o registro das avaliações realizadas.

6.1.3. Específicas de nível 3

6.1.3.1. Desenvolvimento de requisitos

No processo proposto esta area de processo é atendida com o detalhamento do requisito, sua validação nos cenários necessários, e suas atividades são apresentadas na tabela 6.9.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.9 - Desenvolvimento de requisitos

Código	Descrição	Atividades	Documento	Fase
SP1.1	Levantar necessidades.	Definição de escopo, definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP1.2	Transformar as necessidades dos envolvidos em requisitos do cliente.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Estabelecer requisitos de produto e de componente de produto.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP2.2	Alocar requisitos de componente de produto.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.3	Identificar requisitos de interface.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP3.1	Estabelecer conceitos operacionais e cenários.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP3.2	Estabelecer uma definição da funcionalidade requerida e atributos de qualidade.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP3.3	Analisar requisitos.	Definição dos requisitos, definição dos componentes, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, homologação - Apêndice H.	Engenharia, gerência de requisitos, comum.
SP3.4	Analisar requisitos visando ao balanceamento.	Definição dos requisitos, definição dos componentes, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos, comum.
SP3.5	Validar requisitos.	Controle de projeto.	Homologação - Apêndice H.	Comum.

- **SP1.1:** a implementação dessa prática específica se dá com o levantamento dos dados preliminares e detalhados do produto/componente a ser desenvolvido nas fases de engenharia e gerência de requisitos, respectivamente.
- **SP1.2:** a implementação dessa prática específica acontece com o detalhamento dos requisitos do usuário na especificação geral do componente.
- **SP2.1:** a implementação dessa prática específica é resultado do acompanhamento do projeto e homologação, quando, ao longo do processo, é avaliada a adesão dos componentes ao produto.
- **SP2.2:** a implementação dessa prática específica se dá com a atividade de distribuição das funcionalidades e da modelagem do componente.
- **SP2.3:** a implementação dessa prática específica se dá com a atividade de distribuição das funcionalidades e da modelagem do componente.
- **SP3.1:** a implementação dessa prática específica se dá com as atividades de identificação e construção do ambiente operacional e de desenvolvimento do componente.
- **SP3.2:** a implementação dessa prática específica se dá com a identificação, definição e especificação dos requisitos especiais do componente.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP3.3:** a implementação dessa prática específica se dá com a atividade de homologação e verificação dos requisitos de acordo com o plano, ao longo do projeto.
- **SP3.4:** a implementação dessa prática específica se dá com a identificação, definição e especificação dos requisitos especiais e restrições do componente.
- **SP3.5:** a implementação dessa prática específica se dá com as atividades de homologação do componente em cada fase do projeto, de acordo com o planejamento.

6.1.3.2. Solução técnica

O atendimento dessa área de processo se faz com o estudo e a proposta de soluções para o desenvolvimento ou a aquisição do componente. Suas atividades são apresentadas na Tabela 6.10.

Tabela 6.10 - Solução técnica

Código	Descrição	Atividades	Documento	Fase
SP1.1	Desenvolver soluções alternativas e critérios de seleção.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, Gerencia de Requisitos
SP1.2	Selecionar soluções de componentes de produto.	Definição dos requisitos, definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, Gerencia de Requisitos
SP2.1	Desenvolver o design do produto ou dos componentes de produto.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP2.2	Estabelecer um pacote de dados técnicos.	Controlar configuração.	Configuração.	Comum.
SP2.3	Projetar interfaces utilizando critérios.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP2.4	Analisar alternativas: desenvolver, comprar ou reusar.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP3.1	Implementar design.	Definição dos componentes.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, documentos externos.	Engenharia, gerência de requisitos.
SP3.2	Elaborar a documentação de suporte ao produto.	Não.		

- **SP1.1:** a implementação dessa prática específica se dá com o estudo de soluções e a proposta de soluções, de acordo com os critérios que foram definidos no plano de desenvolvimento.
- **SP1.2:** a implementação dessa prática específica se dá com o estudo de soluções e a proposta de soluções, de acordo com os critérios que foram definidos no plano de desenvolvimento.
- **SP2.1:** a implementação dessa prática específica se dá com a realização da atividade de modelagem do componente.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP2.2:** a implementação dessa prática específica se dá com a definição, o controle e a armazenagem da configuração de todos os artefatos produzidos durante o projeto.
- **SP2.3:** a implementação dessa prática específica se dá com a realização da atividade de modelagem do componente.
- **SP2.4:** a implementação dessa prática específica se dá com o estudo de soluções e a proposta de soluções, de acordo com os critérios que foram definidos no plano de desenvolvimento.
- **SP3.1:** a implementação dessa prática específica se dá com a implementação do componente.
- **SP3.2:** a implementação dessa prática específica não pode ser considerada atendida, pois direciona para a confecção da documentação para o usuário final (entende-se por usuário final o usuário do produto). Esse processo foca o desenvolvimento de um componente que não é um usuário convencional, por isso é considerado não realizado, porém, seguindo a visão de desenvolvimento, todos os documentos produzidos pelo projeto poderiam ser considerados uma documentação, mas não documentação para o usuário final.

6.1.3.3. Integração de produto

Essa área de processo é atendida com a realização das atividades de integração do processo proposto e realizada na fase de integração. Suas atividades são apresentadas na Tabela 6.11.

Tabela 6.11 - Integração de produto

Código	Descrição		Documento	Fase
SP1.1	Estabelecer uma estratégia de integração.	Planejamento Projeto do Componente	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP1.2	Estabelecer ambiente de integração do produto.	Controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Gerência de requisitos, comum.
SP1.3	Estabelecer procedimentos e critérios para integração do produto.	Planejamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Revisar descrições de interfaces para assegurar completude.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP2.2	Gerenciar interfaces.	Controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Gerência de requisitos, comum.
SP3.1	Confirmar se os componentes do produto estão prontos para serem integrados.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP3.2	Montar componentes do produto.	Integração	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Implementação e integração.
SP3.3	Avaliar componentes de produto montados.	Controle de projeto.	Homologação - Apêndice H.	Comum.
SP3.4	Empacotar e entregar produto ou componente de produto.	Fechamento.	Projeto.	Avaliação do ciclo.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP1.1:** a implementação dessa prática específica se dá com a modelagem do componente e planejamento.
- **SP1.2:** a implementação dessa prática específica se dá com o controle da configuração, de acordo com o planejamento.
- **SP1.3:** a implementação dessa prática específica se dá com a modelagem do componente e planejamento.
- **SP2.1:** a implementação dessa prática específica se dá com a realização das atividades de homologação, de acordo com o planejamento.
- **SP2.2:** a implementação dessa prática específica se dá com realização das atividades de controle da configuração, de acordo com o planejamento.
- **SP3.1:** a implementação dessa prática específica se dá com realização das atividades de controle de projeto e homologação, de acordo com o planejamento.
- **SP3.2:** a implementação dessa prática específica se dá com a realização da atividade de integração do produto.
- **SP3.3:** a implementação dessa prática específica se dá com a realização da atividade de controle de projeto, de acordo com o planejamento.
- **SP3.4:** a implementação dessa prática específica se dá com realização da atividade de fechamento na fase de avaliação do ciclo, de acordo com o planejamento.

6.1.3.4. Verificação

As atividades de verificação são planejadas durante o planejamento do projeto e realizadas na fase de verificação e validação. Suas atividades são apresentadas na Tabela 6.12.

Tabela 6.12 - **Verificação**

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Código	Descrição	Atividades	Documento	Fase
SP1.1	Selecionar produtos de trabalho para verificação.	Planejamento, testes SW, testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Engenharia, verificação e validação.
SP1.2	Estabelecer ambiente de verificação.	Planejamento e controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Engenharia, comum.
SP1.3	Estabelecer procedimentos e critérios de verificação.	Planejamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Preparar-se para revisão por pares.	Não.		
SP2.2	Conduzir revisão por pares.	Não.		
SP2.3	Analisar dados de revisão por pares.	Não.		
SP3.1	Realizar verificação.	Testes SW, testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.
SP3.2	Analisar resultados da verificação.	Testes SW, testes HW e fechamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação e avaliação do ciclo.

- **SP1.1:** a implementação dessa prática específica se inicia com a realização da atividade de planejamento e continua com as atividades de testes de *software* e *hardware*.
- **SP1.2:** a implementação dessa prática específica se dá com a realização da atividade, a definição e a preparação do ambiente de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com a realização da atividade de planejamento.
- **SP2.1:** essa prática específica é considerada não atendida, pois esse processo também pode ser utilizado para desenvolvimento individual. Contudo, essa prática pode ser planejada para a atividade de planejamento do projeto.
- **SP2.2:** essa prática específica é considerada não atendida, pois esse processo também pode ser utilizado para desenvolvimento individual. Contudo, essa prática pode ser planejada para a atividade de planejamento do projeto.
- **SP2.3:** essa prática específica é considerada não atendida, pois esse processo também pode ser utilizado para desenvolvimento individual. Contudo, essa prática pode ser planejada para a atividade de planejamento do projeto.
- **SP3.1:** a implementação dessa prática se dá com as atividades de testes de *software* e testes de *hardware*, de acordo com o planejamento e o plano de testes.
- **SP3.2:** a implementação dessa prática se dá com as atividades de homologação e a realização da fase de avaliação do ciclo, de acordo com o planejamento.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

6.1.3.5. Validação

As atividades das fases de verificação e validação atendem à área de processo ao longo do projeto, segundo seu planejamento e homologação de avaliação do ciclo. Suas atividades são apresentadas na Tabela 6.13.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.13 - Validação

Código	Descrição	Atividades	Documento	Fase
SP1.1	Selecionar produtos para validação.	Testes SW e testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.
SP1.2	Estabelecer ambiente de validação.	Controlar configuração.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos, configuração - Apêndice G.	Comum
SP1.3	Estabelecer procedimentos e critérios de validação.	Planejamento.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.1	Realizar validação	Testes SW e testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.
SP2.2	Analisar resultados de validação.	Testes SW e testes HW.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, projeto de teste - Apêndice F.	Verificação e validação.

- **SP1.1:** a implementação dessa prática específica se dá com a realização da atividade de planejamento.
- **SP1.2:** a implementação dessa prática específica se dá com a realização da atividade, a definição e a preparação do ambiente de desenvolvimento.
- **SP1.3:** a implementação dessa prática específica se dá com a realização da atividade de planejamento.
- **SP2.1:** a implementação dessa prática se dá com as atividades de testes de *software* e testes de *hardware*, de acordo com o planejamento e o plano de testes.
- **SP2.2:** a implementação dessa prática se dá com as atividades de homologação e a realização da fase de avaliação do ciclo, de acordo com o planejamento.

6.1.3.6. Foco no processo da organização

Essa área de processo não é atendida diretamente pelo processo proposto, pois trata-se de atividades realizadas de forma organizacional. As atividades dessa área de processo são apresentadas na Tabela 6.14.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.14 - Foco no processo da organização

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer necessidades de processo da organização.	Não.		
SP1.2	Avaliar os processos da organização.	Não.		
SP1.3	Identificar melhorias para os processos da organização.	Não.		
SP2.1	Estabelecer planos de ação de processo.	Não.		
SP2.2	Implementar planos de ação de processo.	Não.		
SP3.1	Implantar ativos de processo da organização.	Não.		
SP3.2	Implantar processos padrão.	Não.		
SP3.3	Monitorar a implementação.	Não.		
SP3.4	Incorporar experiências em ativos de processos organizacionais.	Não.		

As práticas da área de processo com foco no processo da organização não são consideradas atendidas, pois essa área de processo está relacionada com a implementação de processos padrão na organização. Tais práticas são realizadas neste processo, mas, para atender às necessidades do processo individualmente, que necessita de idade e adaptação de acordo com a necessidade do componente a ser produzido, o foco que esta área de processo exige é de nível organizacional, sendo assim, inviável de atender.

6.1.3.7. Definição do processo da organização.

Essa área de processo é atendida pelo processo proposto, definindo e seguindo diretrizes pertinentes ao projeto de forma individual. Suas atividades são apresentadas na Tabela 6.15.

Tabela 6.15 - Definição do processo da organização

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer processos padrão.	Não.		
SP1.2	Estabelecer descrições de modelos de ciclo de vida.	Planejamento.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia.
SP1.3	Estabelecer critérios e diretrizes para adaptação.	Planejamento.	Especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia.
SP1.4	Estabelecer o repositório de medições da organização.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP1.5	Estabelecer a biblioteca de ativos de processo da organização.	Controle do projeto.	Plano de desenvolvimento - Apêndice C.	Comum.
SP1.6	Estabelecer padrões de ambiente de trabalho.	Controlar configuração, planejamento.	Configuração - Apêndice G, plano de desenvolvimento - Apêndice C.	Comum, gerência de requisitos.
SP1.7	Estabelecer regras e orientações para equipes.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.

- **SP1.1:** essa prática não é considerada atendida, pois o processo oferece flexibilidade para sua adaptação, de acordo com a necessidade do componente a ser produzido, inviabilizando assim a cobertura da prática.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP1.2:** essa prática é atendida na realização da atividade de planejamento do projeto, com a adaptação do projeto, de acordo com o componente a ser produzido.
- **SP1.3:** o atendimento dessa prática se dá com a realização da atividade de planejamento do projeto, com a adaptação do projeto, de acordo com o componente a ser produzido.
- **SP1.4:** o atendimento dessa prática se dá com a realização da atividade de controle de projeto ao longo do ciclo de vida, de acordo com o planejado.
- **SP1.5:** o atendimento dessa prática se dá com a realização e definição dos itens a serem controlados pela atividade de controlar configuração.
- **SP1.6:** o atendimento dessa prática se dá com a definição do ambiente de configuração pelas atividades de controlar configuração e planejamento.
- **SP1.7:** o atendimento dessa prática se dá com a realização das atividades de planejamento.

6.1.3.8. Treinamento organizacional

Essa área de processo é parcialmente atendida, uma vez que no processo proposto são definidas estratégias de capacitação e treinamento dos envolvidos, de acordo com o necessário para atender ao projeto, porém não define diretrizes organizacionais. Suas atividades são apresentadas na Tabela 6.16.

Tabela 6.16 - Treinamento organizacional

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer necessidades estratégicas de treinamento.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.2	Identificar as necessidades de treinamento sob responsabilidade da organização.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.3	Estabelecer um plano tático de treinamento na organização.	Não.		
SP1.4	Estabelecer a capacidade de treinamento.	Não.		
SP2.1	Fornecer treinamento.	Não.		
SP2.2	Estabelecer registros de treinamento.	Não.		
SP2.3	Avaliar a eficácia dos treinamentos.	Não.		

- **SP1.1 e SP1.2:** essas práticas são atendidas na realização da atividade de planejamento do projeto, com a alocação e a preparação da equipe.
- **SP1.3 a SP2.3:** as práticas na área de processo de treinamento organizacional devem ser realizadas de acordo com as diretrizes organizacionais, e o plano de treinamento indicado nesse processo se dá a partir das necessidades do projeto, de acordo com o componente a ser

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

produzido, por isto essas práticas não são consideradas atendidas no contexto requerido pela área de processo.

6.1.3.9. Gerência integrada de projeto

Essa área é atendida parcialmente pelo processo, pois realiza atividades de integração e controle do projeto do componente individual, bem como de todo o sistema, ao longo do desenvolvimento do projeto. Assim, pode não atender como um todo, uma vez que não está articulada com processos da organização. Suas atividades são apresentadas na Tabela 6.17.

Tabela 6.17 - Gerência integrada de projeto

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer o processo definido para o projeto.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.2	Utilizar os ativos de processo organizacional para planejar as atividades do projeto.	Não.		
SP1.3	Estabelecer o ambiente de trabalho do projeto.	Não.		
SP1.4	Integrar planos.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.5	Gerenciar o projeto utilizando planos integrados.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.6	Estabelecer equipes.	Planejamento.	Plano de desenvolvimento - Apêndice C.	Engenharia, gerência de requisitos.
SP1.7	Contribuir para os ativos de processos organizacionais.	Controlar configuração.	Configuração - Apêndice G, plano de desenvolvimento - Apêndice C.	Comum.
SP2.1	Gerenciar o envolvimento das partes interessadas.	Planejamento, controle de projeto, homologação.	Plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Gerenciar dependências.	Definição dos requisitos, definição dos componentes, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C, documentos externos.	Engenharia, gerência de requisitos.
SP2.3	Solucionar questões críticas de coordenação.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.

- **SP1.1:** o atendimento dessa prática se dá com a realização das atividades de planejamento do projeto, a partir do ciclo de vida definido para ele.
- **SP1.2:** essa prática não é interpretada como não atendida, pois no processo sugerido está relacionada ao projeto do componente, e não segue as diretrizes organizacionais.
- **SP1.3:** essa prática não é interpretada como não atendida, pois no processo sugerido está relacionada ao projeto do componente, e não segue as diretrizes organizacionais.
- **SP1.4:** essa prática é realizada com a atividade de planejamento do projeto.
- **SP1.5:** essa prática é realizada com a atividade de planejamento do projeto.
- **SP1.6:** essa prática é realizada com a atividade de planejamento do projeto.
- **SP1.7:** essa prática é realizada no controle da configuração, de acordo com o planejado.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP2.1:** essa prática é realizada na atividade de homologação em controle do projeto.
- **SP2.2:** essa prática é realizada na atividade de homologação em controle do projeto.
- **SP2.3:** essa prática é realizada na atividade de homologação em controle do projeto.

6.1.3.10. Gerência de risco

Essa área é realizada pelo processo proposto, iniciando com seu planejamento e prosseguindo ao longo do projeto. Suas atividades são apresentadas na Tabela 6.18.

Tabela 6.18 - Gerência de risco

Código	Descrição	Atividades	Documento	Fase
SP1.1	Determinar fontes e categorias de riscos.	Planejamento Controle de Projeto	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Definir parâmetros para riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.3	Estabelecer uma estratégia para gestão de riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Identificar riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Avaliar, categorizar e priorizar riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.1	Elaborar planos de mitigação de riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP3.2	Implementar planos de mitigação de riscos.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.

- **SP1.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP1.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP1.3:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP2.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP2.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP3.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.
- **SP3.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante seu andamento, no controle de projeto.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

6.1.3.11. Análise de decisão e resolução

O processo proposto não oferece atividades específicas para a tomada de decisão, porém, de forma indireta, auxilia em sua execução. As atividades dessa área de processo estão representadas na Tabela 6.19.

Tabela 6.19 - Análise de decisão e resolução

Código	Descrição	Atividades	Documento	Fase
SP1.1	Estabelecer diretrizes para análise e decisão.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.2	Estabelecer critérios de avaliação.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.3	Identificar soluções alternativas.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.4	Selecionar métodos de avaliação.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.5	Avaliar soluções alternativas.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.
SP1.6	Selecionar soluções.	Planejamento, homologação e avaliação do ciclo.	Plano de desenvolvimento - Apêndice C, documento de homologação.	Engenharia, comum e avaliação do ciclo.

- **SP1.1, SP1.2, SP1.3, SP1.4, SP1.5e SP1.6:** as práticas dessa área de processo são descritas no planejamento do projeto, realizadas durante o andamento do projeto na atividade de controle de projeto e finalizadas durante a atividade de fechamento.

6.1.4. Específicas de nível 4

6.1.4.1. Desempenho do processo organizacional

O processo proposto em seu planejamento, pela natureza da realização das atividades, atende parcialmente a essa área de processo, pois as boas práticas resultam melhor desempenho do processo. Contudo, seu foco é o desempenho do produto. Suas atividades são apresentadas na Tabela 6.20.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.20 - Desempenho do processo organizacional

Código	Descrição		Documento	Fase
SP1.1	Estabelecer objetivos para qualidade e para desempenho de processo.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Selecionar processos.	Não.		
SP1.3	Estabelecer medidas de desempenho de processo.	Não.		
SP1.4	Analisar a performance do processo e estabelecer baselines de desempenho de processo.	Não.		
SP1.5	Estabelecer modelos de desempenho de processo.	Não.		

- **SP1.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante o andamento do projeto, no controle de projeto.
- **SP1.2 a SP1.5:** essas práticas específicas não são consideradas atendidas, pois a elas se referem as atividades organizacionais e as atividades de avaliação de desempenho. Nesse processo, referem-se parcialmente ao desempenho do processo e, mais profundamente, ao desempenho do componente produzido.

6.1.4.2. Gerência quantitativa de projeto

O processo proposto atende parcialmente à área de processo de gerência quantitativa do projeto, estabelecendo e definindo as atividades de gestão. Suas atividades são apresentadas na Tabela 6.21.

Tabela 6.21 - Gerência quantitativa de projeto

Código	Descrição	Atividade	Documento	Fase
SP1.1	Estabelecer os objetivos do projeto.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Compor o processo definido.	Planejamento, controle de projeto.	Especificação do produto - Apêndice D, especificação do componente - Apêndice E, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.3	Selecionar subprocessos e atributos.	Não.		
SP1.4	Selecionar medidas e técnicas analíticas.	Não.		
SP2.1	Monitorar o desempenho dos subprocessos selecionados.	Não.		
SP2.2	Gerenciar o desempenho do projeto.	Não.		
SP2.3	Realizar a análise de causa raiz.	Não.		

- **SP1.1:** essa prática específica é realizada no planejamento do projeto e acompanhada durante todo o andamento do projeto no controle de projeto.
- **SP1.2:** essa prática específica é realizada no planejamento do projeto e acompanhada durante todo o andamento do projeto no controle de projeto.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

- **SP1.3 a SP2.3:** essas práticas não são consideradas atendidas pelo processo, pois são relacionadas à medição de sua *performance*. Para isso, o processo deve estar definido e ser rigorosamente seguido, o que não é viável para sua flexibilidade, de acordo com o componente a ser produzido.

6.1.5. Específicas de nível 5

6.1.5.1. Análise causal e resolução

O processo atende a essa área desde o planejamento do projeto, indicando pontos de análise e recebendo acompanhamento em todas as fases. A finalização das atividades envolvidas nesse processo é feita na avaliação do ciclo. Suas atividades são apresentadas na Tabela 6.22.

Tabela 6.22 - Análise causal e resolução

Código	Descrição	Atividade	Documento	Fase
SP1.1	Selecionar resultados para análise	Controle de projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP1.2	Analisar causas.	Controle de projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.1	Implementar propostas de ação.	Controle de projeto.	Homologação - Apêndice H, plano de desenvolvimento - Apêndice C.	Engenharia, comum.
SP2.2	Avaliar efeitos de ações implementadas.	Não.		
SP2.3	Registrar dados.	Fechamento.	Projeto.	Avaliação do ciclo.

- **SP1.1 a SP2.1:** essas práticas são realizadas durante o controle do projeto, de acordo com o planejamento da homologação.
- **SP2.2:** essa prática específica é considerada não realizada, pois não é uma ação explícita do processo. Contudo, ela pode ser incluída no planejamento do projeto a fim de ser realizada.
- **SP2.3:** essa prática específica é realizada na fase de avaliação do ciclo com a atividade de fechamento.

6.1.5.2. Gestão do processo organizacional

Na fase de avaliação do ciclo de vida do projeto, em especial na atividade de fechamento do processo, são realizadas as atividades de avaliação do desempenho desse ciclo, buscando entender os resultados alcançados e reestruturar novos ciclos com base nos sucessos e fracassos. Essas atividades são apresentadas na Tabela 6.23.

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

Tabela 6.23 - Gestão do processo organizacional

Código	Descrição	Atividade	Documento	Fase
SP1.1	Manter objetivos de negócio.	Não.		
SP1.2	Analisar dados de desempenho do processo.	Fechamento.	Projeto.	Avaliação do ciclo.
SP1.3	Identificar áreas com potencial para melhorias.	Fechamento.	Projeto.	Avaliação do ciclo.
SP2.1	Identificar melhorias sugeridas.	Fechamento.	Projeto.	Avaliação do ciclo.
SP2.2	Analisar sugestões de melhorias.	Não.		
SP2.3	Validar melhorias.	Fechamento.	Projeto.	Avaliação do ciclo.
SP2.4	Selecionar e aplicar melhorias para implantação.	Fechamento.	Projeto.	Avaliação do ciclo.
SP3.1	Planejar implantação.	Fechamento.	Projeto.	Avaliação do ciclo.
SP3.2	Gerenciar implantação.	Fechamento.	Projeto.	Avaliação do ciclo.
SP3.3	Avaliar os efeitos de melhorias.	Não.		

- **SP1.1 e SP2.2 e SP3.3:** essas práticas específicas são consideradas não atendidas pelo processo, pois não são realizadas ações que as atendam diretamente. Contudo, no processo é possível realizá-las se forem incluídas no planejamento.
- **SP1.2, SP1.3, SP2.1, SP2.3, SP2.4, SP3.1 e SP3.2:** essas práticas específicas são realizadas na avaliação do ciclo, na atividade de fechamento.

7. Referencias

- European Cooperation for Space Standardization (ECSS). ECSS Secretariat ESA-ESTEC ECSS-E-10 Part 1B, Space engineering – Requirements & Standards Division, Noordwijk, The Netherlands, 2004.
- European Cooperation for Space Standardization (ECSS). ECSS Secretariat ESA-ESTEC ECSS-E-ST-40C, - Engenharia espacial – ECSS Secretaria - ESA-ESTEC - Requisitos e Divisão de Normas - Noordwijk, Holanda
- SOFTWARE ENGINEERING INSTITUTE. - CMMI for Development, Version 1.3 CMMI-DEV, V1.3 - CMMI Product Team – Improving processes for developing better products and services - Technical Report - Pittsburgh: Carnegie Mellon University, November 2010.
- MIYASHIRO, M. A. S., Identificação e melhoria do nível de maturidade de uma organização explorando técnicas de inteligência computacional. 2007. 100 f. Dissertação (Mestrado) - Instituto Nacional de Pesquisas Espaciais, São José Dos Campos, 2007.
- Miyashiro, M. A. S, Ferreira, M.G.V, Sannt'ana N. - Viability of Application of the process areas of CMMI-DEV processes Development of Critical Embedded Systems- Science and Information Conference - Londres (SAI2015) aprovado para publicação e apresentação.
- Miyashiro, M. A. S, Ferreira, M.G.V. One approach to the use of the practices of

Processo para o Desenvolvimento de Sistemas Embarcado	Versão:
Instituto Nacional de Pesquisa Espacial Projeto 01	Data:
Laboratório	

CMMI-DEV V1.3 level 2 in a process of development of Embedded Systems - The Fifth International Conference on Information, Intelligence, Systems, and Applications – Grécia (IISA 2014).

- Miyashiro, M. A. S, Ferreira, M.G.V. - Process for the development of Embedded System following the practices of CMMI Level 2 - Science and Information Conference - Londres (SAI2014). Nesta conferencia estão previstas reuniões para a formação de um grupo de estudo em engenharia de *software* para sistemas embarcados.
- Miyashiro, M. A. S, Ferreira, M.G.V. - The Practices of Level 2 (Managed) CMMI-DEV V1.3 in Development of Embedded Systems - International Conference on Electrical and Electronic Engineering – Hong Kong (EEE2014).
- Miyashiro, M. A. S, Ferreira, M.G.V. - Phase Cyclical Process Requirements for the Development Of Embedded Systems - "IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book):2013 III Brazilian Symposium on Computing Systems Engineering" – Brasil (SBESC2013).
- Miyashiro, M. A. S - Uma Abordagem para Processo de Desenvolvimento de Sistema Embarcado que Atende o Nível 2 de Maturidade do CMMI-DEV." - Tese de Doutorado - Instituto Nacional de Pesquisas Espaciais, São José Dos Campos, com data de defesa em 10/02/2015.
- IBM Rational Unified Process (RUP) - <http://www-01.ibm.com/software/rational/rup> acessado em outubro de 2012.

Apêndice C – Plano de desenvolvimento

<Nome Inst. Parceira>

<Nome do Projeto>
<Nome do Componente>
Plano de Desenvolvimento
Versão <1.0>

<Nome do Autor>

<Nome Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Criado em 25/06/2014
Plano de Desenvolvimento	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Nome Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Criado em 25/06/2014
Plano de Desenvolvimento	

Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
1.4	Referências	4
1.5	Visão Geral	4
2.	Visão Geral do Projeto	4
2.1	Finalidade, Escopo e Objetivos do Projeto	4
2.2	Suposições e Restrições	4
2.3	Produtos Liberados do Projeto	4
3.	Organização do Projeto	5
3.1	Estrutura Organizacional	5
3.2	Papéis e Responsabilidades	5
4.	Ambiente de Desenvolvimento	5
4.1	Ferramentas	5
4.2	Ambiente	5
4.3	Infra-estrutura	5
4.4	Padrão de Desenvolvimento	5
4.5	Padrão de Nomes de Documentos	5
5.	Processo de Gerenciamento	6
5.1	Estimativas do Projeto	6
5.2	Plano de Projeto (Hardware/Software)	6
5.2.1	Plano de Fase	6
5.2.2	Objetivos dos ciclos	6
5.2.3	Releases	6
5.2.4	Programação do Projeto	6
5.2.5	Recursos do Projeto	6
5.3	Monitoramento e Controle do Projeto	7
5.3.1	Plano de Gerenciamento de Requisitos	7
5.3.2	Plano de Controle de Cronograma	7
5.3.3	Plano de Controle de Qualidade	7
5.3.4	Plano de Elaboração de Relatórios	7
5.3.5	Plano de Métricas	7
5.3.6	Plano de Gerenciamento de Configuração	7
5.3.7	Plano de Gerenciamento de Riscos	7
5.4	Plano de Aceitação do Produto	7
5.5	Plano de Aquisição	7
5.5.1	Identificação dos Fornecedores Candidatos	7
5.5.2	Crítérios de Seleção	7
5.5.3	Organização contratural com o fornecedor	7
5.5.4	Plano Sumário de desenvolvimento do fornecedor	7
6.	Anexos	7

<Nome Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Criado em 25/06/2014
Plano de Desenvolvimento	

<Plano de Desenvolvimento>

1. Introdução

*[A introdução do **Plano de Desenvolvimento** deve oferecer uma visão geral de todo o documento. Ela deve incluir a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e uma visão geral deste **Plano de Desenvolvimento**]*

1.1 Finalidade

*[Especifique a finalidade deste **Plano de Desenvolvimento**]*

1.2 Escopo

*[Uma breve descrição do escopo deste **Plano de Desenvolvimento** os Projetos aos quais ele está associado e tudo o que é afetado ou influenciado por este documento]*

1.3 Definições, Acrônimos e Abreviações

*[Esta subseção apresenta as definições de todos os termos, acrônimos e abreviações necessários para a correta interpretação do **Plano de Desenvolvimento**. Essas informações podem ser fornecidas mediante referência ao Glossário do projeto]*

1.4 Referências

[Esta subseção apresenta uma lista completa de todos os documentos mencionados no Plano de Desenvolvimento. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações poderão ser fornecidas por menção a um anexo ou outro documento.

No caso do Plano de Desenvolvimento, a lista de artefatos referenciados inclui:

- *Configuração*
- *Especificação de Componente*
- *Especificação do Produto*
- *Projeto de Testes*
- *Homologação do Produto*
- *Processo de Desenvolvimento]*

1.5 Visão Geral

*[Esta subseção descreve o conteúdo restante do **Plano de Desenvolvimento** e explica como o documento está organizado]*

2. Visão Geral do Projeto

2.1 Finalidade, Escopo e Objetivos do Projeto

[Uma breve descrição da finalidade e dos objetivos deste projeto e uma breve descrição dos produtos que se espera que o projeto libere]

2.2 Suposições e Restrições

[Uma lista das suposições em que este plano se baseia e de quaisquer restrições como, por exemplo, de orçamento, equipe, equipamento e programação, que se aplicam ao projeto]

2.3 Produtos Liberados do Projeto

[Uma tabela listando os artefatos a serem criados durante o projeto, incluindo as datas-alvo de liberação (WBS)]

<Nome Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Criado em 25/06/2014
Plano de Desenvolvimento	

3. Organização do Projeto

3.1 Estrutura Organizacional

[Descreva a estrutura organizacional da equipe do projeto, incluindo as autoridades de gerenciamento e outras autoridades de revisão]

3.2 Papéis e Responsabilidades

[Identifique as unidades organizacionais do projeto que serão responsáveis por cada uma das disciplinas, detalhes do fluxo de trabalho e processos de suporte bem como os responsáveis pelas tomadas de decisões.]

4. Ambiente de Desenvolvimento

[Descreva o ambiente de computação e as ferramentas do componente a serem utilizadas para desempenhar as atividades de desenvolvimento que será utilizado para o projeto]

- *tamanho previsto dos dados do produto*
- *distribuição da equipe do produto*
- *localização física dos servidores e clientes]*

4.1 Ferramentas

[Descreva as ferramentas a serem utilizadas no ambiente de computação]

4.2 Ambiente

[Descreva o Ambiente Necessário]

4.3 Infra-estrutura

[Descreva a Infra Estrutura Necessária

- *Ferramentas*
- *Nomenclaturas*
- *Controles*
- *outros]*

4.4 Padrão de Desenvolvimento

[Descreva todos os itens que devem ser padronizados para todos os produtos e ou componentes e que devem ser observados em todos os ciclos de desenvolvimento]

4.5 Padrão de Nomes de Documentos

[Descreva o padrão de nomes que devem ser utilizados para a os documentos criados durante o projeto, os itens que compõem o nome e o significado de cada item, conforme modelo:

NP = Número do Projeto, número sequencial que identifique o projeto

NCP = Número sequencial que identifique o componente do projeto a ser desenvolvido no ciclo

PPP = Sigla do Projeto, Sigla de identificação do projeto, em geral abreviações do nome do projeto

CD = Código do documento, Código de identificação dos documentos do processo

- *Documento de Plano de Desenvolvimento - 02*
- *Documento de Especificação do Produto - 03*
- *Documento de Especificação de Componente - 04*
- *Documento de Projeto de Testes - 05*

<Nome Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Criado em 25/06/2014
Plano de Desenvolvimento	

- *Documento de Configuração - 06*
- *Documento de Homologação do Produto – 07*

NNN = Número sequencial do documento

VV = Versão do documento, Identificação de modificações no documento

Nome do documento

NP-NCP- PPP-CD-NNN-VV]

5. Processo de Gerenciamento

5.1 Estimativas do Projeto

[O planejamento do componente considerar a construção do Hardware e do Software em atividades distintas e compartilhada, onde visualiza-se com nitidez as necessidades, características e ações pertinentes a cada parte do componente (Hw e Sw). Pode-se elaborar um plano de projeto para as duas partes ou elaborar um plano de projeto para cada parte]

5.2 Plano de Projeto (Hardware/Software)

[Forneça a programação e o custo estimado do projeto, assim como a base dessas estimativas, e os pontos e circunstâncias do projeto em que serão feitas novas estimativas]

5.2.1 Plano de Fase

[Inclua o seguinte:

- *Estrutura de Divisão de Trabalho (WBS)*
- *Uma linha de tempo ou um gráfico de Gantt mostrando o tempo alocado para as fases e ciclos do projeto*
- *Identificação dos principais marcos com os respectivos critérios de êxito*

Defina todas as demonstrações e pontos de release importantes

- *Identificação de todos os marcos de validação*

5.2.2 Objetivos dos ciclos

[Liste os objetivos a serem atingidos para cada um dos ciclos]

5.2.3 Releases

[Uma breve descrição de cada release de componente e se é uma versão beta, de demonstração etc.]

5.2.4 Programação do Projeto

[Diagramas ou tabelas mostrando as datas-alvo para a conclusão dos ciclos e fases, dos pontos de release, das demonstrações e de outros marcos

- *Cronograma geral e cronograma detalhado para todas as fases do projeto]*

5.2.5 Recursos do Projeto

5.2.5.1 Plano de Formação de Equipe

[Identifique aqui os números e tipos de recursos humanos necessários, incluindo todas as experiências ou habilidades especiais, programados por fase ou ciclo do projeto]

5.2.5.2 Plano de Aquisição de Recursos

[Descreva como você pretende localizar e adquirir as pessoas para integrarem a equipe necessária ao projeto]

<Nome Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Criado em 25/06/2014
Plano de Desenvolvimento	

5.2.5.3 Plano de Treinamento

[Liste quaisquer treinamentos especiais necessários aos integrantes da equipe do projeto, com as datas-alvo identificando quando os treinamentos deverão ser concluídos]

5.3 Monitoramento e Controle do Projeto

[Nos itens abaixo quando houver indicar o documento utilizado]

5.3.1 Plano de Gerenciamento de Requisitos

[Descreva a abordagem adotada para monitorar os requisitos]

5.3.2 Plano de Controle de Cronograma

[Descreva a abordagem adotada para monitorar o andamento tendo em vista o cronograma planejado e como executar ações corretivas quando necessário]

5.3.3 Plano de Controle de Qualidade

[Descreva o andamento e os métodos a serem usados para controlar a qualidade dos produtos liberados do projeto e como executar ações corretivas quando necessário]

5.3.4 Plano de Elaboração de Relatórios

[Descreva os relatórios internos e externos a serem gerados, e a frequência e distribuição de publicação]

5.3.5 Plano de Métricas

[Descreva a abordagem adotada para definir e monitorar as métricas]

5.3.6 Plano de Gerenciamento de Configuração

[Descreva a abordagem adotada para controlar a configuração]

5.3.7 Plano de Gerenciamento de Riscos

[Descreva a abordagem adotada para gerenciar riscos]

5.4 Plano de Aceitação do Produto

[Descreva a abordagem adotada e quando houver indicar o documento utilizado]

5.5 Plano de Aquisição

[Descreva a abordagem adotada e quando houver indicar o documento utilizado]

5.5.1 Identificação dos Fornecedores Candidatos

[Descreva a lista de fornecedores capazes de atender aos requisitos]

5.5.2 Critérios de Seleção

[Descreva a abordagem adotada analisar e identificar os fornecedores candidatos]

5.5.3 Organização contratural com o fornecedor

[Indicar o contrato que será utilizado com o fornecedor, (nome do arquivo, forma de armazenamento), responsáveis pelo contrato, (contratante e contratado) incluir cópia do contrato de fornecimento a documentação do projeto]

5.5.4 Plano Sumário de desenvolvimento do fornecedor

[Indicar de forma suficiente o plano de trabalho do fornecedor para o desenvolvimento do componente]

6. Anexos

*[Material adicional de uso para o leitor do **Plano de Desenvolvimento**]*

Apêndice D – Especificação do produto

<Nome da Inst. Parceira>

<Colocar o nome do Projeto>

Especificação do Produto

<Colocar Nome do Produto>

Versão <1.0>

<Nome do Autor>

<Nome da Inst. Parceira>	Versão: <1.0>
<Colocar Nome do Componente>	Data: 6/25/2014
Especificação de Produto	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Nome da Inst. Parceira>	Versão: <1.0>
<Colocar Nome do Componente>	Data: 6/25/2014
Especificação de Produto	

Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo Geral	4
1.3	Referências	4
1.4	Visão Geral	4
2.	Posicionamento.	4
2.1	Descrição Geral do Projeto	4
2.2	Representação Gráfica do Produto	4
3.	Identificação dos Componentes Envolvidos e Responsáveis	4
3.1	Componentes	5
3.2	Ambiente	5
3.3	Ambiente de Desenvolvimento	5
3.4	Ambiente de Uso	5
3.5	Descrição do componente a ser produzido.	5
3.5.1	<Nome do Componente>	5
4.	Recursos do Produto	5
5.	Restrições	5
6.	Faixas de Qualidade	6
7.	Precedência e Prioridade	6
8.	Outros Requisitos do Produto	6
8.1	Padrões Aplicáveis	6
8.2	Requisitos do Sistema	6
8.3	Requisitos de Desempenho	6
8.4	Requisitos Ambientais	6
9.	Requisitos da Documentação	6
9.1	Documentos legais	6
9.2	Outros documentos	6

<Nome da Inst. Parceira>	Versão: <1.0>
<Colocar Nome do Componente>	Data: 6/25/2014
Especificação de Produto	

<Especificação de Produto>

1. Introdução

[A introdução do documento de Visão oferece uma visão geral de todo o documento. Ela contém a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral deste documento Visão]

1.1 Finalidade

[Especifique a finalidade deste documento Visão]

1.2 Escopo Geral

[Uma breve descrição do escopo deste documento Visão; a que Projeto(s) ele está associado e tudo o mais que seja afetado ou influenciado por este documento]

1.3 Referências

[Esta subseção apresenta uma lista completa de todos os documentos mencionados no documento de Visão. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento]

1.4 Visão Geral

[Esta subseção descreve o que o restante do documento Visão contém e explica como o documento está organizado]

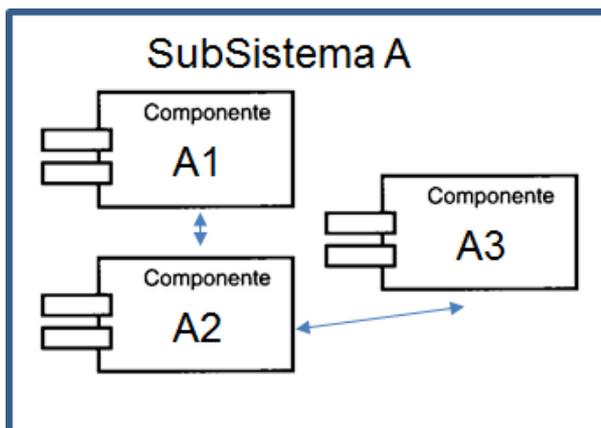
2. Posicionamento.

2.1 Descrição Geral do Projeto

[Faça uma breve descrição do Projeto de forma completa (descrição do sistema ou subsistema completo a ser desenvolvido onde será inserido o componente a ser desenvolvido)]

2.2 Representação Gráfica do Produto

[Forneça uma representação gráfica do produto a ser construído. A técnica/ferramenta a ser utilizado deverá aquela definida no plano de desenvolvimento (escopo)]



3. Identificação dos Componentes Envolvidos e Responsáveis

[Descreva, componentes que serão desenvolvidos no escopo geral do projeto]

<Nome da Inst. Parceira>	Versão: <1.0>
<Colocar Nome do Componente>	Data: 6/25/2014
Especificação de Produto	

3.1 Componentes

Componente	Descrição	Responsável
<i>[Informe component e]</i>	<i>[Faça uma breve descrição dos do componente]</i>	<i>[Informe o grupo, setor ou organização responsável pelo desenvolvimento do componente]</i>

Obs: Deve-se incluir linhas na tabela acima quantos forem os componentes.

3.2 Ambiente

[Detalhe o ambiente onde o produto (ou os componentes) serão submetidos]

3.3 Ambiente de Desenvolvimento

[Detalhe o ambiente de desenvolvimento, bem como produtos e ferramentas a serem utilizadas]

3.4 Ambiente de Uso

[Identifique e detalhe todos os itens que devem ser considerados]

- *Tarefas a serem realizadas?*
- *Tempo de utilização?*
- *Restrições ambientais?*
- *Entre outros...*

3.5 Descrição do componente a ser produzido.

[Descreva cada componente que será produzido (componentes e subcomponentes)]

3.5.1 <Nome do Componente>

Descrição	<i>[Descrição do componente a ser desenvolvido]</i>
Tipo	<i>[Qualifique o componente]</i>
Tarefas a serem realizadas pelo componente	<i>[Descreva as funções que serão desenvolvidas para o componente realizar]</i>
Interfaces	<i>[Descrição das interfaces com outros componentes ou ambiente realizadas com o componente]</i>
Assistência	<i>[Descrever possíveis assistências de outros componentes para a realização das tarefas]</i>
Envolvimento	<i>[Qual é o grau de comprometimento do componente com o todo.]</i>
Obsevações	<i>[Observações relevantes a ser registrada.]</i>

Obs: Deve-se incluir uma tabela para cada componente.

4. Recursos do Produto

[Liste e descreva brevemente os recursos do produto]

5. Restrições

[Descreve quaisquer restrições internas, restrições externas ou outras dependências]

<Nome da Inst. Parceira>	Versão: <1.0>
<Colocar Nome do Componente>	Data: 6/25/2014
Especificação de Produto	

6. Faixas de Qualidade

[Defina as faixas de qualidade para desempenho, robustez, tolerância a erros, usabilidade e características semelhantes que não são capturadas no Conjunto de Recursos]

7. Precedência e Prioridade

[Defina a prioridade dos diferentes recursos do sistema]

8. Outros Requisitos do Produto

[Liste os padrões aplicáveis, os requisitos de hardware ou de plataforma, os requisitos de desempenho e os requisitos de ambiente]

8.1 Padrões Aplicáveis

[Liste todos os padrões e normas com os quais o produto deverá estar em conformidade. Entre eles, poderão estar incluídos padrões legais e reguladores (Normas espaciais, etc.) e padrões de qualidade e de segurança (ISO, CMMI)]

8.2 Requisitos do Sistema

[Defina todos os requisitos do sistema necessários para suportar o componente. Entre eles, poderão estar incluídos as plataformas de rede e os sistemas de operacionais de host suportados, configurações, memória, periféricos e componentes externos]

8.3 Requisitos de Desempenho

[Use esta seção para detalhar os requisitos de desempenho. Os problemas de desempenho podem abranger itens como fatores de carga, capacidade de comunicação, taxa de transferência, precisão e confiabilidade ou tempos de resposta em uma série de condições de carregamento]

8.4 Requisitos Ambientais

[Detalhe os requisitos ambientais, conforme necessário. Para sistemas baseados em hardware, as questões ambientais poderão incluir temperatura, choques, umidade, radiação etc. Para aplicativos de software, os fatores ambientais podem incluir condições de uso, ambiente da execução, disponibilidade de recursos, problemas de manutenção, e recuperação e tratamento de erros]

9. Requisitos da Documentação

[Esta seção descreve a documentação que deverá ser desenvolvida para suportar a implantação bem-sucedida bem como documentos legais exigidos por normas ou modelos para a sua operação]

9.1 Documentos legais

[Descreva os documentos, a finalidade e o conteúdo de cada documento a ser produzido, bem como em qual fase ele será produzido e itens de controle de configuração, quando existir]

9.2 Outros documentos

[Descreva os documentos, a finalidade e o conteúdo de cada documento a ser produzido, bem como em qual fase ele será produzido e itens de controle de configuração, quando existir]

Apêndice E – Especificação do componente

<Nome da Inst. Parceira>

<Nome do Projeto>
Especificação de Componente
<Nome do Componente>
Versão **<1.0>**

<Nome do Autor>

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

Índice Analítico

1.	Introdução	5
1.1	Finalidade	5
1.2	Escopo	5
1.3	Definições, Acrônimos e Abreviações	5
1.4	Referências	5
1.5	Visão Geral	5
2.	Especificação Geral do Componente.	5
2.1	Descrição.	5
2.1.1	Breve Descrição	5
2.1.2	Representação Gráfica do Componente	5
2.2	Metas	5
2.3	Métricas	6
2.3.1	<nome da medida de desempenho1>	6
2.3.2	<nome da medida de desempenho2>	6
2.4	Fluxo de Trabalho	6
2.4.1	Fluxo de Trabalho Básico	6
	<nome do passo do fluxo de trabalho>	6
2.4.2	Fluxos de Trabalho Alternativos	6
	<nome do passo do fluxo de trabalho>	6
2.5	Categoria	6
2.6	Risco	6
2.7	Possibilidades	6
2.8	Proprietário do Processo	6
2.9	Requisitos Especiais	6
2.9.1	<Nome do Requisito Especial 1>	6
2.9.2	<Nome do Requisito Especial 2>	6
2.10	Pontos de Extensão	6
2.10.1	<Nome do Ponto de Extensão1>	6
2.10.2	<Nome do Ponto de Extensão2>	6
2.11	Comportamento	7
2.12	Usabilidade	7
2.13	Confiabilidade	7
2.14	Desempenho.	7
2.15	Problemas de Escala	7
3.	Distribuição das Funcionalidades (Modelagem).	7
3.1	Hardware.	7
3.1.1	Descrição Completa.	7
3.1.2	Representação Gráfica do Hardware.	7
3.1.3	Modelo do Hardware.	7
3.1.4	Descrição das ferramentas utilizadas.	8
3.1.5	Descrição detalhada do Hardware.	8
3.1.6	<Nome do artefato1>	8
3.1.7	Descrição das interfaces do Hardware	8
3.1.8	Restrições de Hardware	8
3.2	Software.	8
3.2.1	Descrição Completa	8
3.2.2	Representação Gráfica do Software	8
3.2.3	Modelo do Software	8

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

3.2.4	Descrição das ferramentas utilizadas.	8
3.2.5	Descrição detalhada do Software	8
3.2.6	<Nome da funcionalidade1>	8
3.2.7	Descrição das interfaces do Software	8
3.2.8	Restrições de Software	8
3.3	Integração.	9
3.3.1	Descrição Completa da Integração	9
3.3.2	Ambiente de Integração	9
4.	Proposta de Solução.	9
4.1	Solução NNN , Nome	9
4.1.1	Parte Software.	9
4.1.2	Parte Hardware.	9
4.1.3	Protótipo.	9
5.	Solução Adotada.	9
6.	Aquisição	9
6.1	Identificação Fornecedor	9
6.2	Critérios de Seleção	9
6.3	Organização contratual com o fornecedor	9

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

<Nome do Componente>

1. Introdução

[A introdução da Especificação Componentes fornece uma visão geral do documento inteiro. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral desta Especificação]

1.1 Finalidade

*[Especifique a finalidade desta **Especificação de Componentes**]*

1.2 Escopo

*[Uma breve descrição do escopo desta **Especificação de Componentes**, ela e de tudo o que for afetado ou influenciado por este documento]*

1.3 Definições, Acrônimos e Abreviações

*[Esta subseção fornece as definições de todos os termos, acrônimos e abreviações necessárias para interpretar corretamente a **Especificação de Componentes**. Em caso de existir um glossário de projeto, indicar nesta seção]*

1.4 Referências

*[Esta subseção fornece uma lista completa de todos os documentos mencionados em outra parte da **Especificação de Componentes**. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento]*

1.5 Visão Geral

*[Esta subseção descreve o que o restante da **Especificação de Componentes** contém e explica como o documento está organizado]*

2. Especificação Geral do Componente.

2.1 Descrição.

2.1.1 Breve Descrição

[A descrição aborda brevemente o papel e a finalidade do componente]

[Descrever os fatores gerais que afetam o componente e seus requisitos como:

- *perspectiva do produto*
- *funções do produto*
- *restrições*
- *suposições e dependências*
- *conjuntos de requisitos]*

2.1.2 Representação Gráfica do Componente

[Utilizar diagramas, modelos entre outros recursos que representem claramente o componente, suas funções, os subcomponentes. A representação deve permitir a visão independente do hardware e do software do componente, bem como as funcionalidades que serão realizadas por eles]

2.2 Metas

[Uma especificação das metas ou dos objetivos mensuráveis do componente]

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

2.3 Métricas

[Uma especificação das métricas relevantes ao componente e uma definição das metas que você deseja atingir usando essas métricas]

2.3.1 <nome da medida de desempenho1>

[Uma breve descrição da medida de desempenho]

2.3.2 <nome da medida de desempenho2>

[Uma breve descrição da medida de desempenho]

2.4 Fluxo de Trabalho

[Uma descrição textual do fluxo de trabalho representado pelo componente]

2.4.1 Fluxo de Trabalho Básico

<nome do passo do fluxo de trabalho>

[Uma breve descrição do passo do fluxo de trabalho]

2.4.2 Fluxos de Trabalho Alternativos

<nome do passo do fluxo de trabalho>

[Uma breve descrição do passo do fluxo de trabalho]

2.5 Categoria

[Indica o nível de responsabilidade das funções realizadas pelo componente em seu ambiente, se o 'central', 'de suporte', 'de gerenciamento' ou 'outras']

2.6 Risco

[Especifica os riscos de funcionamento e implementar do Componente]

2.7 Possibilidades

[Descreve o potencial de aprimoramento estimado ao componente]

2.8 Proprietário do Processo

[Define quem é o proprietário dos resultados alcançados]

2.9 Requisitos Especiais

[Descrição dos requisitos especiais do componente]

2.9.1 <Nome do Requisito Especial 1>

[Uma breve descrição do requisito especial]

2.9.2 <Nome do Requisito Especial 2>

[Uma breve descrição do requisito especial]

2.10 Pontos de Extensão

[Pontos de extensão ao requisito]

2.10.1 <Nome do Ponto de Extensão1>

[Uma definição do local do ponto de extensão no fluxo]

2.10.2 <Nome do Ponto de Extensão2>

[Uma definição do local do ponto de extensão no fluxo]

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

2.11 Comportamento

[Esta seção expressa os objetivos gerais do comportamento do produto que não são específicos de um determinado componente]

2.12 Usabilidade

[Esta seção inclui todos os requisitos que afetam a usabilidade]

2.13 Confiabilidade

[Os requisitos de confiabilidade (do ponto de vista do produto) devem ser especificados aqui. Abaixo, algumas sugestões:

- *Disponibilidade — especifique o percentual de tempo disponível (xx.xx%), horas antecipadas de uso, etc.*
- *Exatidão — especifique a precisão (resolução) e a exatidão (por algum padrão conhecido) que são necessárias na saída]*

2.14 Desempenho.

[As características de desempenho devem ser resumidas nesta seção. Inclua tempos de resposta específicos. Quando aplicável, faça referência a nomes dos componentes relacionados.

- *Tempo de resposta de uma transação (médio, máximo)*
- *Taxa de transferência;*
- *Capacidade;*
- *Uso de recursos]*

2.15 Problemas de Escala

[Liste e descreva em poucas palavras qualquer informação sobre as limitações ou precauções que precisam ser consideradas]

3. Distribuição das Funcionalidades (Modelagem).

[E a atividade de construção de modelos que explique as características e o comportamento da parte software e da parte hardware do componente a ser desenvolvido. Em sua construção os modelos devem ser usados na elaboração e representação da arquitetura das características e funcionalidades que o componente deve prover e contribuir para o seu planejamento e na sua construção. Em geral implica na construção de modelos gráficos que simbolizam os artefatos dos componentes de software e hardware utilizados e os seus inter-relacionamentos. A escolha da técnica e da ferramenta a ser utilizada, deve ser escolhida de acordo com as características do componente a ser desenvolvido com as características do ambiente de desenvolvimento e dos envolvidos. Pode-se utilizar representação através de fluxogramas, linguagens gráficas, ferramentas e técnicas direcionadas para o componente a ser produzido.]

3.1 Hardware.

3.1.1 Descrição Completa.

[A descrição deve abordar toda as ações a serem resolvidas pela parte hardware do componente]

3.1.2 Representação Gráfica do Hardware.

[Representar preferencialmente através de modelos e diagramas o hardware e toda a sua funcionalidade, incluindo interfaces com o software e outros componentes]

3.1.3 Modelo do Hardware.

[Protótipo/ modelo de hardware com todos os artefatos utilizados]

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

3.1.4 *Descrição das ferramentas utilizadas.*

[Descrição das ferramentas a serem utilizadas para as atividades de modelagem e prototipação do componente.]

3.1.5 *Descrição detalhada do Hardware.*

[Descrição de todos os itens (artefatos) utilizados para a produção do hardware e a função de cada item no componente]

3.1.6 <Nome do artefato1>

[Descrição do artefato]

3.1.6.1 *Descrição detalhada da Artefato1*

[Descrição de todos os itens (artefatos) utilizados para a produção do hardware, bem como a função de cada item no componente, incluir quantas linhas forem necessárias]

3.1.7 *Descrição das interfaces do Hardware*

[Descrição detalhada das interfaces que o hardware faz].

3.1.8 *Restrições de Hardware*

[Descrição detalhada das restrições do hardware, bem como consequências]

3.2 Software.

3.2.1 *Descrição Completa*

[A descrição deve abordar toda as ações a serem resolvidas pela parte software do componente]

3.2.2 *Representação Gráfica do Software*

[Representar preferencialmente através de modelos e diagramas o software e toda a sua funcionalidade, incluindo interfaces com o hardware e outros componentes]

3.2.3 *Modelo do Software*

[Protótipo/ modelo de software com todos os artefatos utilizados]

3.2.4 *Descrição das ferramentas utilizadas.*

[Descrição das ferramentas a serem utilizadas para as atividades de modelagem e prototipação do componente.]

3.2.5 *Descrição detalhada do Software*

[Descrição de todos os itens (artefatos) utilizados para a produção do software e a função de cada item no componente]

3.2.6 <Nome da funcionalidade1>

[Descrição da funcionalidade]

3.2.6.1 *Descrição detalhada da Funcionalidade1*

[Descrição de como deve ser implementada a funcionalidade]

3.2.7 *Descrição das interfaces do Software*

[Descrição detalhada das interfaces que o software faz]

3.2.8 *Restrições de Software*

[Descrição detalhada das restrições do software, bem como consequências]

<Nome da Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <26/10/14>
Especificação do Componente	

3.3 Integração.

3.3.1 Descrição Completa da Integração

[A descrição deve abordar toda as ações a serem realizadas para a perfeita integração das partes do componente]

3.3.2 Ambiente de Integração

[Descrever o ambiente necessário para a integração das partes do componente]

4. Proposta de Solução.

[Deve-se descrever a proposta de solução do componente bem como o resultado da avaliação do protótipo, Para os componentes podem ser desenvolvidas diferentes soluções e para cada solução apresentada deve ser incluído mais um subitem no item 4.(4.1, 4.2, 4.3...)]

4.1 Solução NNN , Nome

[Descrição da solução do componente. Cada solução criar um novo item onde NNN deve ser uma numeração equencia para a solução e o Nome deve ser uma atribuição para identificação da solução].

4.1.1 Parte Software.

[Descrição da parte de software da solução do componente].

4.1.2 Parte Hardware.

[Descrição da parte de hardware da solução do componente]

4.1.3 Protótipo.

[Protótipo do componente, HW, SW e completo].

5. Solução Adotada.

[Deve-se descrever a proposta de solução do componente bem como o resultado da avaliação do protótipo, Para os componentes podem ser desenvolvidas diferentes soluções e para cada solução apresentada deve ser incluído mais um subitem no item 4.(4.1, 4.2, 4.3...)]

6. Aquisição

[Descreva a abordagem adotada para aquisição]

6.1 Identificação Fornecedor

[Descrição do Fornecedor do componente]

6.2 Critérios de Seleção

[Descreva os itens que foram utilizados para a seleção do fornecedor]

6.3 Organização contratual com o fornecedor

[Indicação e anexo do contrato de fornecimento]

Apêndice F – Projeto de teste

<Nome Inst. Parceira>

<Nome do Projeto>
<Nome do Componente>
Projeto de Teste
Versão <1.0>

<Nome do Autor>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Índice Analítico

1.	Introdução	5
1.1	Finalidade	5
1.2	Escopo	5
1.3	Alvo do Teste	5
1.4	Terminologia e Acrônimos do Documento	5
1.5	Referências	5
1.6	Estrutura do Documento	5
2.	Missão de Avaliação e Motivação dos Testes	5
2.1	Informações Detalhadas	5
2.2	Missão de Avaliação	6
2.3	Motivadores dos Testes	6
3.	Itens-alvo dos Testes	6
4.	Resumo dos Testes Planejados	6
4.1	Resumo das Inclusões dos Testes	6
4.2	Resumo das Inclusões dos Testes	6
5.	Abordagem dos Testes	8
5.1	Técnicas e Tipos de Teste	8
5.1.1	Teste de Integridade de Dados e de Banco de Dados	8
5.1.2	Teste de Função	9
5.1.3	Teste da Interface do Usuário	10
5.1.4	Teste de Carga	11
5.1.5	Teste de Stress	12
5.1.6	Teste de Volume	14
5.1.7	Teste de Segurança	16
5.1.8	Teste de Tolerância a Falhas e de Recuperação	17
5.1.9	Teste de Configuração	19
5.1.10	Teste de Instalação	20
6.	Critérios de Entrada e de Saída	21
6.1	Plano de Teste	21
6.1.1	Critérios de Entrada de Plano de Teste	21
6.1.2	Critérios de Saída de Plano de Teste	21
6.1.3	Critérios de Suspensão e de Reinício	22
7.	Produtos Liberados	22
7.1	Sumários de Avaliação de Testes	22
7.2	Relatórios da Cobertura de Teste	22
7.3	Relatórios da Qualidade Perceptível	22
7.4	Registros de Incidentes e Solicitações de Mudança	22
7.5	Conjunto de Testes de Regressão e Scripts de Teste de Suporte	22
7.6	Produtos de Trabalho Adicionais	22
7.6.1	Resultados Detalhados dos Testes	22
7.6.2	Scripts de Teste Funcionais Automatizados Adicionais	22
7.6.3	Guia de Teste	22
8.	Fluxo de Trabalho de Teste	23
9.	Necessidades Ambientais	23

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

9.1	Hardware Básico do Componente	23
10.	Responsabilidades, Perfil da Equipe e Necessidades de Treinamento	23
10.1	Pessoas e Papéis	23
10.2	Perfil da Equipe e Necessidades de Treinamento	23
11.	Marcos do teste	24
12.	Procedimentos e Processos de Gerenciamento	24
12.1	Medição e Avaliação da Extensão do Teste	24
12.2	Avaliação dos Produtos Liberados deste Plano de Teste	24
12.3	Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções	24
12.4	Estratégias de Rastreabilidade	24
12.5	Aprovação e Encerramento	24

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

<Plano de Teste>

1. Introdução

1.1 Finalidade

[A finalidade do Plano de Teste é reunir todas as informações necessárias ao planejamento e ao controle do esforço de teste. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.]

Este Plano de Teste referente ao Processo de Desenvolvimento de Software suporta os seguintes objetivos:

- *[Identifica os itens que devem ser inspecionados pelos testes.]*
- *Identifica a motivação e as ideias subjacentes às áreas de teste a serem abrangidas.*
- *Descreve a abordagem de teste que será usada.*
- *Identifica os recursos necessários e fornece uma estimativa dos esforços de teste.*
- *Lista os elementos liberados do projeto de teste]*

1.2 Escopo

[Descreva os níveis de teste por exemplo, Unidade, Integração ou Sistema — e os tipos de teste — como Funcionalidade, Usabilidade, Confiabilidade, Desempenho e Suportabilidade — que serão abordados por este Plano de Teste. Também é importante fornecer uma indicação geral das áreas importantes que serão excluídas do escopo]

1.3 Alvo do Teste

[Forneça uma breve descrição do alvo para o qual o Plano de Teste está sendo escrito. Isso ajudará os leitores do documento a identificarem se ele realmente está destinado ao seu uso e também ajudará a evitar que o documento seja usado de forma inadequada.]

Observação: Frequentemente, o estilo e o conteúdo do documento são alterados em função do alvo]

1.4 Terminologia e Acrônimos do Documento

[Esta subseção fornece as definições de todos os termos, acrônimos e abreviações necessárias à adequada interpretação do Plano de Teste. Caso o projeto tenha um glossário inclua uma referência ao Glossário do projeto na seção Referências]

1.5 Referências

[Esta subseção fornece uma lista dos documentos mencionados em qualquer outra parte do Plano de Teste. Identifique cada documento por título, número da versão (ou do relatório, se aplicável), data, organização de publicação ou autor original. Especifique as fontes a partir das quais as “versões oficiais” das referências podem ser obtidas]

1.6 Estrutura do Documento

[Esta subseção descreve o que o restante do Plano de Teste contém e fornece uma introdução de como o restante do documento está organizado]

2. Missão de Avaliação e Motivação dos Testes

[Forneça uma visão geral da missão e da motivação dos testes que serão conduzidos]

2.1 Informações Detalhadas

[Forneça uma breve descrição dos fundamentos que justificam o esforço de teste definido neste Plano de Teste. Inclua informações como, por exemplo, o problema principal que está sendo resolvido, os principais benefícios da solução, a arquitetura planejada da solução e um breve histórico do projeto. Quando essas informações estiverem definidas em outros documentos, você poderá incluir referências a esses documentos mais detalhados caso seja apropriado. Esta seção só deverá conter de três a cinco parágrafos]

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

2.2 Missão de Avaliação

[Forneça uma breve sentença que defina a missão do esforço de avaliação atual. Essa sentença poderá incorporar uma ou mais preocupações incluindo:

- *localizar o maior número de erros possível*
- *localizar problemas importantes e avaliar os riscos da qualidade perceptível*
- *advertir sobre os riscos de projeto percebidos*
- *certificar um padrão*
- *verificar uma especificação (requisitos, design ou alegações)*
- *advertir sobre a qualidade do produto e satisfazer os envolvidos*
- *advertir sobre os testes*
- *cumprir as determinações do processo*
- *e assim por diante*

Cada missão fornece um contexto diferente para o esforço de teste e altera a maneira como o teste deverá ser abordado]

2.3 Motivadores dos Testes

[Forneça um resumo dos principais elementos que motivarão o esforço de teste. Os testes poderão ser motivados por uma série de fatores como, por exemplo, riscos de qualidade, riscos técnicos, riscos do projeto, casos de uso, requisitos funcionais, requisitos não funcionais, elementos de design, falhas ou erros suspeitos, solicitações de mudança, etc.]

3. Itens-alvo dos Testes

A listagem abaixo identifica os itens — software, hardware e elementos de suporte do produto — que foram identificados como alvo dos testes. Essa lista representa os itens que serão testados.

[Forneça uma lista de nível superior dos principais itens que estarão sujeitos a testes. Essa lista deve incluir itens produzidos diretamente pela equipe de desenvolvimento do projeto e itens de que dependem esses produtos; por exemplo, o hardware de processamento básico, dispositivos periféricos, sistemas operacionais, produtos ou componentes de terceiros, etc. É recomendável agrupar a lista por categoria e atribuir importância relativa a cada motivador]

4. Resumo dos Testes Planejados

[Esta seção fornece um resumo de nível superior dos testes que serão executados. O resumo fornecido aqui representa uma visão geral de nível superior dos testes que serão e dos que não serão executados]

4.1 Resumo das Inclusões dos Testes

*[Forneça um resumo de nível superior dos principais testes planejados atual. Observe o que será incluído no plano e registre o que **não** será incluído explicitamente na seção denominada Resumo das Inclusões dos Testes]*

4.2 Resumo das Exclusões dos Testes

*[Forneça um resumo de nível superior dos possíveis testes que poderiam ter sido conduzidos, mas que foram **explicitamente excluídos** deste plano. Se você não for implementar ou executar um tipo de teste, informe claramente que o teste não será executado ou implementado e justifique. A seguir, há exemplos de justificativas que poderão ser usadas:*

- *“Esses testes não contribuem para alcançar a missão de avaliação”*
- *“Não há recursos suficientes para executar esses testes”*

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

- *“Esses testes são desnecessários devido aos testes executados por xxx”*

Segundo um prisma heurístico, se você achar que é perfeitamente concebível que um dos membros de seu público espere que um determinado aspecto de teste seja incluído e se você não pretender ou não puder incluí-lo, justifique sua exclusão. Se a equipe concordar que a exclusão é óbvia, você provavelmente não precisará listá-la]

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

5. Abordagem dos Testes

[Esta seção apresenta a estratégia recomendada para criar e implementar os testes necessários]

5.1 Técnicas e Tipos de Teste

5.1.1 Teste de Integridade de Dados e de Banco de Dados

[Os bancos de dados e os processos de banco de dados quando houver deverão ser testados como um subsistema independente. Esse teste deve testar os subsistemas sem que a Interface do Usuário do objetivo do teste faça interface com os dados. Identifique as ferramentas e técnicas que poderão existir para suportar os testes identificados na tabela a seguir]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<i>[Experimentar processos e métodos de acesso a banco de dados independentes da UI para que você possa observar e registrar comportamentos-alvo incorretos ou a existência de dados corrompidos]</i>
Técnica:	<ul style="list-style-type: none"> • <i>[Dispare cada processo e método de acesso a banco de dados, propagando solicitações de dados ou dados válidos e inválidos em cada um deles.</i> • <i>Inspeccione o banco de dados para assegurar que os dados foram distribuídos conforme o planejado e que todos os eventos de banco de dados ocorreram de forma adequada, ou revise os dados retornados para assegurar que os dados corretos foram recuperados pelas razões corretas]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>restaurador e reproduzidor de imagem da configuração básica</i> • <i>ferramentas de backup e de recuperação</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas e utilitários SQL de banco de dados</i> • <i>ferramentas de geração de dados]</i>
Critérios de Êxito:	<i>[A técnica suporta o teste de todos os principais processos e métodos de acesso a banco de dados]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Considerações Especiais:	<ul style="list-style-type: none"> • <i>[Os testes poderão exigir drivers ou um ambiente de desenvolvimento.</i> • <i>Os processos deverão ser disparados manualmente.</i> • <i>Deverão ser usados bancos de dados pequenos ou de tamanho mínimo (com um número limitado de registros) para aumentar a visibilidade de quaisquer eventos não aceitáveis]</i>
--------------------------	---

5.1.2 Teste de Função

[O teste de função do objetivo do teste deve concentrar-se em todos os requisitos de teste que possam ser diretamente associados as funções e regras de negócios. A meta desse teste é verificar a adequada aceitação, o processamento e a recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta; ou seja, verificar o aplicativo e seus processos internos interagindo com o aplicativo através da Interface Gráfica do Usuário (GUI) e analisar a saída ou os resultados. A tabela a seguir identifica um resumo do teste recomendado para cada aplicativo]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<i>[Experimentar a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo]</i>
Técnica:	<p><i>[Execute os recursos e os fluxos ou as funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se:</i></p> <ul style="list-style-type: none"> • <i>os resultados esperados ocorrerão quando forem usados dados válidos</i> • <i>as mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos</i> • <i>cada regra de negócio será aplicada de forma adequada]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>restaurador e reproduzidor de imagem da configuração básica</i> • <i>ferramentas de backup e de recuperação</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas de geração de dados]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Critérios de Êxito:	<p><i>[A técnica suporta o teste de:</i></p> <ul style="list-style-type: none"> • <i> todos os principais cenários de caso de uso</i> • <i> todos os principais recursos]</i>
Considerações Especiais:	<p><i>[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste de funcionamento]</i></p>

5.1.3 Teste da Interface do Usuário

[O Teste da Interface do Usuário (UI) verifica a interação do usuário com o software. A meta do teste de UI é assegurar que a UI forneça ao usuário o acesso e a navegação adequados através das funções do objetivo do teste. Além disso, o teste de UI assegura que os objetos contidos na UI funcionem conforme o esperado e estejam em conformidade com padrões corporativos ou da indústria]

Parte do Componente:	<p><i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i></p>
Objetivo da Técnica:	<p><i>[Experimentar o seguinte para observar e registrar a conformidade com padrões e o comportamento-alvo:</i></p> <ul style="list-style-type: none"> • <i>A navegação pelo objetivo do teste para verificar se reflete os requisitos e as funções de negócios, incluindo a navegação janela a janela e campo a campo, e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras).</i> • <i>Os objetos e as características poderão ser experimentados como, por exemplo, tamanho, posição, estado e foco]</i>
Técnica:	<p><i>[Crie ou modifique testes para cada função e verificar e os estados de objeto apropriados]</i></p>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica necessita da Ferramenta de Automação de Scripts de Teste.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Critérios de Êxito:	<p><i>[A técnica suporta o teste de cada tela ou janela principal que será muito usada pelo usuário final]</i></p>
Considerações Especiais:	<p><i>[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste]</i></p>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

5.1.4 Teste de Carga

[O teste de carga é um teste de desempenho que revela o objetivo do teste a diferentes cargas de trabalho para medir e avaliar as habilidades e os comportamentos de desempenho dele, a fim de verificar se este continua a funcionar adequadamente com essas diferentes cargas de trabalho]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<i>[Experimentar funções ou transações designadas em várias condições de carga de trabalho, a fim de observar e registrar o comportamento-alvo e os dados de desempenho do component.]</i>
Técnica:	<ul style="list-style-type: none"> • <i>[Use os Scripts de Teste de Transação desenvolvidos para os Testes de Ciclos de Negócios ou de Funcionamento como uma base, mas lembre-se de remover as iterações e os atrasos desnecessários.</i> • <i>Modifique os arquivos de dados a fim de aumentar o número de transações ou modifique os testes a fim de aumentar o número de vezes que cada transação ocorre.</i> • <i>As Cargas de Trabalho devem incluir Cargas de Pico (por exemplo, Diárias, Semanais, Mensais, etc.).</i> • <i>As Cargas de Trabalho devem representar cargas Médias assim como cargas de Pico.</i> • <i>As Cargas de Trabalho devem representar Picos Instantâneos e Picos Sustentados.</i> • <i>As Cargas de Trabalho devem ser executadas com diferentes Configurações do Ambiente de Teste]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>ferramenta de controle e de Programação de Carga de Transações</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas de restrição de recursos (por exemplo, Enlatados)</i> • <i>ferramentas de geração de dados]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Critérios de Êxito:	<i>[A técnica suporta o teste de Emulação de Carga de Trabalho, que é a emulação bem-sucedida da carga de trabalho sem que nenhuma falha ocorra devido a problemas de implementação do teste]</i>
Considerações Especiais:	<i>•[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste]</i>

5.1.5 Teste de Stress

[O teste de stress é um tipo de teste de desempenho implementado e executado para se compreender como ocorrem falhas no componente devido a condições que estão no limite ou fora do limite das tolerâncias esperadas. Normalmente, isso envolve poucos recursos ou a concorrência de recursos. As condições de poucos recursos revelam como ocorrem falhas no objetivo do teste que não estão aparentes em condições normais]

[Observação: As transações mencionadas na tabela a seguir são transações de negócios lógicas]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<i>[Experimentar as funções do objetivo do teste nas seguintes condições de stress a fim de observar e registrar o comportamento-alvo que identifica e documenta as condições que fazem com que o componente deixe de funcionar adequadamente:</i> <ul style="list-style-type: none"> <i>• pouca ou nenhuma memória disponível no servidor (memória RAM e espaço de armazenamento persistente)</i> <i>• número máximo real ou fisicamente capaz de clientes conectados ou simulados</i> <i>• vários usuários executando as mesmas transações nos mesmos dados ou contas</i> <i>• conjunto ou volume de transações que geram “sobrecarga” (consulte Determinação do Perfil de Desempenho acima)]</i>
Técnica:	<ul style="list-style-type: none"> <i>• [Use os testes de Carga ou de Determinação do Perfil de Desempenho.</i> <i>• Para testar recursos limitados, os testes deverão ser executados em uma única máquina, e a memória RAM e o espaço de armazenamento persistente no servidor deverão ser reduzidos ou limitados.</i> <i>• Para os testes de stress restantes, deverão ser usados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto ou o volume de transações no pior caso]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>ferramenta de controle e de Programação de Carga de Transações</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas de restrição de recursos (por exemplo, Enlatados)</i> • <i>ferramentas de geração de dados]</i>
Critérios de Êxito:	<p><i>[A técnica suporta o teste de Emulação de Stress. O componente poderá ser emulado, de maneira eficaz, em uma ou mais condições definidas como condições de stress, e poderá ser capturada uma observação do estado resultante do componente durante e depois de a condição ter sido emulada]</i></p>
Considerações Especiais:	<ul style="list-style-type: none"> • <i>[Para forçar a rede, talvez seja necessário que ferramentas da rede a sobrecarreguem com mensagens ou pacotes.</i> • <i>O armazenamento persistente usado para o componente deverá ser reduzido temporariamente, a fim de restringir o espaço disponível para que o banco de dados se desenvolva.</i> • <i>Sincronize o acesso simultâneo dos clientes aos mesmos registros ou contas de dados]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

5.1.6 Teste de Volume

[O teste de volume revela o objetivo do teste a grandes volumes de dados a fim de determinar se serão atingidos limites que farão com que o software deixe de funcionar. Esse teste também identifica o volume ou a carga máxima contínua que o objetivo do teste pode suportar durante um determinado período de tempo. Por exemplo, se o objetivo do teste estiver processando um conjunto de registros de banco de dados para gerar um relatório, um Teste de Volume usará um grande banco de dados de testes e verificará se o comportamento do software foi normal e se gerou o relatório correto]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<p><i>[Experimentar o objetivo do teste nos seguintes cenários de elevado volume para observar e registrar o comportamento-alvo:</i></p> <ul style="list-style-type: none"> <i>• O número máximo (real ou fisicamente capaz) de clientes conectados, ou simulados, todos executando a mesma função de negócios (desempenho), no pior caso, durante um longo período de tempo.</i> <i>• Foi atingido o tamanho máximo do banco de dados (real ou em escala) e várias consultas ou transações de relatório são executadas simultaneamente]</i>
Técnica:	<ul style="list-style-type: none"> <i>• [Use os testes de Carga ou de Determinação do Perfil de Desempenho.</i> <i>• Deverão ser usados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto ou o volume de transações no pior caso (consulte Teste de Stress) durante um longo período de tempo.</i> <i>• Será criado o tamanho máximo do banco de dados (real, em escala ou preenchido com dados representativos) e serão usados vários clientes para executar consultas e transações de relatório simultaneamente durante longos períodos de tempo]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>ferramenta de controle e de Programação de Carga de Transações</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas de restrição de recursos (por exemplo, Enlatados)</i> • <i>ferramentas de geração de dados]</i>
CrITÉRIOS de Êxito:	<p><i>[A técnica suporta o teste de Emulação de Volume. É possível emular, de forma eficaz, grandes quantidades de usuários, dados, transações ou outros aspectos do sistema utilizados em volume e poderá ser capturada uma observação sobre as mudanças de estado do componente durante o teste de volume]</i></p>
Considerações Especiais:	<p><i>[Que período de tempo seria considerado aceitável para as condições de elevado volume, conforme observado acima?]</i></p>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

5.1.7 Teste de Segurança

Com base no nível de segurança desejado, assegura as funções]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<p><i>[Experimentar o objetivo do teste nas seguintes condições para observar e registrar o comportamento-alvo:</i></p> <ul style="list-style-type: none"> • <i>Segurança no nível do aplicativo: um ator poderá acessar somente as funções ou os dados para o quais seu tipo de usuário tenha recebido permissão.</i> • <i>Segurança no nível do componente: somente as funcionalidades do componente e aos aplicativos têm permissão para acessá-los]</i>
Técnica:	<ul style="list-style-type: none"> • <i>[Segurança no nível do aplicativo: identifique e liste cada tipo de usuário e as funções ou os dados para os quais cada um deles tem permissão de acesso]</i> <ul style="list-style-type: none"> ○ <i>Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário.</i> ○ <i>Modifique o tipo de usuário e executar novamente os testes para os mesmos usuários. Em cada caso, verifique se as funções ou dados adicionais estão corretamente disponíveis ou se têm seu acesso negado.</i> • <i>Acesso no nível do componente: [Consulte Considerações Especiais abaixo]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente.]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>Ferramentas de investigação e contra a violação da segurança por “hackers”</i> • <i>Ferramentas de Administração da Segurança do Sistema Operacional]</i>
CrITÉRIOS de Êxito:	<i>[A técnica suporta, para cada tipo de ator conhecido, o teste das funções apropriadas ou dos dados afetados pelas configurações de segurança]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Considerações Especiais:	<i>[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste]</i>
--------------------------	--

5.1.8 Teste de Tolerância a Falhas e de Recuperação

[O teste de tolerância a falhas e de recuperação assegura que o objetivo do teste possa tolerar e se recuperar, de maneira eficaz, de uma série de falhas de hardware, software ou de rede com perda indevida de dados ou da integridade dos dados]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<p><i>[Simular as condições de falha e experimentar os processos de recuperação (manuais e automatizados) para restaurar o estado conhecido e desejado do banco de dados, dos aplicativos e do componente. Os seguintes tipos de condições estão incluídos no teste para observar e registrar o comportamento-alvo após a recuperação:</i></p> <ul style="list-style-type: none"> <i>• interrupção da energia para o cliente</i> <i>• interrupção da energia para o servidor</i> <i>• interrupção da comunicação através dos servidores de rede</i> <i>• ciclos incompletos (processos de filtragem de dados interrompidos, processos de sincronização de dados interrompidos)</i> <i>• ponteiros ou chaves de banco de dados inválidas</i> <i>• elementos de dados inválidos ou corrompidos no banco de dados]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Técnica:	<p><i>[Os testes de Funcionamento e ser usados como uma base para criar uma série de transações para suportar os testes de tolerância a falhas e de recuperação e principalmente para definir os testes que serão executados para verificar se a recuperação teve êxito.</i></p> <ul style="list-style-type: none"> • <i>Interrupção da energia para o cliente: desligue o PC.</i> • <i>Interrupção da energia para o servidor: simule ou inicie procedimentos de desligamento do servidor.</i> • <i>Interrupção através de servidores de rede: simule ou inicie uma perda de comunicação com a rede (desconecte fisicamente os cabos de comunicação ou desligue os servidores ou os roteadores de rede).</i> <p><i>Depois que as condições acima ou as condições simuladas tiverem sido alcançadas, as transações adicionais deverão ser executadas e, quando o estado desse segundo ponto do teste for atingido, os procedimentos de recuperação deverão ser disparados.</i></p> <p><i>O teste de ciclos incompletos utiliza a mesma técnica descrita acima, exceto pelos processos de banco de dados propriamente ditos, que deverão ser anulados ou prematuramente encerrados.</i></p> <p><i>O teste das condições a seguir exige que seja atingido um estado conhecido do banco de dados.</i></p> <p><i>Vários campos, ponteiros e chaves de banco de dados deverão ser corrompidos manualmente e diretamente no banco de dados (através das ferramentas de banco de dados). As transações adicionais deverão ser executadas usando os Testes de Funcionamento do Aplicativo e deverão ser executados ciclos completos]</i></p>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente.]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>restaurador e reprodutor de imagem da configuração básica</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas de backup e de recuperação]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Critérios de Êxito:	<p><i>[A técnica suporta o teste de:</i></p> <ul style="list-style-type: none"> <i>• Um ou mais desastres simulados envolvendo uma ou mais combinações do aplicativo, banco de dados e componente.</i> <i>• Uma ou mais recuperações simuladas envolvendo uma ou mais combinações do aplicativo, banco de dados e componente em um estado conhecido desejado]</i>
Considerações Especiais:	<ul style="list-style-type: none"> <i>• [O teste de recuperação é altamente invasivo. Os procedimentos para desconectar cabos (simular perda de energia ou de comunicação) talvez não sejam desejáveis ou viáveis. Poderão ser necessários métodos alternativos como, por exemplo, ferramentas de software de diagnóstico.</i> <i>• Serão necessários Recursos dos Componentes (ou Operações de Computador), Bancos de Dados e Grupos de Redes.</i> <i>• Esses testes deverão ser executados após o expediente de trabalho ou em uma máquina isolada]</i>

5.1.9 Teste de Configuração

[O teste de configuração verifica o funcionamento do objetivo do teste em diferentes configurações de software e de hardware]

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<i>[Experimentar o objetivo do teste nas configurações de hardware e de software necessárias, a fim de observar e registrar o comportamento-alvo em diferentes configurações e identificar mudanças no estado da configuração]</i>
Técnica:	<ul style="list-style-type: none"> <i>• [Use Scripts de Teste de Funcionamento.</i> <i>• Abra e feche vários softwares relacionados que não sejam o objetivo do teste (por exemplo, os aplicativos Microsoft Excel e Word) como parte do teste ou antes do início do teste.</i> <i>• Execute as transações selecionadas para simular atores interagindo com softwares que sejam o objetivo do teste e com os que não sejam o objetivo do teste.</i> <i>• Repita o processo acima, minimizando a memória convencional disponível na estação de trabalho cliente]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>restaurador e reproduzidor de imagem da configuração básica</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)]</i>
CrITÉRIOS de Êxito:	<p><i>[A técnica suporta o teste de uma ou mais combinações dos itens-alvo do teste que são executadas em ambientes de implantação suportados e esperados]</i></p>
Considerações Especiais:	<ul style="list-style-type: none"> • <i>[Que software, que não seja o objetivo do teste, é necessário, está disponível e é acessível na área de trabalho?</i> • <i>Quais os aplicativos normalmente usados?</i> • <i>Que dados estão em execução nos aplicativos; por exemplo, uma grande planilha aberta no Excel ou um documento de 100 páginas no Word?]</i>

5.1.10 Teste de Instalação

[O teste de instalação tem duas finalidades]

Parte do Componente :	<p><i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i></p>
Objetivo da Técnica:	<p><i>[Experimentar a instalação do objetivo do teste em cada configuração de hardware exigida nas condições a seguir para observar e registrar o comportamento da instalação e as mudanças no estado da configuração:</i></p> <ul style="list-style-type: none"> • <i>nova instalação: uma nova máquina, em que nunca foi instalado anteriormente o Processo de Desenvolvimento de Software</i> • <i>atualização: uma máquina em que foi instalado anteriormente o Processo de Desenvolvimento de Software, na mesma versão</i> • <i>atualização: uma máquina em que foi instalado anteriormente o Processo de Desenvolvimento de Software, em uma versão mais antiga]</i>

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

Técnica:	<ul style="list-style-type: none"> • <i>[Desenvolva scripts automatizados ou manuais para validar a condição da máquina-alvo.</i> <ul style="list-style-type: none"> ○ <i>nova: nunca instalada</i> ○ <i>mesma versão ou versão mais antiga já instalada</i> • <i>Inicie ou execute a instalação.</i> • <i>Utilizando um subconjunto predeterminado de scripts de Teste de Funcionamento, execute as transações.]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente.]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>restaurador e reprodutor de imagem da configuração básica</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)]</i>
CrITÉRIOS de Êxito:	<p><i>[A técnica suporta o teste da instalação do produto desenvolvido em uma ou mais configurações de instalação.]</i></p>
Considerações Especiais:	<p><i>[Que transações de Processo de Desenvolvimento de Software devem ser selecionadas para constituir um teste que comprove que o aplicativo de Processo de Desenvolvimento de Software foi instalado com êxito e que não está faltando nenhum componente de software principal?]</i></p>

6. CritÉrios de Entrada e de Saída

6.1 Plano de Teste

[Especifique nesta seção o plano de teste da parte hardware e da parte software do componente, pode ser elaborado em conjunto ou criar uma nova subseção ex: 6.1 Plano de teste do hardware, 6.2 Plano de testes do Software, pode-se também criar uma subseção para o componente completo já integrado ex: 6.3. Plano de teste do componente. As subseções não necessitam ser iguais podendo ser atribuídos critério particulares em cada uma delas de acordo com as necessidades e características]

6.1.1 CritÉrios de Entrada de Plano de Teste

*[Especifique os critÉrios que serão usados para determinar se a execução do **Plano de Teste** poderá ser iniciada]*

6.1.2 CritÉrios de Saída de Plano de Teste

*[Especifique os critÉrios que serão usados para determinar se a execução do **Plano de Teste** foi concluída ou se a continuação da execução não será vantajosa]*

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

6.1.3 Critérios de Suspensão e de Reinício

[Especifique os critérios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos antes que o plano tenha sido totalmente executado. Especifique também segundo que critérios os testes poderão ser reiniciados]

7. Produtos Liberados

[Nesta seção, liste os vários artefatos que serão criados pelo esforço de teste e que serão produtos liberados úteis aos vários envolvidos do esforço de teste. Não liste todos os produtos do trabalho; liste apenas os que propiciam benefícios diretos tangíveis aos envolvidos e os que permitem medir o êxito do esforço de teste.

[Para esta seção levar considerar as orientações do item 6.1]

7.1 Sumários de Avaliação de Testes

[Forneça um breve resumo da forma e do conteúdo dos sumários de avaliação de testes e indique com que frequência eles serão gerados]

7.2 Relatórios da Cobertura de Teste

[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a extensão do teste e indique com que frequência eles serão gerados Forneça uma indicação referente ao método e às ferramentas usadas para registrar, medir e reportar a extensão do teste]

7.3 Relatórios da Qualidade Perceptível

[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a qualidade perceptível do produto e indique com que frequência eles serão gerados. Forneça uma indicação referente ao método e às ferramentas usadas para registrar, medir e reportar a qualidade perceptível do produto. Você poderá incluir análises dos Incidentes e Solicitações de Mudança ao longo da Cobertura de Teste]

7.4 Registros de Incidentes e Solicitações de Mudança

[Forneça um breve resumo do método e das ferramentas usadas para registrar, rastrear e gerenciar incidentes dos testes, as solicitações de mudança associadas e seus status]

7.5 Conjunto de Testes de Regressão e Scripts de Teste de Suporte

[Forneça um breve resumo dos recursos dos testes que serão distribuídos para permitir testes de regressão contínuos dos builds subsequentes do produto, a fim de ajudar a detectar as regressões na qualidade do produto]

7.6 Produtos de Trabalho Adicionais

*[Nesta seção, identifique os produtos de trabalho que são opcionais ou os que não deverão ser usados para medir ou avaliar a execução bem-sucedida do **Plano de Teste**]*

7.6.1 Resultados Detalhados dos Testes

[Trata-se de um conjunto de planilhas do Microsoft Excel relacionando os resultados determinados para cada caso de teste ou refere-se ao repositório dos registros de testes e dos resultados determinados mantidos por um produto de teste especializado]

7.6.2 Scripts de Teste Funcionais Automatizados Adicionais

[Estes scripts consistem em um conjunto de arquivos de código-fonte para scripts de teste automatizados ou no repositório do código-fonte e dos executáveis compilados para scripts de teste mantidos pelo produto de automação de testes]

7.6.3 Guia de Teste

[O Guia de Teste abrange um amplo conjunto de categorias, incluindo Catálogos de Idéias de Testes, Orientações de Práticas Adequadas, Padrões de Teste, Modelos de Erros e de Falhas, Padrões de Design de Automação, etc.]

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

8. Fluxo de Trabalho de Teste

*[Forneça um resumo do fluxo de trabalho a ser seguido pela equipe de Teste no desenvolvimento e na execução deste **Plano de Teste**. Para esta seção levar considerar as orientações do item 6.1]*

9. Necessidades Ambientais

*[Esta seção apresenta os recursos não humanos necessários ao **Plano de Teste** (Hardware e Software)]*

9.1 Hardware Básico do Componente

[Os conjuntos de tabelas a seguir apresentam os recursos do componente necessários ao esforço de teste descrito neste Plano de Teste.

Esta que esta seção deve ser preenchida ao logo do tempo conforme o necessário]

Recursos do Componente		
Recurso	Quantidade	Nome e Tipo

10. Responsabilidades, Perfil da Equipe e Necessidades de Treinamento

*[Esta seção apresenta os recursos necessários para abordar o esforço de teste descrito no **Plano de Teste** — as principais responsabilidades e os conjuntos de conhecimentos ou de habilidades exigidos desses recursos. (Hardware e Software)]*

10.1 Pessoas e Papéis

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

[Observação: Adicione ou exclua itens conforme o necessário.]

Recursos Humanos		
Papel	Recursos Mínimos Recomendáveis (número de papéis alocados em tempo integral)	Responsabilidades ou Comentários Específicos

10.2 Perfil da Equipe e Necessidades de Treinamento

Esta seção resume como abordar o perfil da equipe e o treinamento dos profissionais que ocuparão os papéis de teste no projeto.

[O modo como abordar o perfil da equipe e o treinamento dos profissionais varia de projeto para projeto. Se esta seção integrar um Plano de Teste Mestre, indique em que pontos do ciclo de vida do projeto serão necessárias diferentes habilidades e um número diferente de integrantes da equipe]

<Nome Inst. Parceira>	Version: <1.0>
<Nome do Componente>	Date: <25/06/2014
Projeto de Teste	

11. Marcos do teste

[Identifique os principais marcos da programação que definem o contexto do Esforço de Teste. (Hardware e Software)]

Marco	Data de Início Planejada	Data de Início Real	Data de Término Planejada	Data de Término Real
Término do teste				

12. Procedimentos e Processos de Gerenciamento

*[Resuma os processos e os procedimentos que deverão ser usados quando surgirem problemas no **Plano de Teste** e em sua execução (Hardware e Software)]*

12.1 Medição e Avaliação da Extensão do Teste

[Resuma o processo de medição e avaliação a ser usado para rastrear a extensão do teste]

12.2 Avaliação dos Produtos Liberados deste Plano de Teste

*[Resuma o processo de avaliação para revisar e aceitar os produtos liberados deste **Plano de Teste**]*

12.3 Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções

[Defina como os problemas referentes a processos serão relatados, como serão selecionadas pessoas para resolvê-los e o processo a ser seguido para se chegar a uma solução]

12.4 Estratégias de Rastreabilidade

[Refleta sobre estratégias de rastreabilidade adequadas referentes a:

- *Cobertura de Teste em relação às Especificações — possibilita a medição da extensão do teste*
- *Motivações de Teste — possibilitam a avaliação da relevância dos testes a fim de ajudar a determinar se eles deverão ser mantidos ou não*
- *Elementos de Design de Software — possibilitam o rastreamento das mudanças de design subsequentes que exigirão que os testes sejam executados novamente ou sejam cancelados*
- *Solicitações de Mudança Resultantes — fazem com que os testes que descobriram a necessidade da mudança sejam identificados e executados novamente para verificar se a solicitação de mudança foi efetuada com êxito]*

12.5 Aprovação e Encerramento

[Resuma o processo de aprovação e liste os cargos (e os nomes dos ocupantes atuais) que deverão aprovar inicialmente o plano e encerre com a execução satisfatória do plano]

Apêndice G – Homologação

<Nome da Inst. Parceira>

<Nome do Projeto>
<Nome do Componente>
Plano de Homologação
Versão <1.0>

<Nome do Autor>

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <2014-06-25>
Plano de Homologação	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <2014-06-25>
Plano de Homologação	

Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
1.4	Referências	4
1.5	Visão Geral	4
2.	Responsabilidades	4
3.	Tarefas de Aceitação do Produto	4
3.1	Critérios de Aceitação do Produto	4
3.2	Auditoria da Configuração Física	4
3.3	Auditoria Funcional da Configuração	4
3.4	Programação	4
4.	Requisitos de Recursos	5
4.1	Requisitos do Hardware	5
4.2	Requisitos do Software	5
4.3	Requisitos da Documentação	5
4.4	Requisitos de Pessoal	5
4.5	Requisitos de Dados de Teste	5
4.6	Outros Requisitos	5
5.	Resolução de Problemas e Ação Corretiva	5
6.	Ambiente de Aceitação do Produto	5
7.	Identificação de Avaliações de Artefato Necessárias	5
8.	Ferramentas, Técnicas e Metodologias	5
9.	Avaliação do Ciclo	5
9.1	Processo	5
9.2	Fases do Ciclo de vida	5
9.3	Planejamento do Projeto	5
9.4	Problemas e resoluções	5
9.5	Pessoas	6
9.6	Componente produzido	6

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <2014-06-25>
Plano de Homologação	

Plano de Homologação

1. Introdução

[A introdução do Plano de Homologação oferece uma visão geral de todo o documento. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e uma visão geral deste Plano de Homologação]

1.1 Finalidade

*[Especifique a finalidade deste **Plano de Homologação**]*

1.2 Escopo

*[Uma breve descrição do escopo deste **Plano de Homologação**; os Projetos aos quais ele está associado e tudo o que é afetado ou influenciado por este documento]*

1.3 Definições, Acrônimos e Abreviações

*[Esta subseção apresenta as definições de todos os termos, acrônimos e abreviações necessários para a correta interpretação do **Plano de Homologação**. Essas informações podem ser fornecidas mediante referência ao Glossário do projeto]*

1.4 Referências

*[Esta subseção apresenta uma lista completa de todos os documentos mencionados no **Plano de Homologação**. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento]*

1.5 Visão Geral

*[Esta subseção descreve o conteúdo restante do **Plano de Homologação** e explica como o documento está organizado]*

2. Responsabilidades

[Identifique explicitamente as responsabilidades do cliente e da equipe de desenvolvimento na preparação e realização das atividades de aceitação do produto]

3. Tarefas de Aceitação do Produto

3.1 Critérios de Aceitação do Produto

[Identifique os critérios objetivos para determinar a aceitabilidade dos artefatos de produtos liberados deste projeto. Esses critérios devem ser formalmente acordados entre o cliente e a equipe de desenvolvimento]

3.2 Auditoria da Configuração Física

[Identifique e liste os artefatos resultantes deste projeto que devem ser entregues e aceitos pelo cliente.]

3.3 Auditoria Funcional da Configuração

[Para cada artefato identificado na Auditoria de Configuração Física, identifique os métodos de avaliação e o nível de detalhes que serão usados para determinar se o artefato atende aos critérios de aceitação do produto. Os métodos podem abranger testes de execução de softwares, demonstrações do produto, revisões de documentação etc.]

3.4 Programação

[Um cronograma indicando o início e o término de cada tarefa de aceitação do produto, incluindo atividades de preparação e configuração]

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <2014-06-25>
Plano de Homologação	

4. Requisitos de Recursos

4.1 Requisitos do Hardware

[Por exemplo, itens de hardware, equipamentos de interface, itens de firmware]

4.2 Requisitos do Software

[Por exemplo, sistemas operacionais, compiladores, drivers de teste, geradores de dados de teste]

4.3 Requisitos da Documentação

[Por exemplo, documentação de teste, referências técnicas]

4.4 Requisitos de Pessoal

[Por exemplo, integrantes da equipe de desenvolvimento, representantes do cliente, autoridades de terceiros]

4.5 Requisitos de Dados de Teste

[Por exemplo, o tamanho, o tipo e a composição dos dados para suportar os testes de aceitação]

4.6 Outros Requisitos

[Por exemplo, equipamentos especiais]

5. Resolução de Problemas e Ação Corretiva

[Esta seção descreve os procedimentos para reportar e resolver os problemas identificados durante as atividades de Aceitação do Produto. Geralmente eles são abordados incluindo-se o artefato Plano de Resolução de Problemas através de referência]

6. Ambiente de Aceitação do Produto

[Descreva os planos para configurar o ambiente de aceitação do produto]

7. Identificação de Avaliações de Artefato Necessárias

[Com base na descrição da Auditoria Funcional da Configuração, identifique a avaliação de cada artefato individual que será realizada. Para cada artefato, liste o tipo de avaliação (teste, revisão etc.) e os objetivos]

Observação: Será preparado um artefato Caso de Teste para cada teste identificado aqui]

8. Ferramentas, Técnicas e Metodologias

[Uma lista de todas as ferramentas, técnicas e metodologias específicas que deverão ser usadas na execução das atividades de Aceitação do Produto]

9. Avaliação do Ciclo

[Análise e descrição do andamento de todas as fases e atividades do ciclo e sugestões de melhoria para outros ciclos.]

9.1 Processo

[Parecer relacionado ao processo de desenvolvimento]

9.2 Fases do Ciclo de vida

[Parecer relacionado as atividades que foram definidas em cada fase do ciclo de vida]

9.3 Planejamento do Projeto

[Parecer relacionado ao planejamento geral do projeto.]

9.4 Problemas e resoluções

[Parecer relacionados aos problemas que surgiram, agilidade de solução, qualidade das soluções, etc.]

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <2014-06-25>
Plano de Homologação	

9.5 Pessoas

[Parecer relacionados as pessoas envolvidas com o projeto, comprometimento, iniciativa, trabalho em grupo, necessidade de treinamento, etc.]

9.6 Componente produzido

[Parecer relacionados a qualidade do componente produzido, em atendimento aos requisitos, saídas ao longo do processo, etc.]

Apêndice H – Configuração

<Nome da Inst. Parceira>

<Nome do Projeto>

<Nome do Componente>

Gerenciamento da Configuração

Versão <1.0>

<Nome do Autor>

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <24/06/2014>
Gerenciamento da Configuração	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <24/06/2014>
Gerenciamento da Configuração	

Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
1.4	Referências	4
1.5	Visão Geral	4
2.	Gerenciamento de Configuração de Software	4
2.1	Organização, Responsabilidades e Interfaces	4
2.2	Controle de Configuração e Mudança	4
2.2.1	Processamento e Aprovação de Solicitações de Mudança	4
2.2.2	Responsável pelo Controle de Mudança	4
2.2.3	Mecanismo de armazenamento e Controle de Mudança.	4
2.2.4	Relatórios de Controle de Mudança.	4
2.3	Itens da Configuração	5
2.3.1	Ambiente de Desenvolvimento	5
2.3.2	Artefatos Produzidos.	5
2.3.3	Itens de Software.	5
2.3.4	Itens de Hardware.	5
3.	Controle de Subcontratados e Fornecedores	5
3.1.1	Itens de Software de Subcontratados.	5
3.1.2	Itens de Hardware de Subcontratados.	5
4.	Matriz de Rastreabilidade	5

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <24/06/2014>
Gerenciamento da Configuração	

Gerenciamento da Configuração

1. Introdução

*[A introdução do **Gerenciamento de Configuração** oferece uma visão geral de todo o documento. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e uma visão geral deste **Gerenciamento de Configuração**.]*

1.1 Finalidade

*[Especifique a finalidade do **Gerenciamento de Configuração**]*

1.2 Escopo

*[Uma breve descrição do escopo do **Gerenciamento de Configuração**; o modelo ao qual ele está associado e tudo o que é afetado ou influenciado por este documento]*

1.3 Definições, Acrônimos e Abreviações

*[Esta subseção apresenta as definições de todos os termos, acrônimos e abreviações necessários para a correta interpretação do **Gerenciamento de Configuração**. Essas informações podem ser fornecidas mediante referência ao Glossário do projeto]*

1.4 Referências

*[Esta subseção apresenta uma lista completa de todos os documentos mencionados no **Gerenciamento de Configuração**. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento]*

1.5 Visão Geral

*[Esta subseção descreve o conteúdo restante do **Gerenciamento de Configuração** e explica como o documento está organizado.*

Observar que o gerenciamento da configuração deve prover todas as fases do processo.]

2. Gerenciamento de Configuração de Software

2.1 Organização, Responsabilidades e Interfaces

[Descreva as atividades e quem será o responsável pela sua execução.

2.2 Controle de Configuração e Mudança

[Descreva em que pontos do ciclo de vida do projeto ou produto as baselines devem ser estabelecidas. As baselines mais comuns devem ser definidas ao final de cada uma das fases de Iniciação, Elaboração, Construção e Transição. Elas também podem ser geradas no final de iterações ocorridas dentro das várias fases ou com frequência ainda maior]

2.2.1 Processamento e Aprovação de Solicitações de Mudança

[Descreva o processo pelo qual os problemas e as mudanças são submetidos, revisados e dispostos]

2.2.2 Responsável pelo Controle de Mudança

[Descreva o responsável pelos procedimentos para processar solicitações de mudança e aprovações a serem seguidos]

2.2.3 Mecanismo de armazenamento e Controle de Mudança.

[Descreva como será mantido os controles, tipo de arquivos (baseline), software de apoio quando houver]

2.2.4 Relatórios de Controle de Mudança.

[Descreva os relatórios que serão elaborados para análise e acompanhamento das mudanças]

<Nome da Inst. Parceira>	Versão: <1.0>
<Nome do Componente>	Data: <24/06/2014>
Gerenciamento da Configuração	

2.3 Itens da Configuração

[Indicações dos itens que serão controlados]

2.3.1 Ambiente de Desenvolvimento

[Indique todos os itens do ambiente desenvolvimento que devem ser controlados]

- Equipamentos.
- Produtos de Software
- Entre outros...]

2.3.2 Artefatos Produzidos.

[Indique todos os artefatos que devem ser controlados pelo Gerenciamento de configuração quando produzidos durante o projeto]

- Especificação de Produto.
- Especificação de Componente
- Entre outros...]

2.3.3 Itens de Software.

[Indique todos os artefatos que devem ser controlados quando produzidos]

- Programa Fonte.
- Executáveis.
- Modelo de Software
- Entre outros...]

2.3.4 Itens de Hardware.

[Indique todos os artefatos que devem ser controlados quando produzidos]

3. Controle de Subcontratados e Fornecedores

[Descreva de que forma o componente adquirido ou desenvolvido fora do ambiente do projeto será incorporado, identifique também seus pontos de controle de configuração]

3.1.1 Itens de Software de Subcontratados.

[Indique todos os artefatos que podem ser controlados quando utilizados]

- Artefatos.

3.1.2 Itens de Hardware de Subcontratados.

[Indique todos os artefatos que podem ser controlados quando utilizados]

- Componente.

4. Matriz de Rastreabilidade

[Descreva de que forma a matriz de rastreabilidade será organizada, meio de armazenamento, endereçamento. Deve-se definir quais as informações deverão estar nas matrizes e forma de rastreabilidade. A matriz pode ser organizada em ferramentas de controle de configuração, em arquivos tipo Word ou Excel, ou outros meios]

Apêndice I – Processo completo versado em inglês

Process for the Development of Embedded System

**CMMI Level 2
Version 1.0**

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Revision History

Date	Version	Description	Author

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

List of Figures

Figure 1.1 - Organization of the standard ECSS Disciplines.....	11
Figure 2.1 - Set Symbology.	12
Figure 2.2 - Set of symbols.	13
Figure 2.3 - Representation of parts of a system.....	14
Figure 2.4 - Cycle process.	15
Figure 2.5 - The cyclic process components.....	16
Figure 3.1 - Common activities.....	18
Figure 3.2 - Phase 1 – Engineering.	19
Figure 3.3 - Phase 1 activities.	20
Figure 3.4 - Phase 2 - Requirements analysis.	21
Figure 3.5 - Activity of phase 2.....	22
Figure 3.6 - Phase 3 - Analysis and design.	23
Figure 3.7 - Phase 3 activities.	24
Figure 3.8 - Phase 4 - Implementation and integration.....	25
Figure 3.9 - Activities of phase 4 - implementation.	26
Figure 3.10 - Activities of phase 4 - integration.....	27
Figure 3.11 - Phase 5 - Verification and validation.....	28
Figure 3.12 - Activities of phase 5.	29
Figure 3.13 - Phase 6 - cycle assessment.	30
Figure 3.14 - Activities of phase 6.	31
Figure 4.1 - Guidelines for the understanding of the flowchart.	33
Figure 4.2 - Conduct of the proceedings of general flowchart - part-1.	34
Figure 4.3 - Conduct of the proceedings of general flowchart - part-2.	35
Figure 4.4 - Reports to flowchart of phases.	36
Figure 4.5 – Engineering.	37
Figure 4.6 - Project control.	38
Figure 4.7 - Configuration control - 2.	39
Figure 4.8 - Define scope-3.	40
Figure 4.9 - Planning-4.	41
Figure 4.10 - Analyze risks-5.....	42
Figure 4.11 - Plan metrics-6.....	43
Figure 4.12 - Requirements analysis.	44
Figure 4.13 - Specification requirement-7.....	45
Figure 4.14 - Component Definition-8.....	46
Figure 4.15 - System planning of-9.....	47
Figure 4.16 - Component design-10.....	48
Figure 4.17 - Set control items-11.....	49

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Figure 4.18 - Flowchart analysis and design phase.....	50
Figure 4.19 - Detail component-12.....	51
Figure 4.20 - Project hardware-13.....	52
Figure 4.21 - Software design-14.....	53
Figure 4.22 - Prototype-15.....	54
Figure 4.23 - Validate model-16.....	55
Figure 4.24 - Implementation and integration.....	56
Figure 4.25 - SW Implementation-17.....	57
Figure 4.26 - HW Implementation-18.....	58
Figure 4.27 - Unit testing-19.....	59
Figure 4.28 - Integration-20.....	60
Figure 4.29 - Component test-21.....	61
Figure 4.30 - Flowchart of verification and validation phase.....	62
Figure 4.31 - SW Test-22.....	63
Figure 4.32 - HW test-23.....	64
Figure 4.33 - Cycle assessment.....	65
Figure 4.34 - Closing-24.....	66
Figure 4.35 - Evaluation-25.....	67
Figure 5.1 - Word document properties (templates).....	68
Figure 5.2 - Word template resource representation.....	69
Figure 5.3 - Representation of the word document.....	69
Figure 5.4 - header and footer template.....	70
Figure 5.5 - Typing Office in document.....	71

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

List Of Tables

Table 6.1 - Generic practices.	73
Table 6.2 - Management of requirements.	74
Table 6.3 - Measurement and analysis.	75
Table 6.4 - Monitoring and project control.	76
Table 6.5 - Management in accordance with suppliers.	77
Table 6.6 - Project planning.	78
Table 6.7 - Configuration management.	80
Table 6.8 - Process and product quality assurance.	81
Table 6.9 - Requirements development.	82
Table 6.10 - Technical solution.	83
Table 6.11 - Product integration.	85
Table 6.12 - Verification.	86
Table 6.13 - Validation.	87
Table 6.14 - Focus Organization Process.	88
Table 6.15 - Definition of organization process.	89
Table 6.16 - Organizational training.	90
Table 6.17 - Integrated project management.	91
Table 6.18 - Risk management.	92
Table 6.19 - Decision analysis and resolution.	92
Table 6.20 - Organizational process performance.	93
Table 6.21 - Project quantitative management.	94
Table 6.22 - Causal analysis and resolution.	94
Table 6.23 - Management of organizational process.	95

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

INDEX

List of Figure

List of Figures	3
List Of Tables	5
1. Introduction	9
1.1. Finality	9
1.2. Scope of the document.....	9
1.3. Involved	10
1.4. Positioning the proposed process in the standard ECSS.....	10
1.4.1. Organization of the standard ECSS disciplines.....	11
2. Process graphic representation	12
2.1. How to understand the layout of process components.....	12
2.2. Notations used	13
2.3. Cyclic process	13
3. Description of Phases	16
3.1. Description of document templates that will result from activities carried out in stages of the process.	16
3.2. Common activities to the phases (FC).....	17
3.3. Engineering – (F1).....	19
3.4. Requirements Analysis (F2).....	21
3.5. Analysis / Design (F3).	23
3.6. Implementation and integration – (F4).	25
3.7. Verification and validation – (F5).	28
3.8. Cycle Assessment – (F6).	30
4. General flowchart of the process.....	33
4.1. Guidelines for the understanding of the flowchart	33
4.2. Flowchart of process.....	34
4.3. Flowchart of each phase process	36
4.3.1. Guidelines for the phases of the flowchart of understanding	36
4.3.2. Flowchart of engineering phase - F1	37
4.3.2.1. Flowchart of project control activity - 1	38
4.3.2.2. Flowchart configuration control activity - 2	39
4.3.2.3. Flowchart activity set scope - 3	40
4.3.2.4. Flowchart activity planning - 4	41
4.3.2.5. Activity flowchart analyze risks - 5.....	42
4.3.2.6. Flowchart activity plan metrics - 6.....	43
4.3.3. Flowchart requirement analysis phase – F2.....	44
4.3.3.1. Flowchart activity specification requirement - 7.....	45

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

4.3.3.2. Flowchart setting activity component - 8.....	46
4.3.3.3. Flowchart activity system planning - 9.....	47
4.3.3.4. Flowchart component project activity - 10	48
4.3.3.5. Flowchart activity set control items - 11.....	49
4.3.4. Phase of the flow chart of analysis and design – F3.....	50
4.3.4.1. Flowchart detail component activity - 12	51
4.3.4.2. Flowchart of project activity hardware - 13	52
4.3.4.3. Flowchart software project activity - 14.....	53
4.3.4.4. Flowchart prototype activity - 15	54
4.3.4.5. Flowchart activity validate model - 16.....	55
4.3.4.6. Flowchart of the implementation and integration phase – F4	56
4.3.4.7. Flowchart activity SW implementation -17	57
4.3.4.8. Flowchart of HW implementation activity - 18.....	58
4.3.4.9. Flowchart of unit testing activity - 19.....	59
4.3.4.10. Flowchart integration activity - 20	60
4.3.4.11. Flowchart component test activity - 21.....	61
4.3.5. Flowchart for verification and validation phase – F5.....	62
4.3.5.1. Flowchart activity SW test - 22.....	63
4.3.5.2. Flowchart of HW test activity - 23	64
4.3.6. Phase Flow chart cycle assessment phase – F6.....	65
4.3.6.1. Flowchart closing activity - 24.....	66
4.3.6.2. Flowchart activity evaluation - 25	67
5. Documents using standard.	68
5.1. Document property.	68
5.2. Auto complete document.	68
5.3. Typing in the Document Interior.....	70
6. CMMI and Process	72
6.1. Level 2	72
6.2. Service CMMI Practices	72
6.2.1. Generic.....	72
6.2.2. Specific Level 2.....	74
6.2.2.1. Manages requirements.	74
6.2.2.2. Measurement and analysis	75
6.2.2.3. Monitoring and project control.....	76
6.2.2.4. Management in accordance with suppliers	77
6.2.2.5. Project planning.....	78
6.2.2.6. Configuration management.....	79
6.2.2.7. Quality assurance	81
6.2.3. Specific level 3.....	82
6.2.3.1. Requirements development.....	82

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

6.2.3.2. Technical solution	83
6.2.3.3. Product integration	85
6.2.3.4. Verification	86
6.2.3.5. Validation.....	87
6.2.3.6. Focus on the organization process.....	88
6.2.3.7. Definition of organization process.	88
6.2.3.8. Organizational training	89
6.2.3.9. Project integrated management	90
6.2.3.10. Risk management	91
6.2.3.11. Decision analysis and resolution	92
6.2.4. Specific level 4	93
6.2.4.1. Organizational process performance	93
6.2.4.2. Quantitative project management.....	93
6.2.5. Specific level 5.....	94
6.2.5.1. Causal analysis and resolution	94
6.2.5.2. Management of organizational process	95
7. References.....	96

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

1. Introduction

1.1. Finality

The Document Embedded Systems Development Process aims to define all the phases, activities and artifacts proposed to develop an embedded system.

This process is the result of a research paper for the doctor title in Engineering and Technology Space / Space Systems Management conducted at the National Institute for Space Research - INPE. Entitled "AN APPROACH TO SYSTEM DEVELOPMENT PROCESS BOARDED ATTENDING THE LEVEL OF 2 CMMI-DEV MATURITY." With defense date scheduled for February 10, 2015 at 10:00 am by doctoral student Magda Aparecida Silverio Miyashiro to be held in Room A - INPE Rotunda Building, with the Examining Committee of the components the following doctors:

Prof. Dr. Ronaldo Arias President of the examination board - INPE;

Prof. Dr. Mauricio Vieira Gonçalves Ferreira Advisor - INPE,

Prof. Dr. Nilson Sant'Anna Advisor - INPE;

Prof. Dr. Vanderlei Cunha Parro Guest Inst. Maua Technology;

Prof. Dr. Mauro Spinola Guest USP.

1.2. Scope of the document

This document presents a development process for embedded systems that incorporates the practices required by the CMMI Level 2 where:

To facilitate the interpretation of the process, section 2 presents the meaning of each component and its graphic representation and its symbolism and also shows the cyclic process. Section 3 provides a breakdown and description of functions and description of all components of the cyclic process, section 4 presents the suggested general flowchart of procedure, section 5 presents the automated structure for the use of templates with the resources of word and section 6 presents as the cyclical process meets each practice model.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

1.3. Involved

Elaboration

Ms: Magda Aparecida Silvério Miyashiro

Case Study

INPE - Hardware and Software Laboratory of Ground System Development Division (DSS) coordination of Space Engineering and Technology (ETE) at INPE (In initial phase of the use).

IMT-Maua Institute of Technology, System Center Embedded Electronics - NSEE (In initial phase of the use).

Advisor

Prof^o. Dr. Mauricio G.V. Ferreira

Prof^o. Dr. Nilson Sant'anna

1.4. Positioning the proposed process in the standard ECSS

The pattern used by the National Institute for Space Research - INPE, in conducting a space mission is the European Cooperation for Space Standardization (ECSS), so for the development of this process has been defined as a special requirement that it conforms to that standard.

The Standard ECSS in each branch disciplines and their requirements are covered by specific standards and supported by Technical manuals and memoranda when necessary:

- **Standards** - Document specifically for use in agreements between undertakings for the implementation of activities related to space area.
- **Manual** - not normative document that provides more detailed information as guidance, guidelines, technical data, opinions or recommendations on how to implement activities related to the space area, where there are two types of manual are guidelines and best practices and data processing.
- **Technical Memoranda** - not normative document that provides useful information for the space community on a particular theme.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

1.4.1. Organization of the standard ECSS disciplines

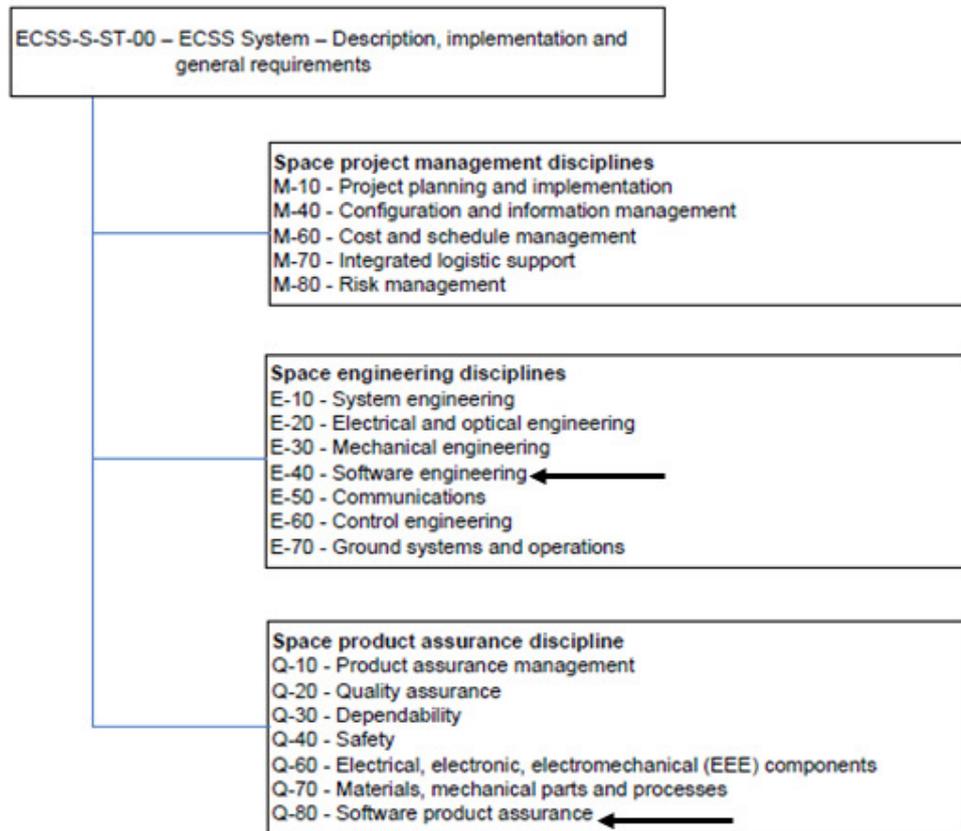


Figure 1.1 - Organization of the standard ECSS Disciplines.

The rules dealing with software engineering and software quality are ECSS-E-ST-40 - Space Engineering Software, ECSS-Q-ST-80 - Space product assurance - Software product, respectively.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

2. Process graphic representation

2.1. How to understand the layout of process components

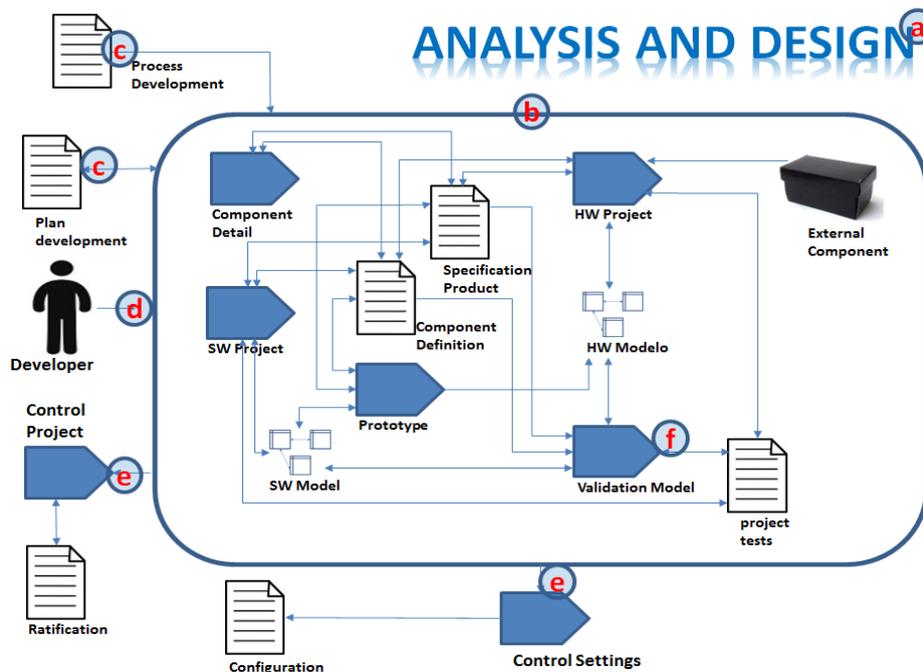


Figure 2.1 - Set Symbology.

a) Phase name: used for identification.

b) Rounded Rectangle: used to define the scope of the activities performed at that stage, all components of the process that within the rectangle represent activities or artifacts produced or handled at that stage, all process components that are outside the rectangle are for activities and artifacts that are used or performed in all phases. Note: The process components that are located outside of the rectangle and has no association with rectangle are produced by the activity that is associated with the process component. The process components that are associated with the rectangle are understood to be associated with all devices that can be associated.

c) Associated document directly to the rectangle: used as reference during the stage. Observe the direction of the arrow is single arrow exiting the document towards the rectangle means document reading. If it has bidirectional sense, it means the document can be updated during the phase.

d) Actor associated with the rectangle: all internal activities to the rectangle are performed by him.

e) Activity directly associated to the rectangle: are performed at every stage.

f) Associated with document Activity: carrying out the activity results in the associated document.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Considerations: The associations should exist between actor-activity and activity-document because every activity has a result and every document is the result of some activity and every activity is performed by an actor.

There are no associations between actor-document, document-document and activity-activity. For example, the actor always updates a document through an activity.

2.2. Notations used

The symbols used to standardize the representation of the process components is shown in Figure 2.2.

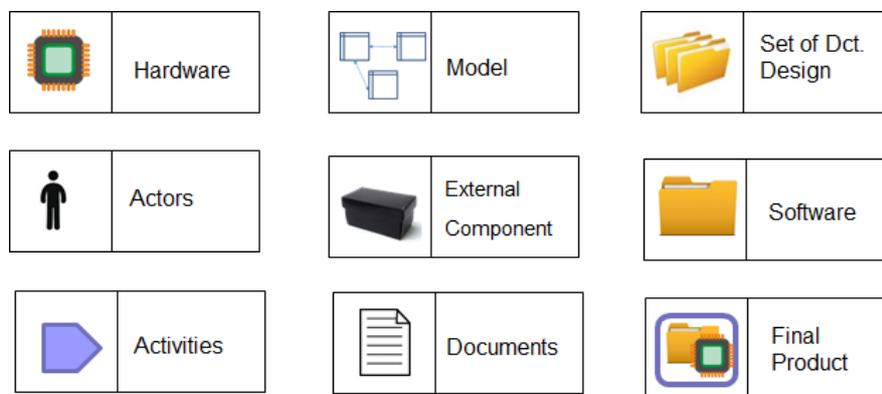


Figure 2.2 - Set of symbols.

2.3. Cyclic process

Development cycle is the sequential completion of all phases and activities of the life cycle. Each development cycle will result in a component of an embedded system and all documents produced during its development. When an embedded system is composed of many components must have different development cycles, one for each of compomente embedded systems, as shown in Figure 2.3.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

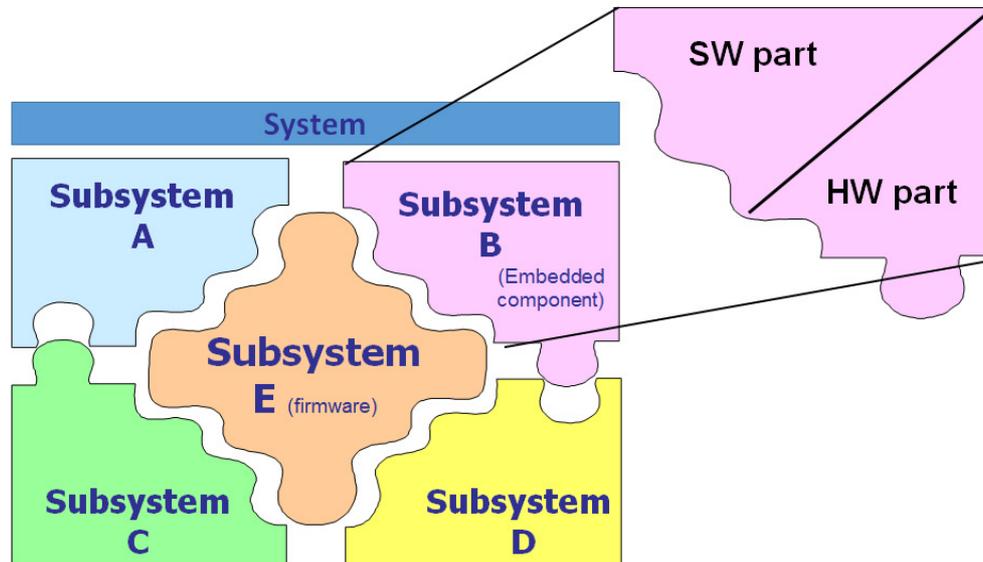


Figure 2.3 - Representation of parts of a system.

The structure of the development process consists of various steps to be performed in a cyclic and sequential manner as shown in Figure 2.3.

Cyclic: the process is called cyclic because each component (hardware + software) of embedded system to be developed according to your specifications must perform all phases and activities, thus forming an embedded system components development cycle and for each component will be developed a plan in accordance with the process and all stages and planned activities will take place.

Sequential: The phases and activities must be performed sequentially, where a phase starts after the previous is completed and approved, using as reference the classic model.

In this approach, we propose the process for the development of embedded systems with focus on project management and good system development practices, through a complete development cycle, that is, all life cycle phases shown in Figure 2.4.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

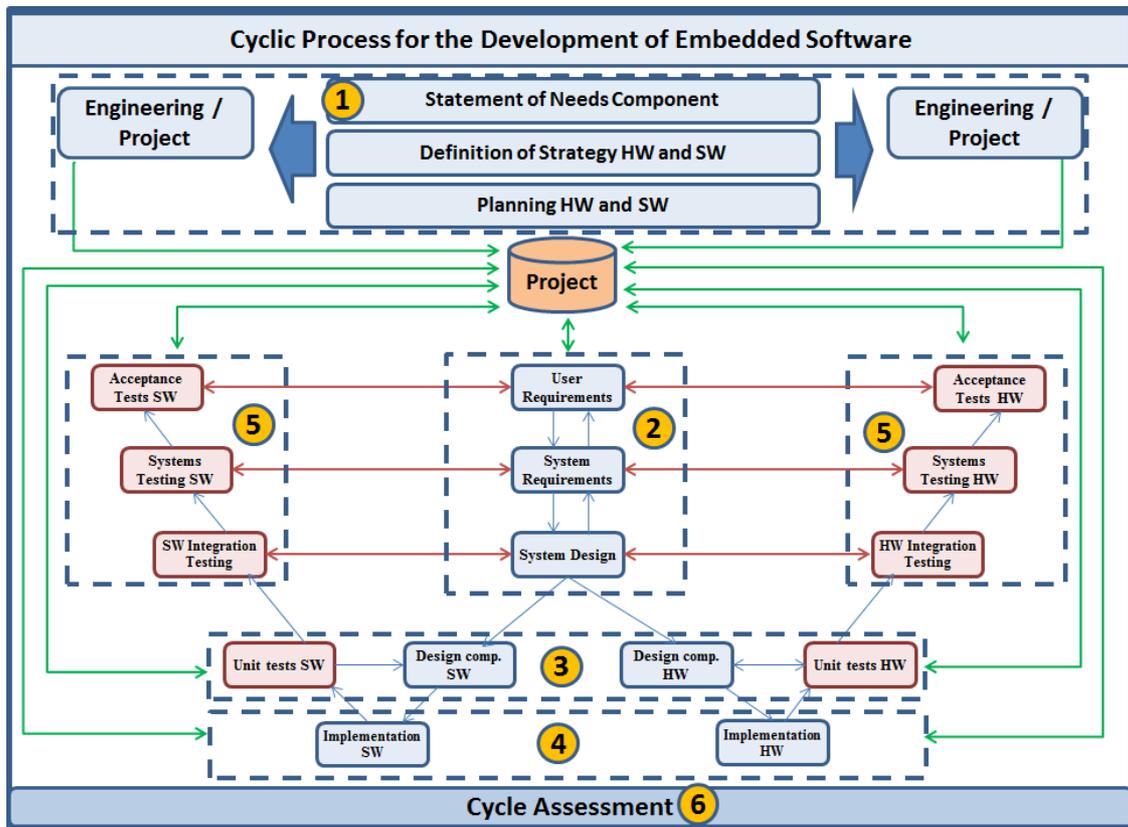


Figure 2.4 - Cycle process.

Each cycle should establish working relationships, define and distribute the roles to team members and set goals, strategies and work plans for all phases of the process. Where appropriate, should make changes in the process, according to the results of the assessment cycle.

In the structure shown in Figure 2.5, the phases are the grouping of activities in the life cycle; activities are procedures that must be performed at each stage; and actions are detailing the completion of each activity specifying the sequence to be followed, deviations where relevant, the expected outputs and artifacts produced.

The process is organized in phases, consisting of activities and implemented through actions governed by procedures and documents (models) that result in common artifacts, which make up the embedded system product (programs and documentation).

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

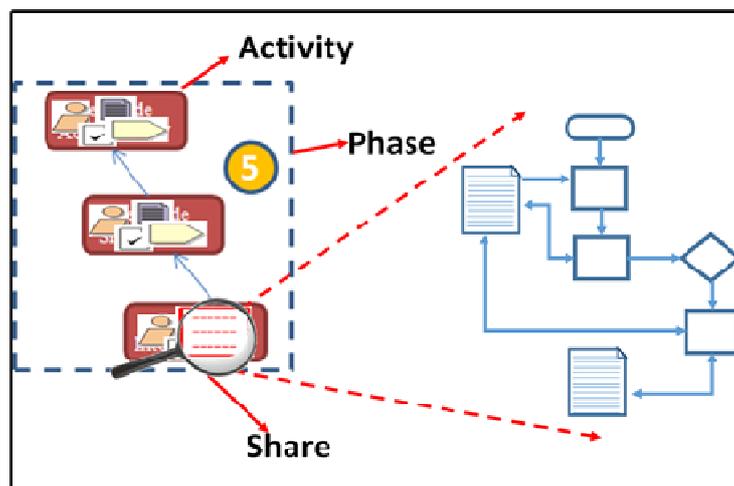


Figure 2.5 - The cyclic process components.

3. Description of Phases

3.1. Description of document templates that will result from activities carried out in stages of the process.

- **Development Process** - A document with a set of activities, organized and formalized that produce outputs (artifacts) predefined in order to assist in building components with the requisite quality. Adapted document from the rup_tstpln rup artifact.
- **Approval Plan** - Document resulting from the follow-up and validation of each project phase, which records the checking of all artifacts produced by each phase, according to the specifications, expectations and the expected outputs, as well as the commitment and results involved. This document is produced in the Engineering phase and fed in all other phases. This document may be disregarded when its information is incorporated in the development plan document. Adapted document, from the rup_pacpln rup artifact.
- **Configuration** - Resulting Document from planning, control and records of configuration items, for the development environment, use and, languages as well as all the artifacts produced during the process including it Configuration document itself. This document is produced in the engineering phase and fed in all other phases in accordance with the changes. Adapted document, from the rup_cmpln rup artifact.
- **Development Plan** - A document with the result of the set of activities, organized and formalized with allocation, sequences of time, resources and dependencies between tasks to the development of the hardware and the embedded software component. Adapted document, from the rup_sdpln rup artifact.
- **Product specification** - The document is the output of different activities carried out during the process (activities related to the description and definition of the product). This document covers all information, specifications and needs of the complete product to be

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

built (embedded system). This document is of great importance, but is optional for this process as of the cyclic process directs the component development individually. It is understood that the product specification is an existing document to higher development instance. Adapted document rup artifact rup_vision.

- **Component Definition** - The document is the output of different activities throughout the process, activities related to the description, definition and component design (hardware and software). This document covers all information, specifications and component needs to be produced and inserted into the embedded system. Adapted document from the rup_cmpln rup artifact.
- **Test Project** - Is a document resulting from the development of systematic procedures to perform component testing (hardware and software). Should be planned in detail according to workflow according to the evolution of the product at all stages that are relevant. Adapted document the rup_tstpln rup artifact.
- **External components** - Complete documentation of external components to the project that may be used. Are considered outside those that were built in other projects or purchased from the market.
- **SW model** - It's the resulting graphical representation of the construction activity of the software model that explains the characteristics and behavior of the component part of the software to be developed and its hardware integration and their integration into embedded system. Its content is inserted into the Product Specification document and Component.
- **HW model** - It's the resulting graphical representation of the construction activity of the hardware model that explains the features and the hardware component of the behavior to be developed, as well as its integration with the software and their integration into embedded system. Its content is inserted into the Product Specification document and Component.
- **Software** - Program or set of programs built according to the specifications and documents resulting from previous activities.
- **Hardware** - The electronic components built according to the specifications and documents resulting from previous activities.
- **Product-Final** - Embedded component produced by the hardware and software integration.

3.2. Common activities to the phases (FC).

Activities to be carried out during development process of embedded component, shown in Figure 3.1.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

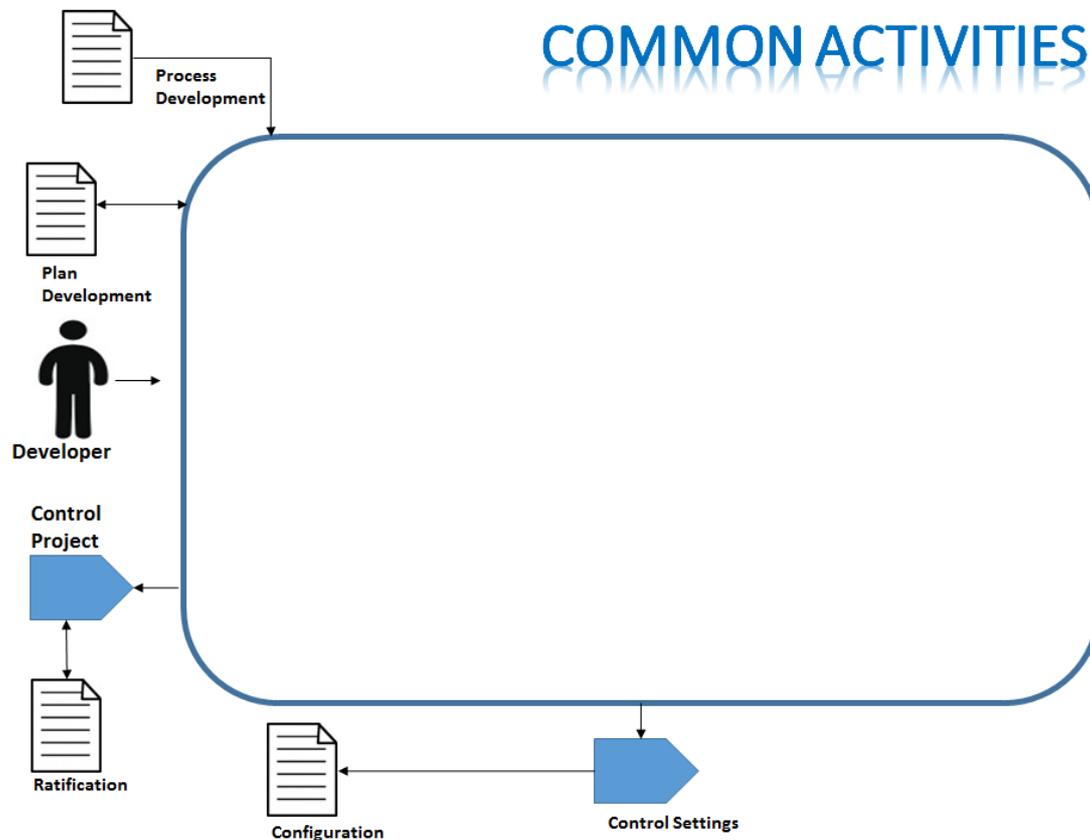


Figure 3.1 - Common activities.

Actors:

- **Developer** - Is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **Control setup -2** - In this activity you define the structure and configuration required for the development of the project, identify and describe all items to be controlled, and all the changes that produced artifacts may suffer, as well as all those in charge.
- **Project control -1** - This activity is supposed to perform actions for project monitoring, monitoring of actions taken, project performance evaluation according to plan and make necessary changes to the project according to the project context.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

Involved artifacts:

- Development process.
- Approval plan.
- Configuration.
- Development plan.

3.3. Engineering – (F1)

Activities that must be performed in the engineering phase of the embedded component, is shown in Figure 3.2.



Figure 3.2 - Phase 1 – Engineering.

Representation of phase activities

Organization of the engineering phase of activities, shown in Figure 3.3.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

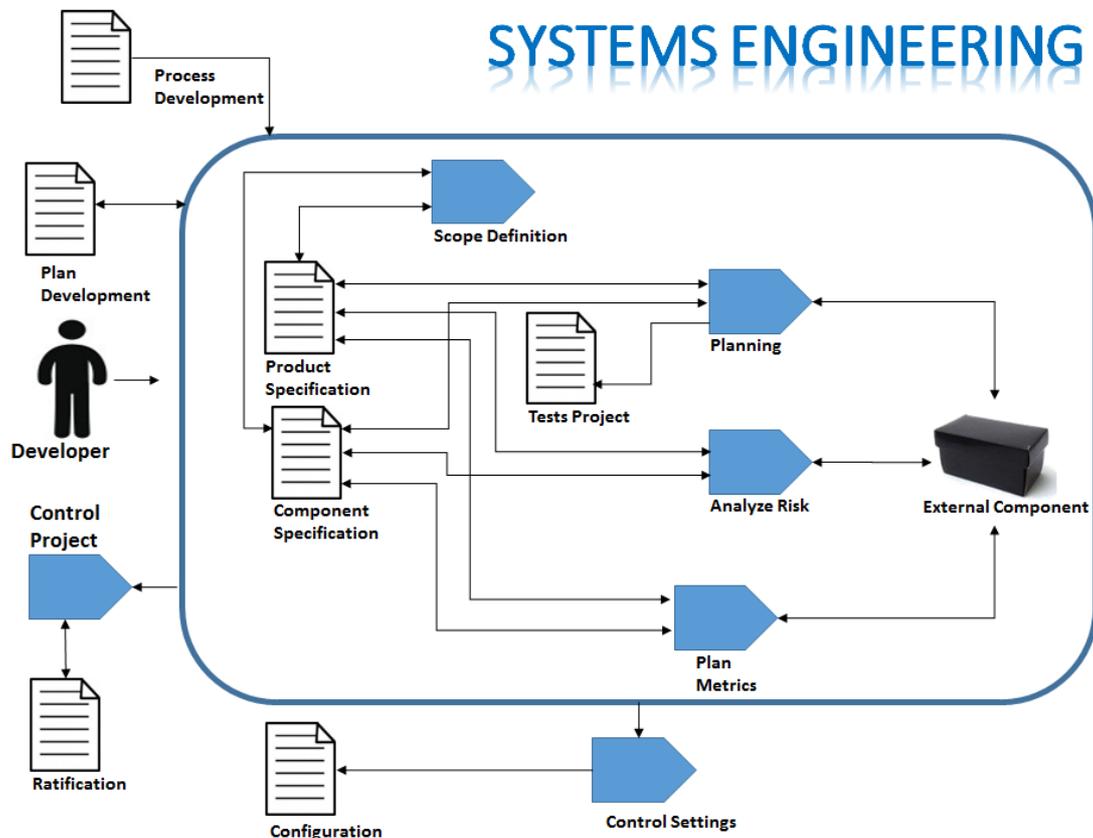


Figure 3.3 - Phase 1 activities.

Actors:

- **Developer** - Is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **Scope definition -3** - Activity carried out to define the border of the features to be performed by each component and their insertion in the context, it can also present the features of the full product (system). The product specification document is optional, since the procedure is applied individually for each component.
- **Planning - 4** - Activity adaptation, preparation, organization and structuring of the process for the development of the component according to the process guidelines and offers parameters for monitoring, measurement and decision making.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

- **Analyze risk - 5** - Checking the critical points that may present imperfection or failure during the course of the project that might compromise the development of the component. It is the study of each development phase to identify the possible risks that may occur and avoid them. In their actions are included: identification, analysis, classification, Planning, Monitoring and Tracking Control of risks.
- **Plan métrics – 6** - Activity performed for the identification and understanding of the measures that need to be considered as a control mechanism and evaluation. In this activity should be defined quality metrics development process of the component in a special way, for the components of embedded systems, should be planned very precisely, the product quality measurement to be developed. In this activity must be identified all the component requirements to be developed and measured along its construction.

Involved Artifacts:

- Product Specification
- Component Specification
- Test Project
- External components

3.4. Requirements Analysis (F2)

Activities that must be performed in the requirements analysis phase of the embedded component, and shown in Figure 3.4.

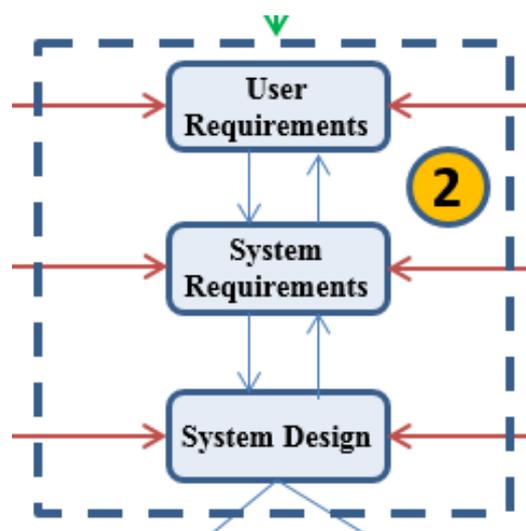


Figure 3.4 - Phase 2 - Requirements analysis.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

Representation of Phase activities

Organization of the activities of the Requirements Analysis phase, shown in Figure 3.5.

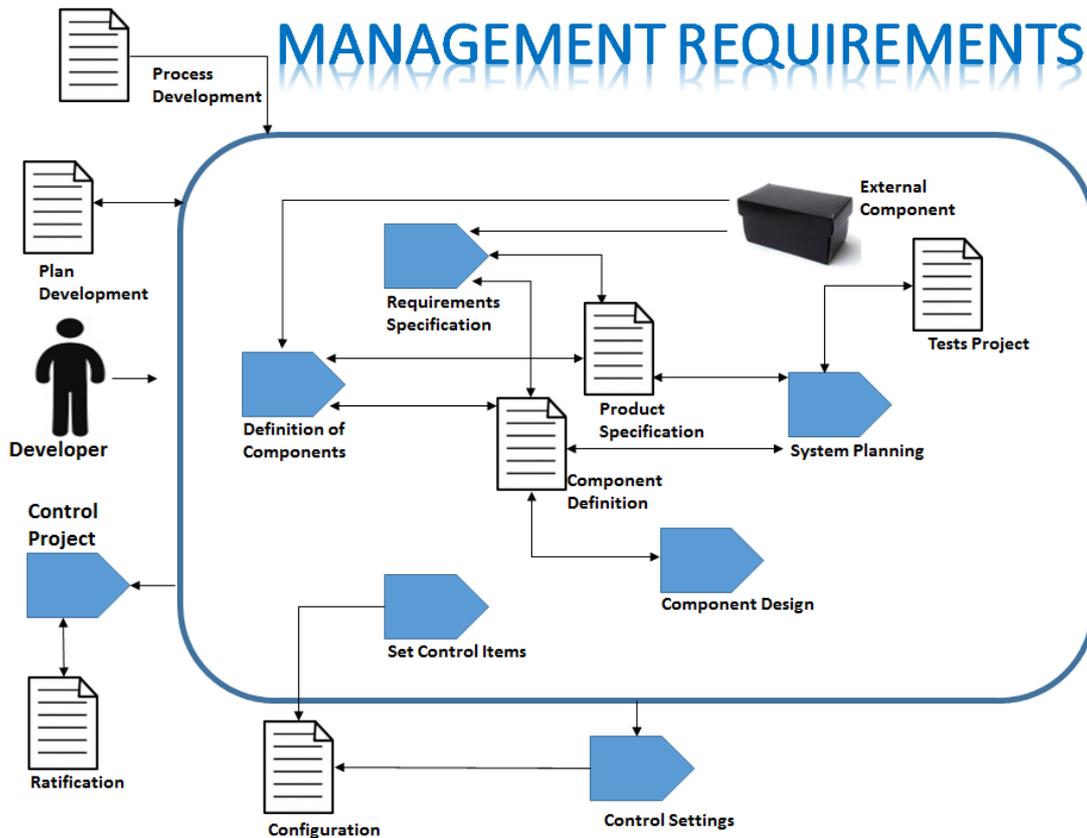


Figure 3.5 - Activity of phase 2.

Actors:

- **Developer** - Is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **Requirements specification - 7** - Detailed definition of product requirements to be built.
- **Definition of the components - 8** - Details of the component to be

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

developed.

- **Set Control items – 11** - According to the requirements, you must identify all items in the product / component and environmental structure of items to be controlled by the management of configuration. Indication of how control and tools to be used.
- **Component design – 10** - Component of the project development (hardware and software) to be built.
- **System planning - 9** - Individualization and specification of the tasks assigned to the software and hardware component to be built.

Involved Artifacts:

- Product Specification.
- Definition Component.
- Test Project.
- External components.

3.5. Analysis / Design (F3).

Activities that have to be performed in the embedded component analysis and Project, shown in Figure 3.6.



Figure 3.6 - Phase 3 - Analysis and design.

Representation of phase activities

Organization of activities analysis and design phase, shown in Figure 3.7

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

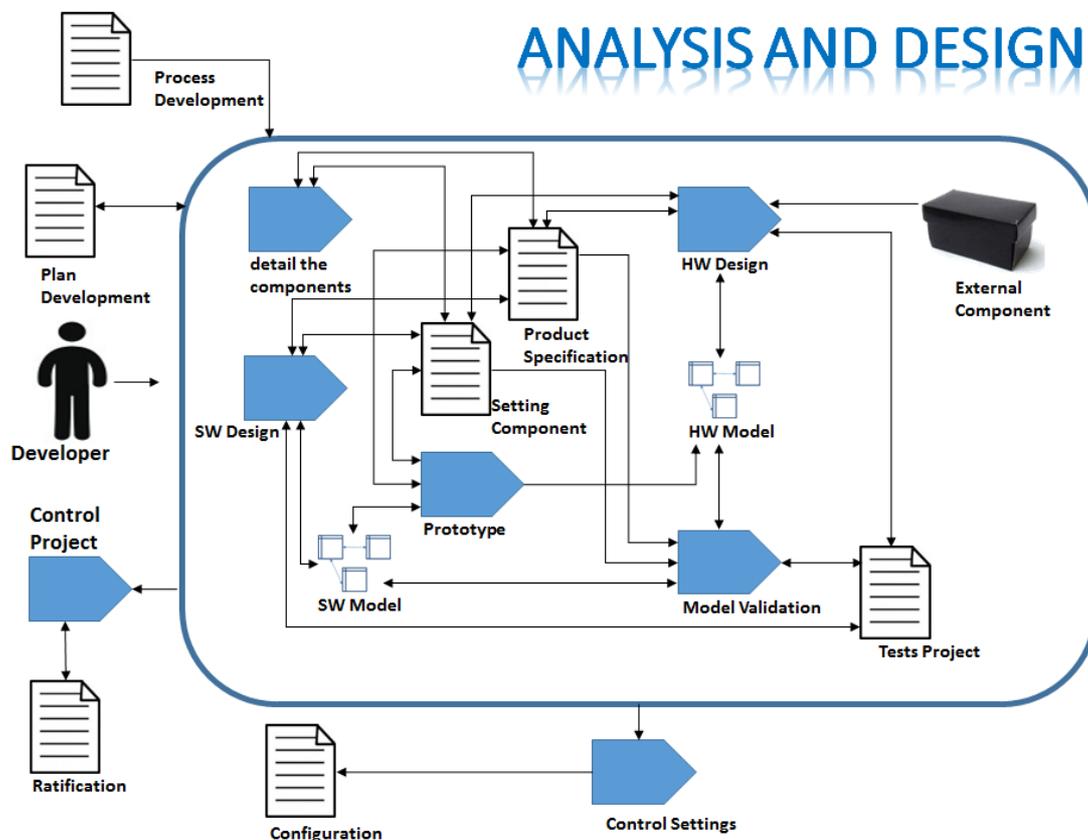


Figure 3.7 - Phase 3 activities.

Actors:

- **Developer** - is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **Detail components – 12** - Details of the items that will be used by hardware and software components according to the individualization of tasks, this activity will feed the Product Specification documents and Definition of components.
- **SW Design – 14** - Software architectural design for the realization of prototyping and subsequent construction according to the architecture of the hardware. Analysis and definition of modeling and prototyping tools to be used. This activity will feed the Product Specification documents and Definition of components.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **HW design - 13** - hardware architecture design for the realization of prototyping and subsequent construction according to the software architecture. Analysis and definition of modeling and prototyping tools to be used. This activity will feed the Product Specification documents and Definition of components.
- **Prototype – 15** - Prototype construction of hardware models and software design to be validated.
- **Model validation – 16** - Validation of models and prototypes built in compliance with according to plan requirements.

Involved Artifacts:

- Product specification.
- Definition component.
- Test project.
- External components.
- SW model.
- HW model

3.6. Implementation and integration – (F4).

Activities to be carried out in the Implementation and Integration phase of the embedded component, and shown in Figure 3.8.



Figure 3.8 - Phase 4 - Implementation and integration.

Representation of the implementation activities of phase

Organization of the Implementation phase of activities, shown in Figure 3.9.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

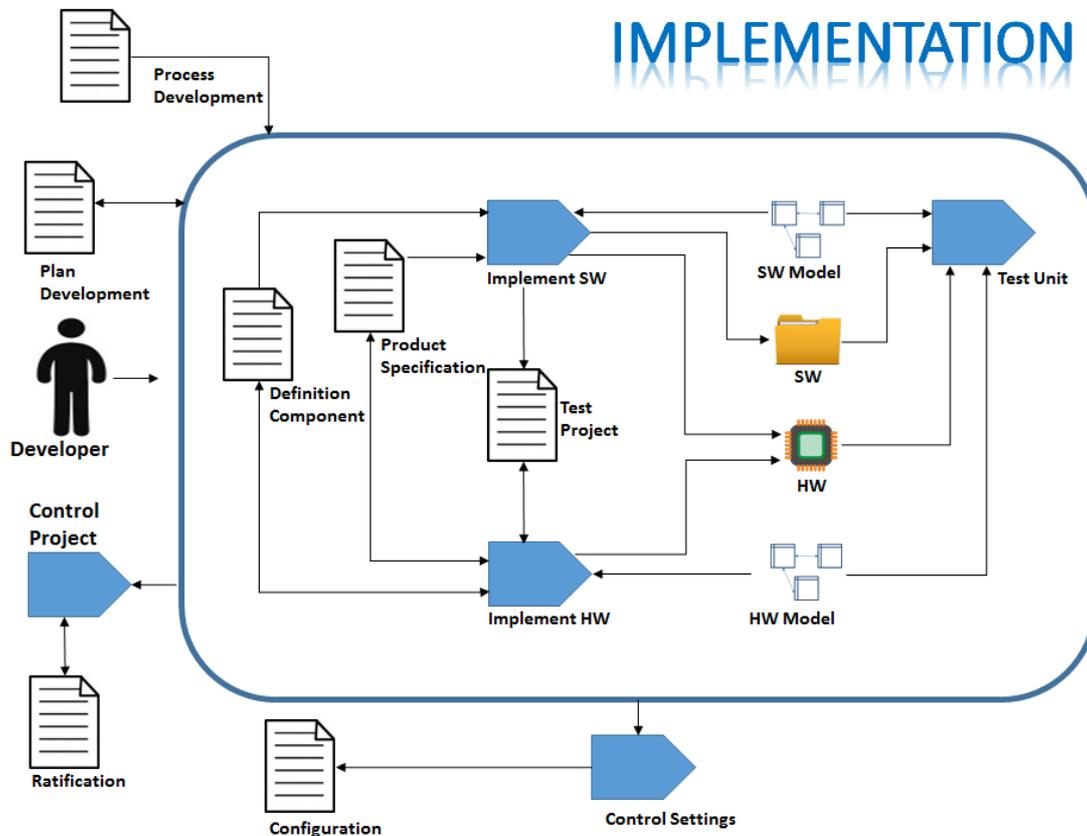


Figure 3.9 - Activities of phase 4 - implementation.

Actors:

- **Developer** - is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **Implement SW** – 17 - Construction of the software according to the specification and validated models.
- **Implement HW** – 18 - Production or acquisition of hardware that meets the specifications validated.
- **Test Unit** – 19 - Test and conference activities of each part of component (hardware and software) according to the test plan.

Involved artifacts:

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

- Product Specification.
- Definition Component.
- Test project.
- SW model.
- HW model.
- Software.
- Hardware.

Representation of phase integration activities

Organization of integration phase of activities, shown in Figure 3.10.

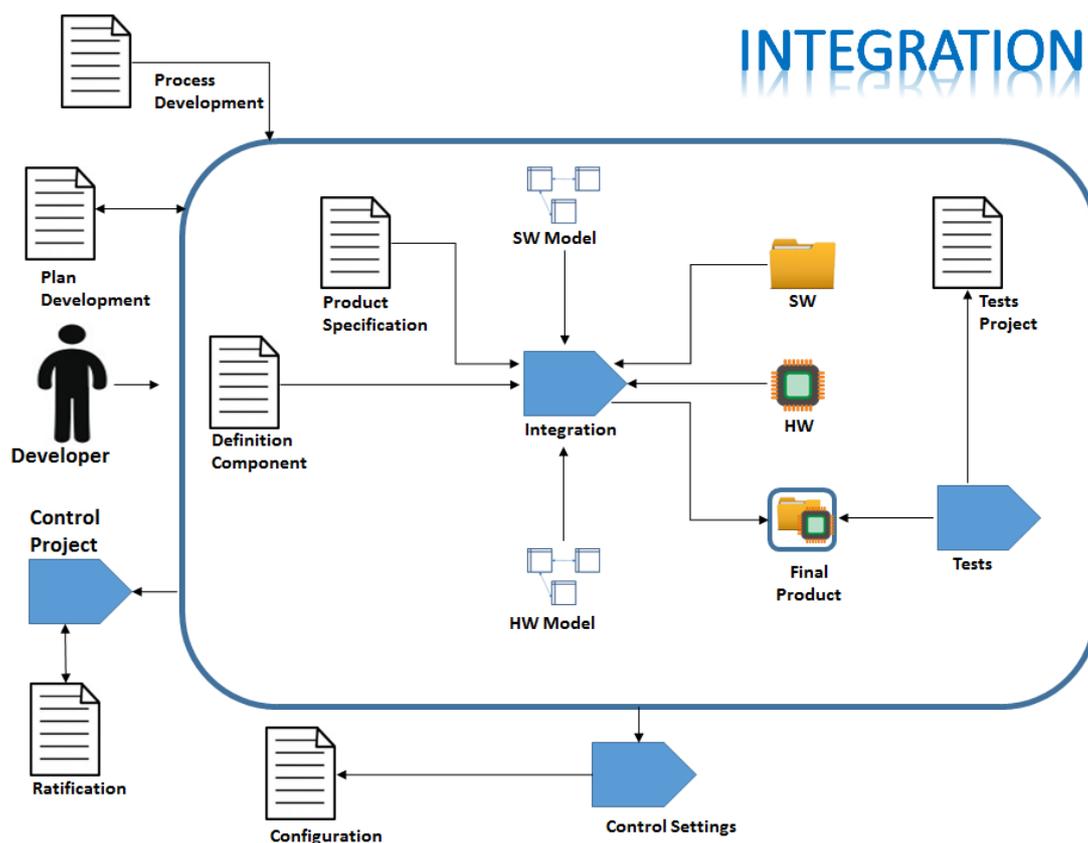


Figure 3.10 - Activities of phase 4 - integration.

Actors:

- **Developer** - is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

be held.

Activities:

- **Integration-SW-HW – 20** - Integration (connection) of the features of **the** hardware part and component part of the software;
- **Test – 21** - Test activities and checking on each of the component parts and the integrated component as well as the simulation of its environment usage according to the test plan.

Involved artifacts:

- Product specification.
- Definition component.
- Test project.
- SW model.
- HW model.
- Software.
- Hardware.
- Product-final.

3.7. Verification and validation – (F5).

Activities that must be performed in the verification and validation phase of the embedded component, and shown in Figure 3.11.

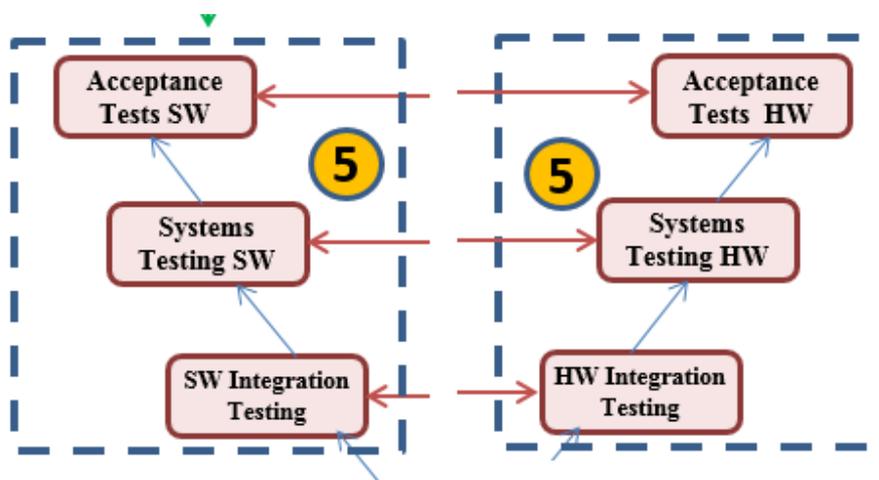


Figure 3.11 - Phase 5 - Verification and validation.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

Representation of phase activities

Organization verification and validation phase of activities, shown in Figure 3.12.

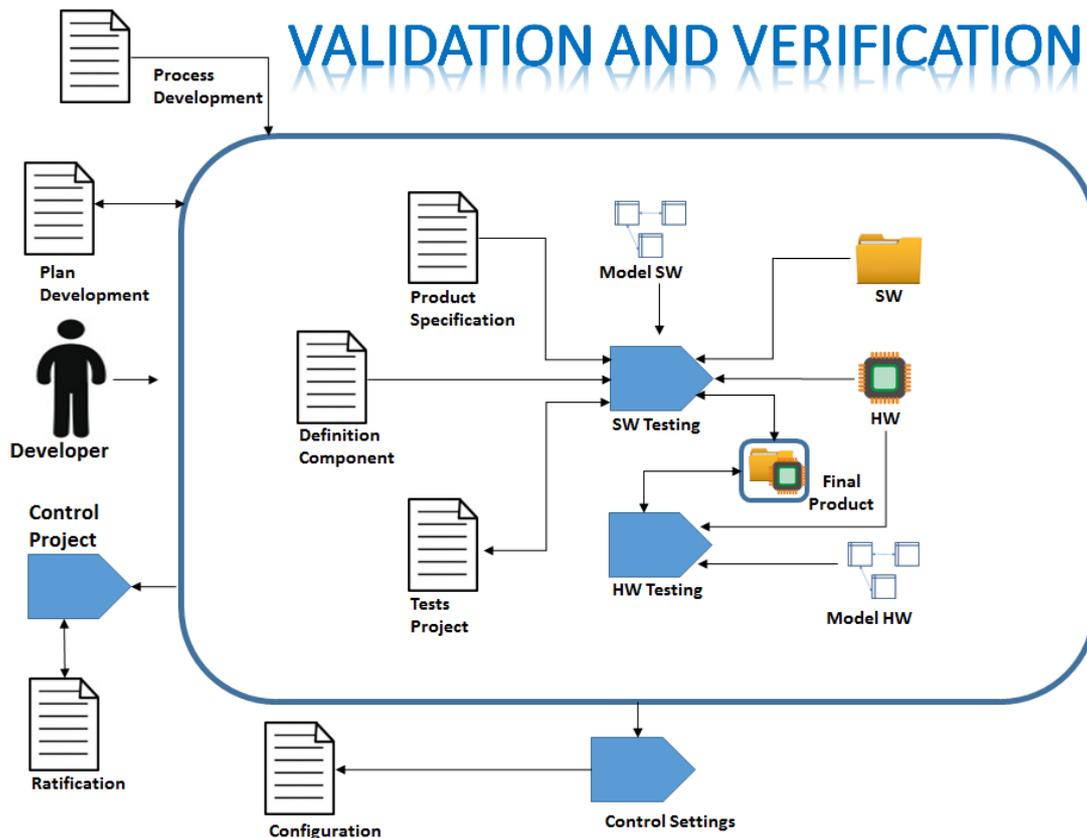


Figure 3.12 - Activities of phase 5.

Actors:

- **Developer** - is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **SW testing – 22** - Verification Tests and validation of the integrated component of software and its integration and services to hardware requirements, according to the test plan.
- **HW testing - 23** - Verification Tests and validation of the integrated component of hardware and integration and compliance with software

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

requirements, according to the test plan.

Involved artifacts:

- Product specification.
- Definition component.
- Test project.
- SW model.
- HW model.
- Software.
- Hardware.
- Product-final.

3.8. Cycle Assessment – (F6).

Activities that must be performed in cycle assessment phase of the embedded component, and shown in Figure 3.13.

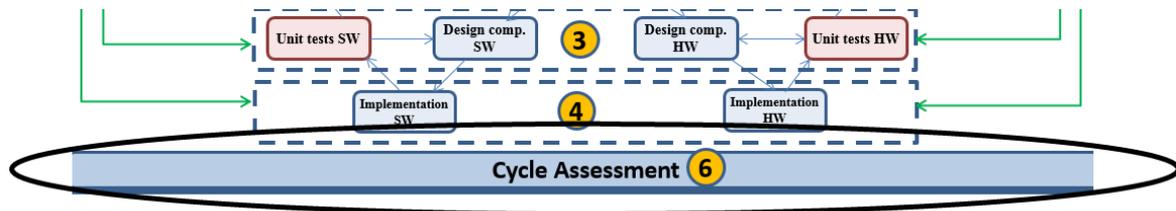


Figure 3.13 - Phase 6 - cycle assessment.

Representation of phase activities

Organization of the cycle assessment phase of the activities, shown in figure 3.14.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

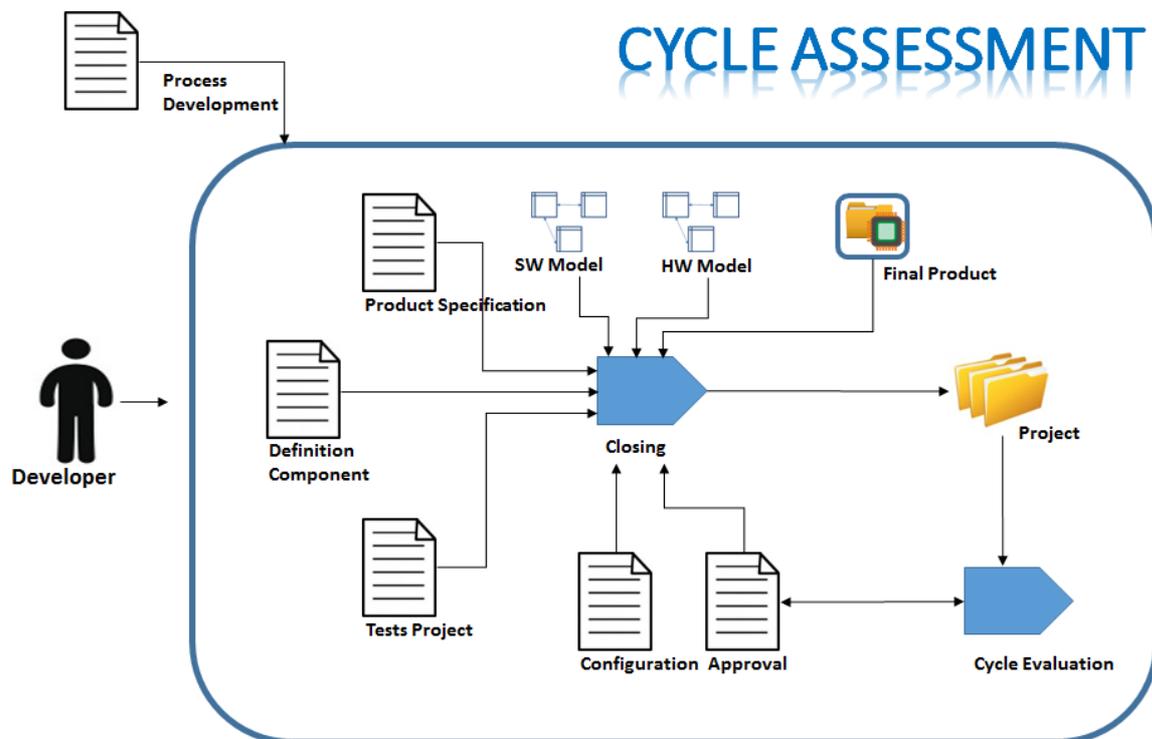


Figure 3.14 - Activities of phase 6.

Actors:

- **Developer** - is responsible for carrying out the activities defined in the process. It can take different roles in the process depending on the activity to be held.

Activities:

- **Closing – 24** - Completion and closing of all activities component of the development cycle and project design of the book.
- **Cycle evaluation – 25** - Analysis and evaluation of all activities performed during the cycle, the productivity and quality of process and product, and suggestions for adaptations and improvements for the next cycles.

Involved artifacts:

- Product specification.
- Definition component.
- Test project.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- Configuration
- SW model.
- HW model.
- Software.
- Hardware.
- Product-final.
- Approval.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4. General flowchart of the process

4.1. Guidelines for the understanding of the flowchart

- a) The proposed sequential flow for the realization of embedded component development process should be followed according to the following behavior shown in Figure 4.2 and Figure 4.3.
- b) Actions taken on the right bar (timing) are performed sequentially.
- c) The actions taken on left bar should be performed for all activities of the left bar.
- d) The acronyms F1, F2, F3, F4, F5, F6 and FC represent the codes of the phases where the activities are carried out.
- e) Notations Used.

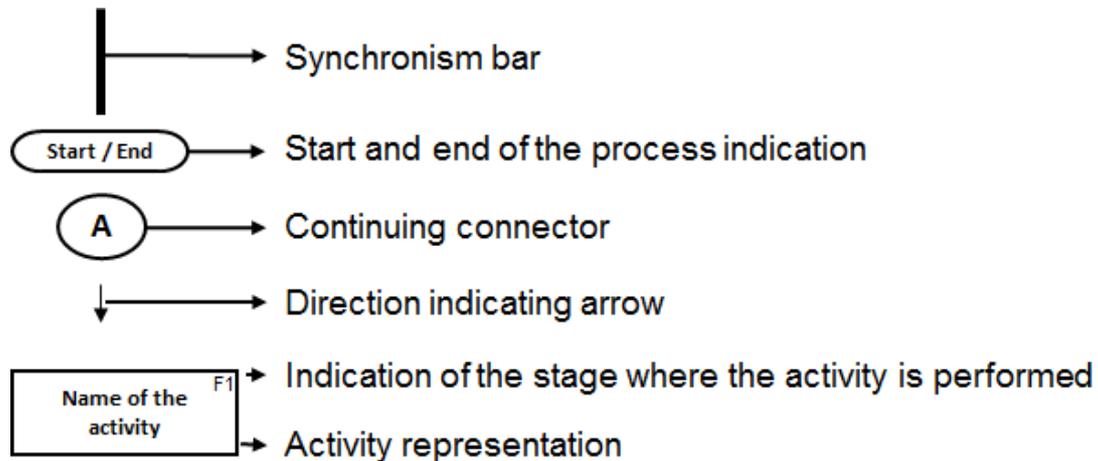


Figure 4.1 - Guidelines for the understanding of the flowchart.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

4.2. Flowchart of process

This suggests flowchart generally the main activities that must be performed including all the stages of the process (part 1 and part 2).

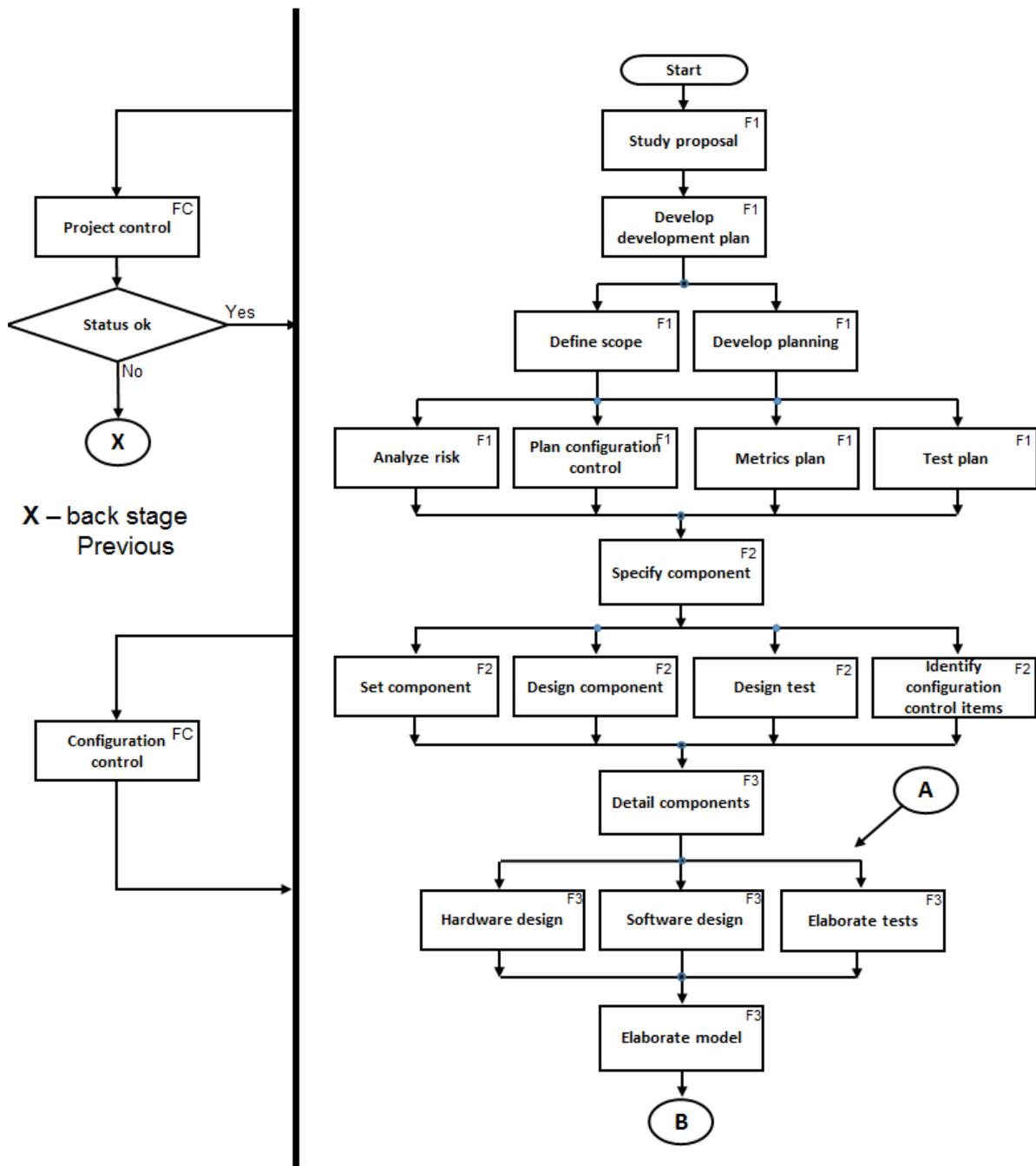


Figure 4.2 - Conduct of the proceedings of general flowchart - part-1.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

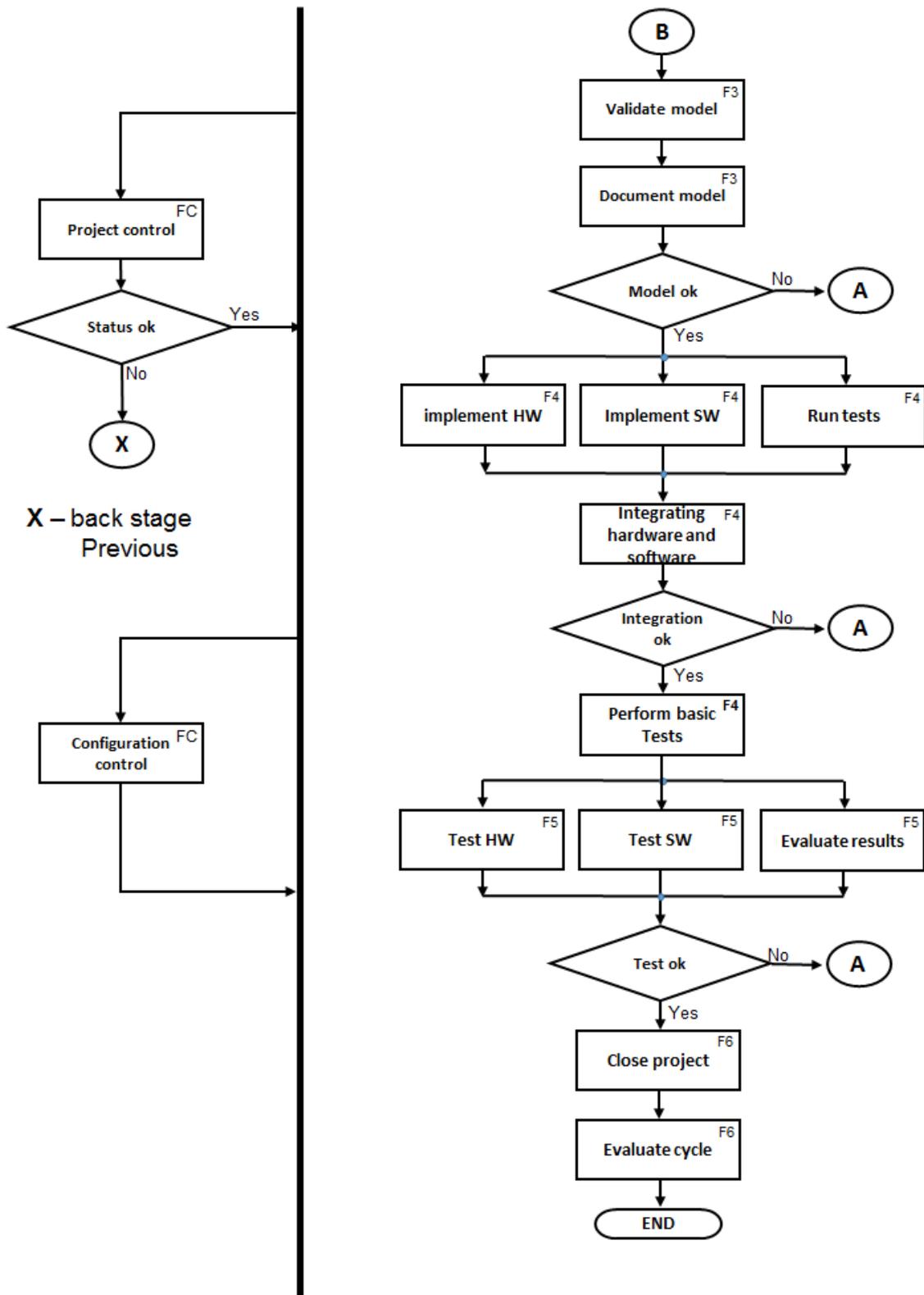


Figure 4.3 - Conduct of the proceedings of general flowchart - part-2.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3. Flowchart of each phase process

4.3.1. Guidelines for the phases of the flowchart of understanding

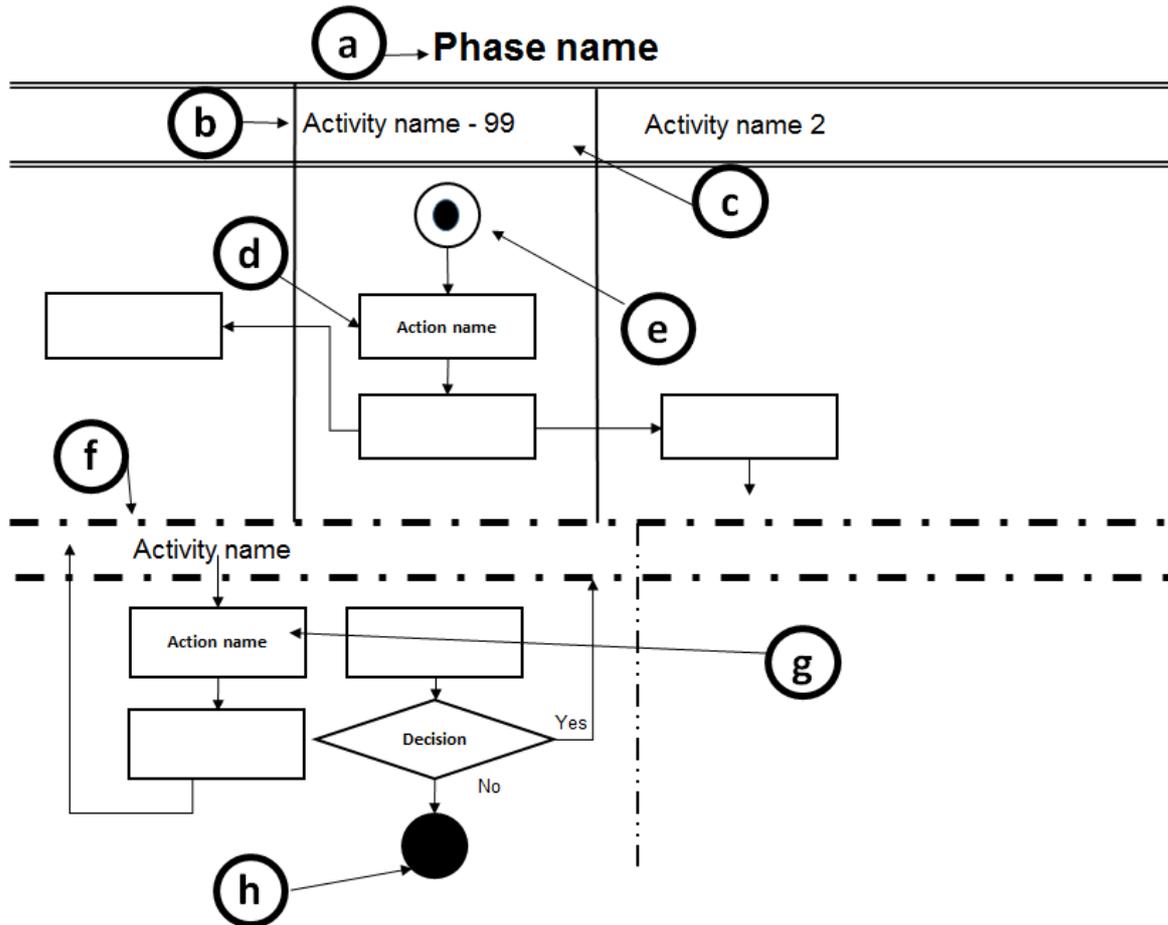


Figure 4.4 - Reports to flowchart of phases.

- a) Phase name refers to the flow.
- b) Ray where he held the shares in each activity.
- c) Sequential number of the activity in the process.
- d) The action to be performed.
- e) The activities located in these rays are activities that take place as a result of all actions of the activities of rays above.
- f) Start of flow.
- g) Action to take as a result of all other shares of the activities of rays above.

The flow charts of activities and their implementation does not require special guidelines.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.2. Flowchart of engineering phase - F1

This suggests flowchart generally the main activities of phase.

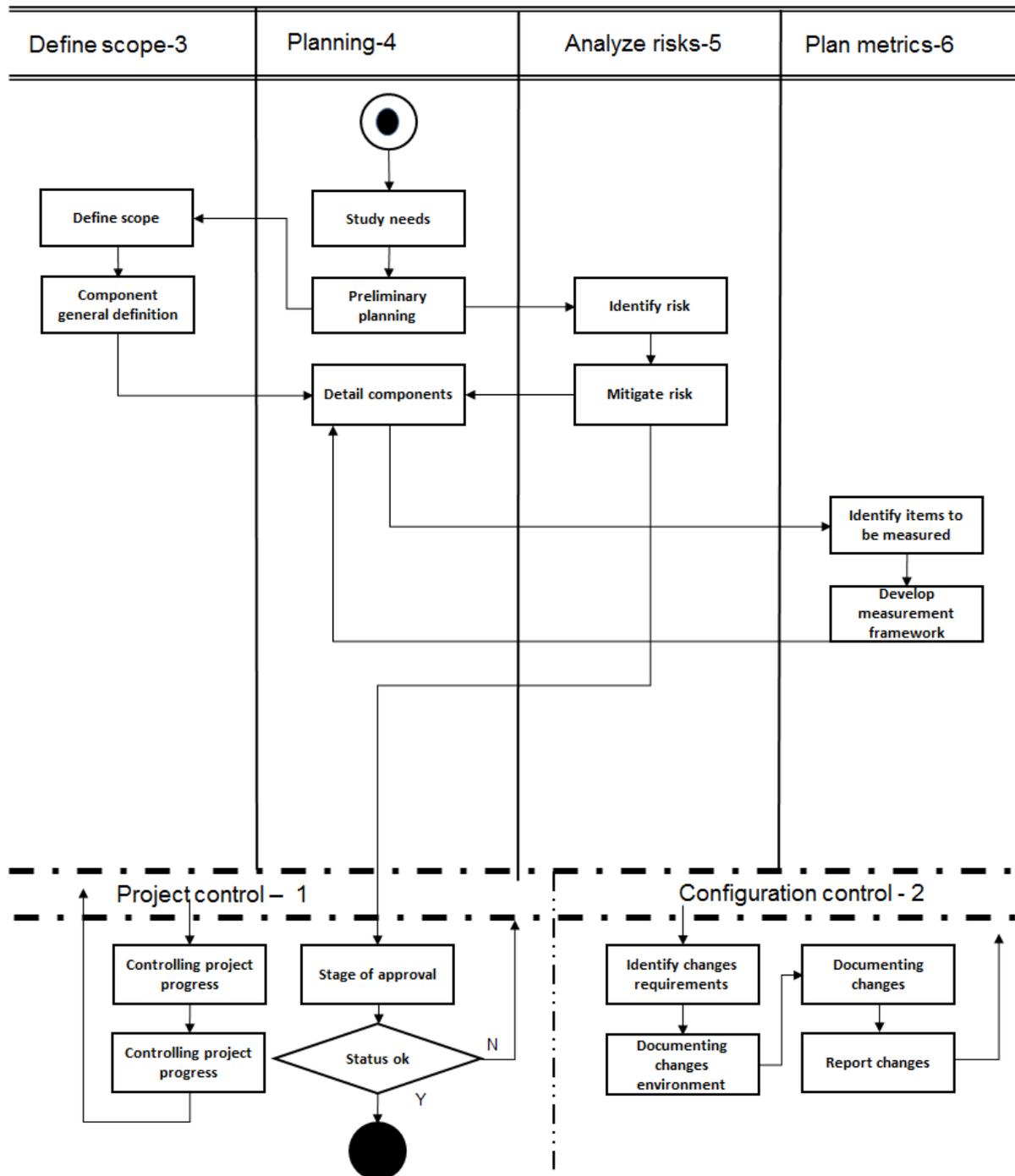


Figure 4.5 – Engineering.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.2.1. Flowchart of project control activity - 1

The activities of this flowchart are designed to perform control of the project at all stages.

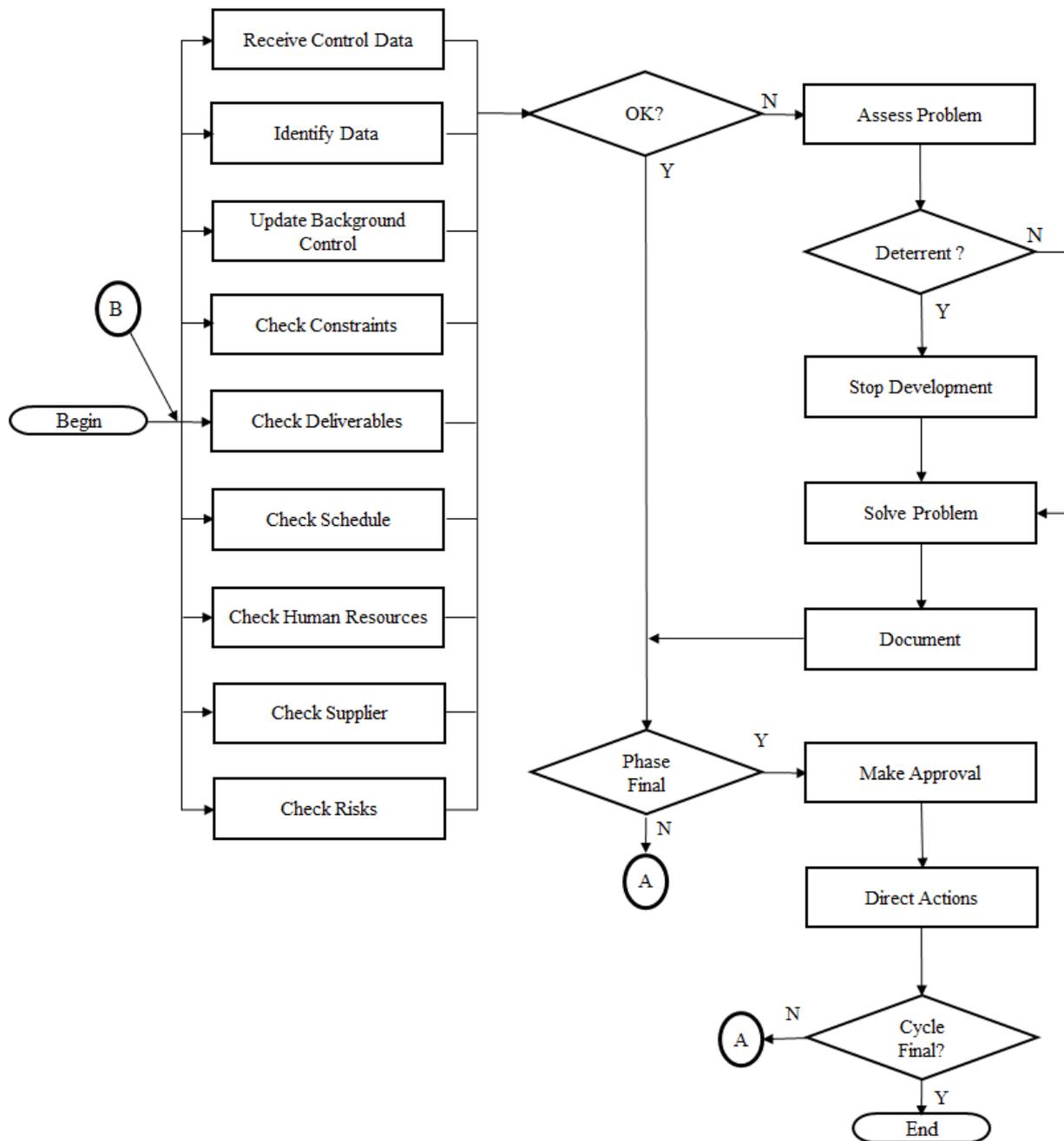


Figure 4.6 - Project control.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.2.2. Flowchart configuration control activity - 2

The activities of this flowchart are designed to perform configuration control and version changes of all components of the project being carried out at all stages.

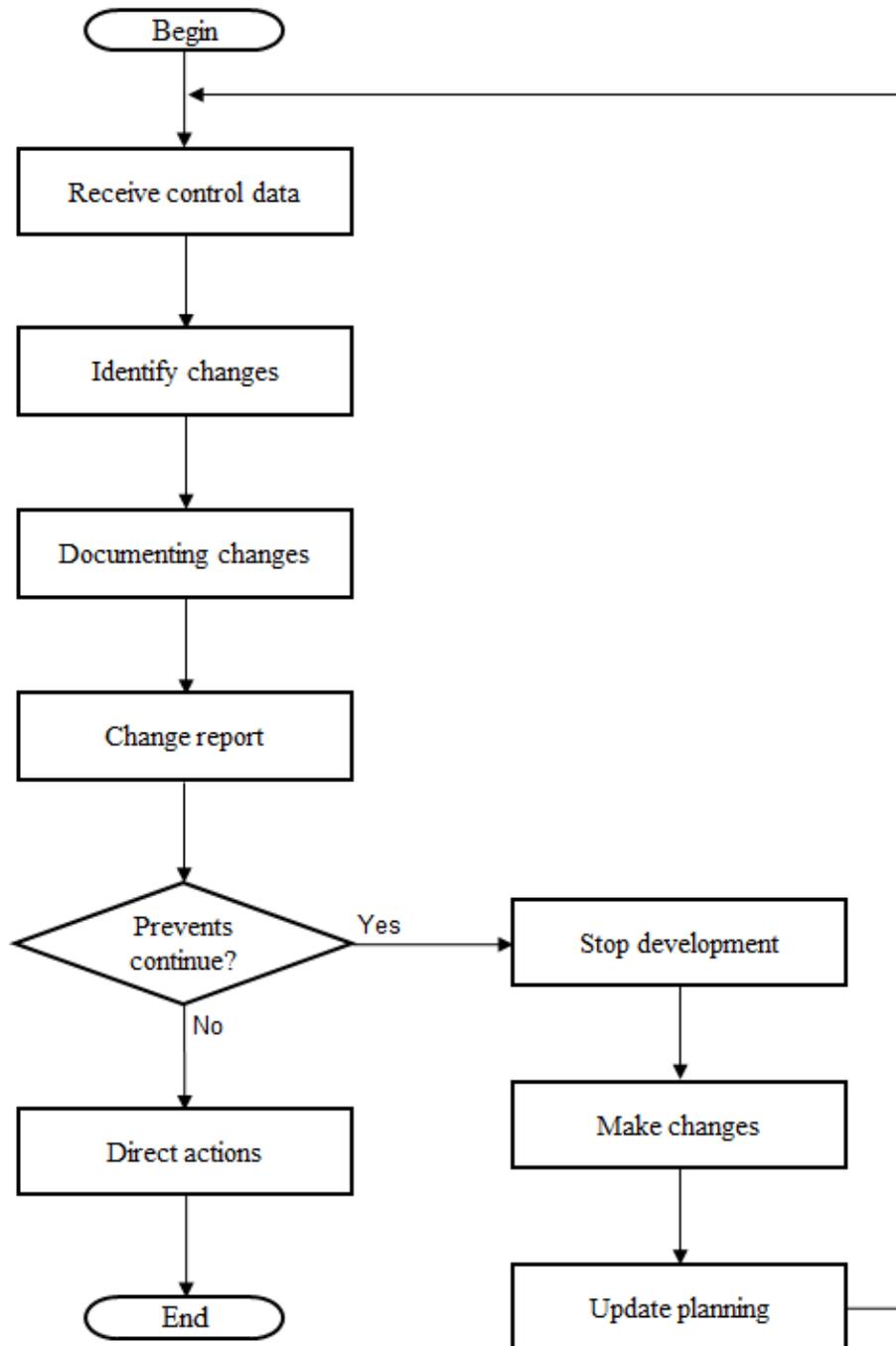


Figure 4.7 - Configuration control - 2.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.2.3. Flowchart activity set scope - 3

The activities of this flowchart are designed to perform the component scope definition to be produced, and their integration into complete product.

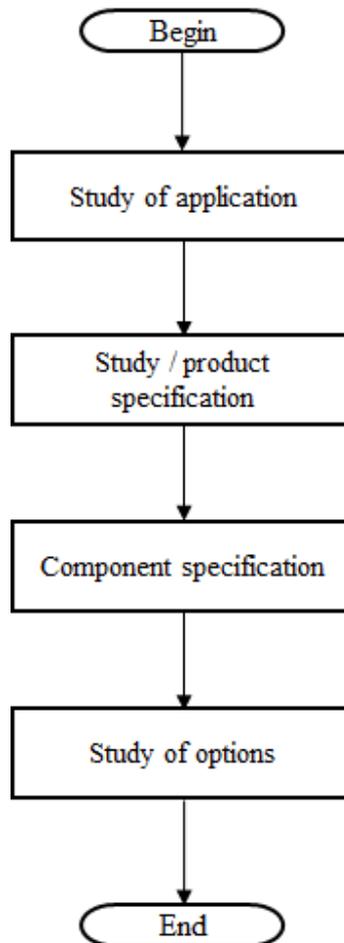


Figure 4.8 - Define scope-3.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.2.4. Flowchart activity planning - 4

The activities of this flowchart are designed to perform the component of project planning to be produced.

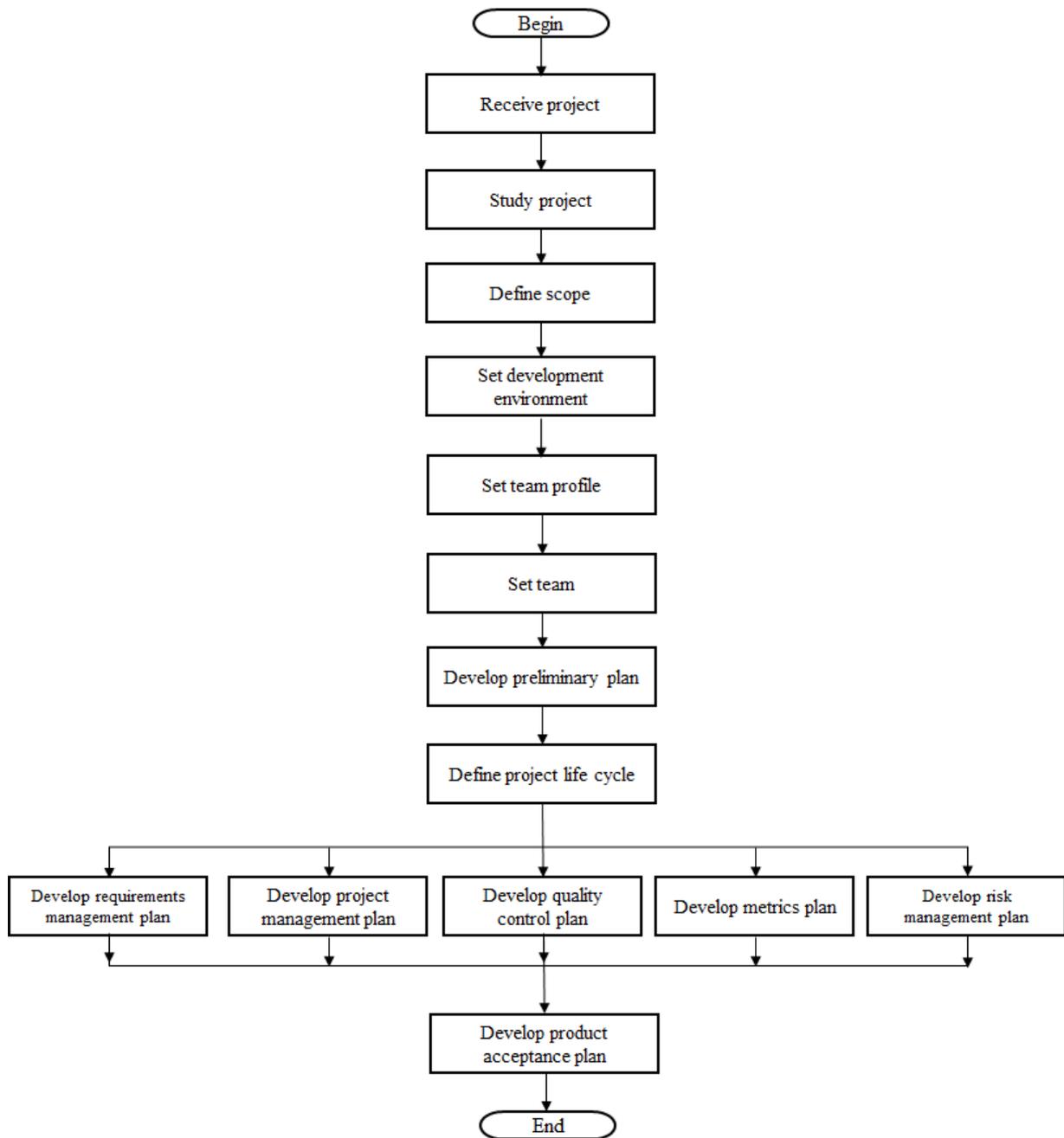


Figure 4.9 - Planning-4.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.2.5. Activity flowchart analyze risks - 5

The activities of this flowchart are designed to identify and prevent the possible risks of problems that can happen during the course of the Project.

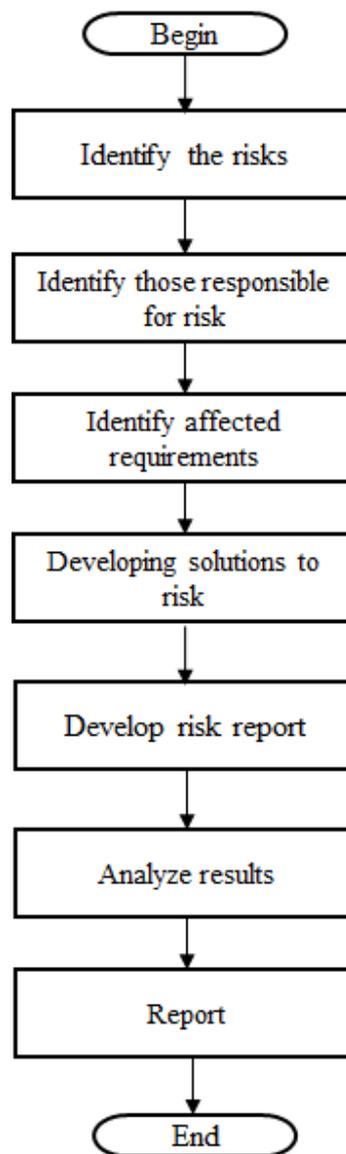


Figure 4.10 - Analyze risks-5.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

4.3.2.6. Flowchart activity plan metrics - 6

The activities of this flowchart are designed to perform the measurement of the components to be measured and monitoring during the course of the project. Control items are planned and can component items to be produced and also items (artifacts) of the process.

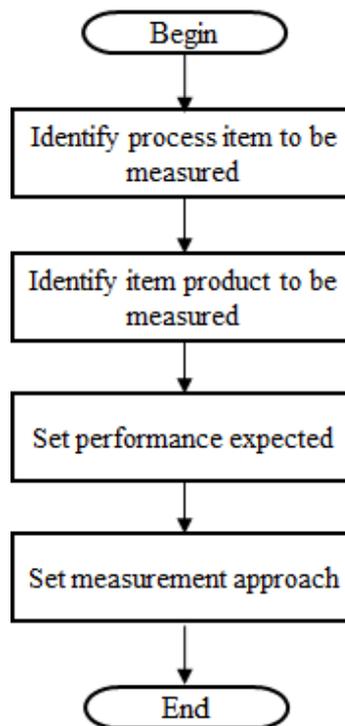


Figure 4.11 - Plan metrics-6.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

4.3.3. Flowchart requirement analysis phase – F2

This suggests flowchart generally the main activities of phase.

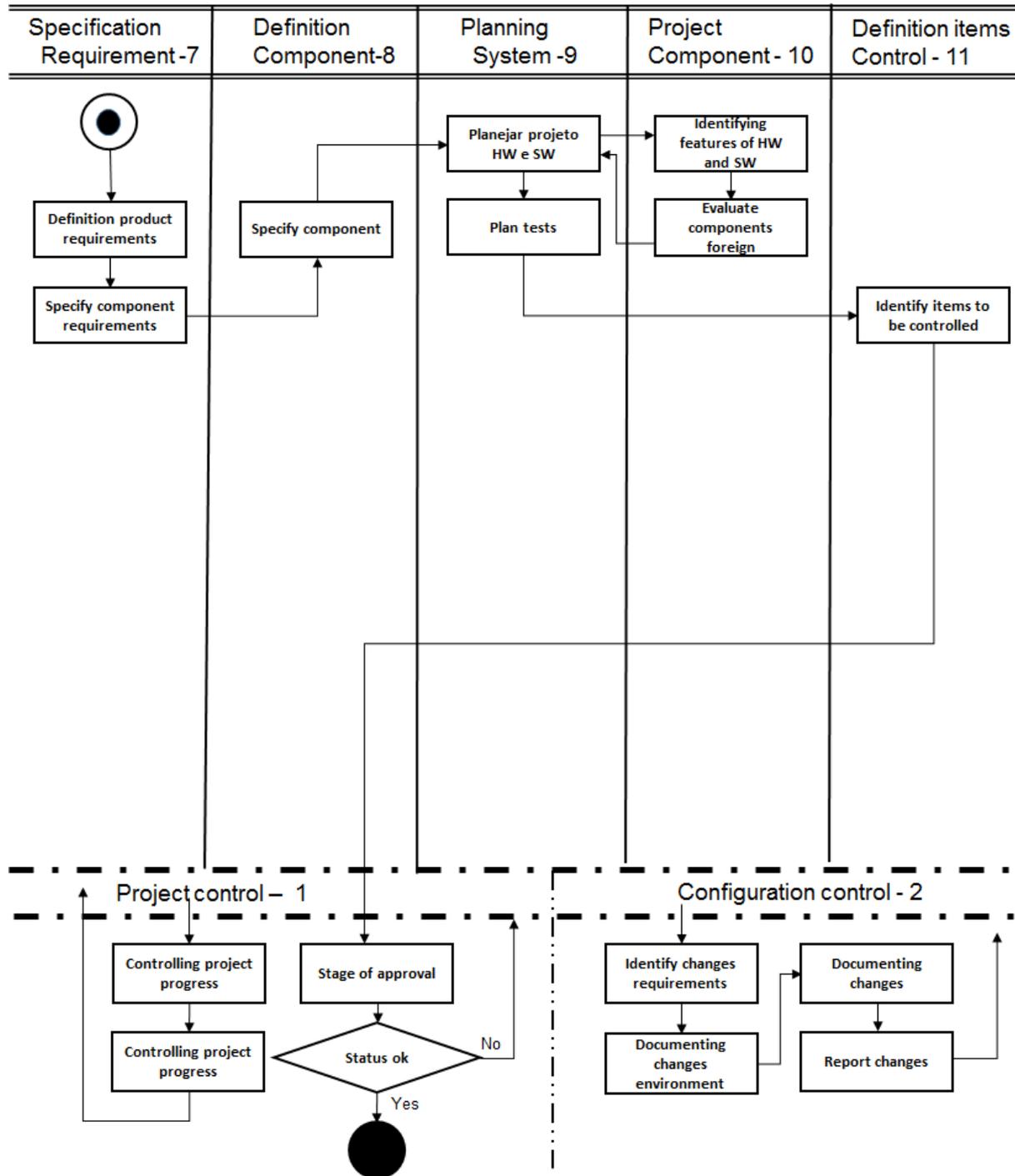


Figure 4.12 - Requirements analysis.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.3.1. Flowchart activity specification requirement - 7

The activities of this flowchart are designed to perform the component specification was produced to, and should include from understanding the full product to your requirements and identify the component requirements and their influence on the product.

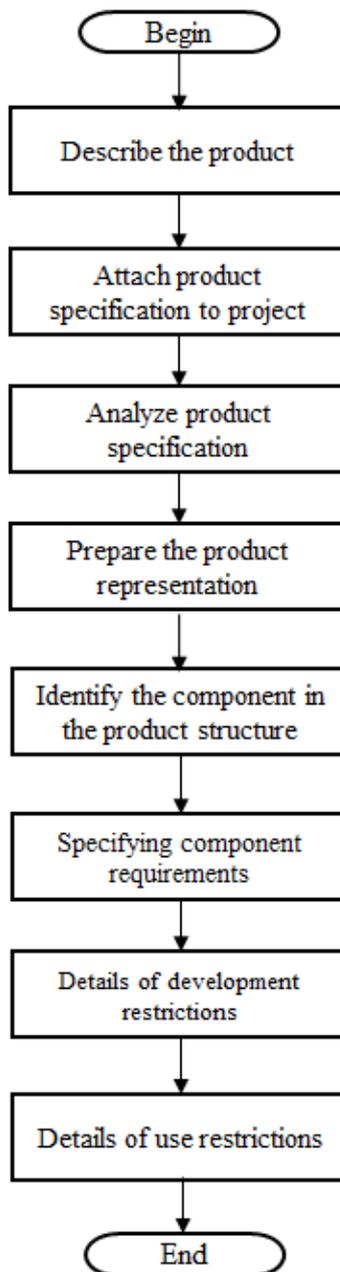


Figure 4.13 - Specification requirement-7.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

4.3.3.2. Flowchart setting activity component - 8

The activities of this flowchart are intended to specify in detail the component to be produced, incorporating its restrictions and given their special requirements.

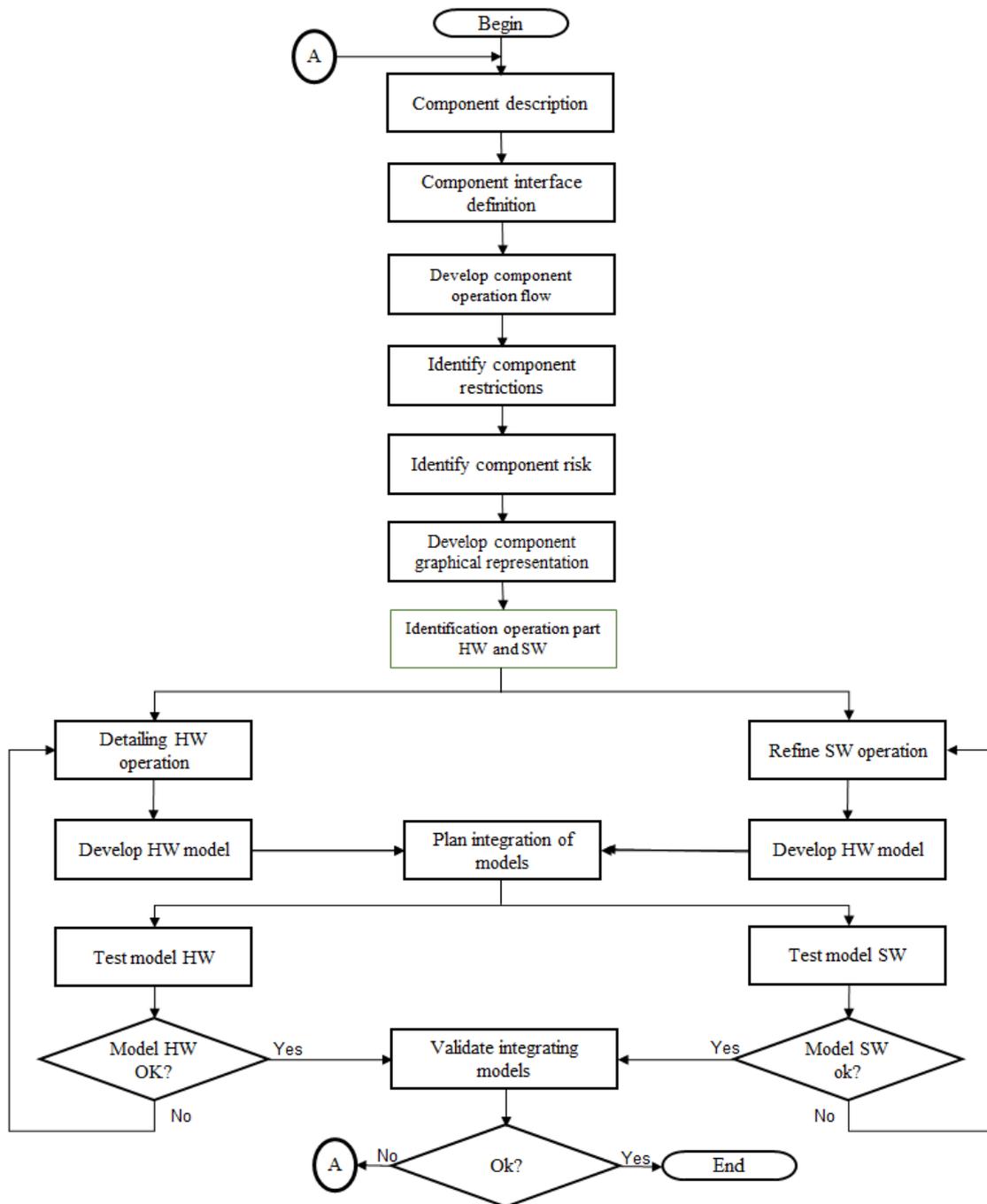


Figure 4.14 - Component Definition-8.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.3.3. Flowchart activity system planning - 9

The activities of this flowchart are designed to perform the component production and project planning at all stages.

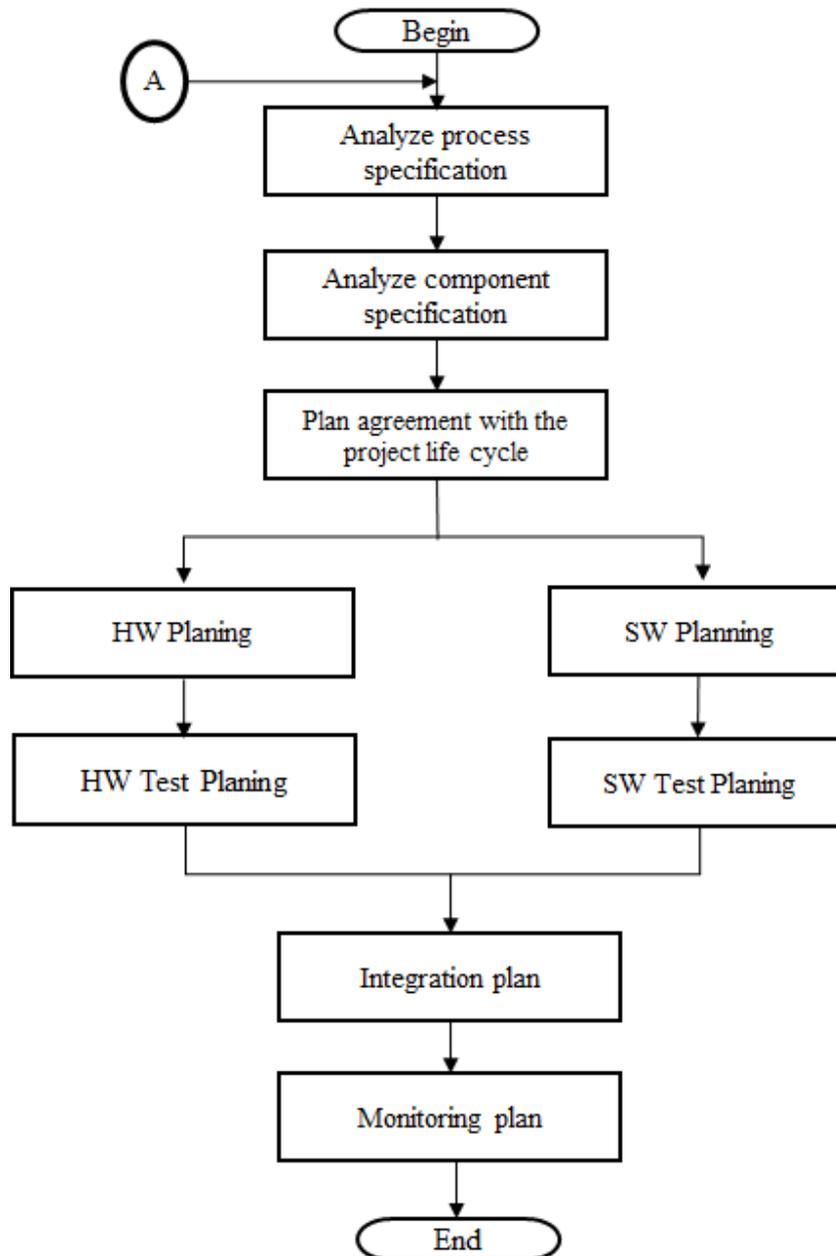


Figure 4.15 - System planning of-9.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.3.4. Flowchart component project activity - 10

The activities of this flowchart are intended prepare the detailed component design to production of the SW part and part HW.

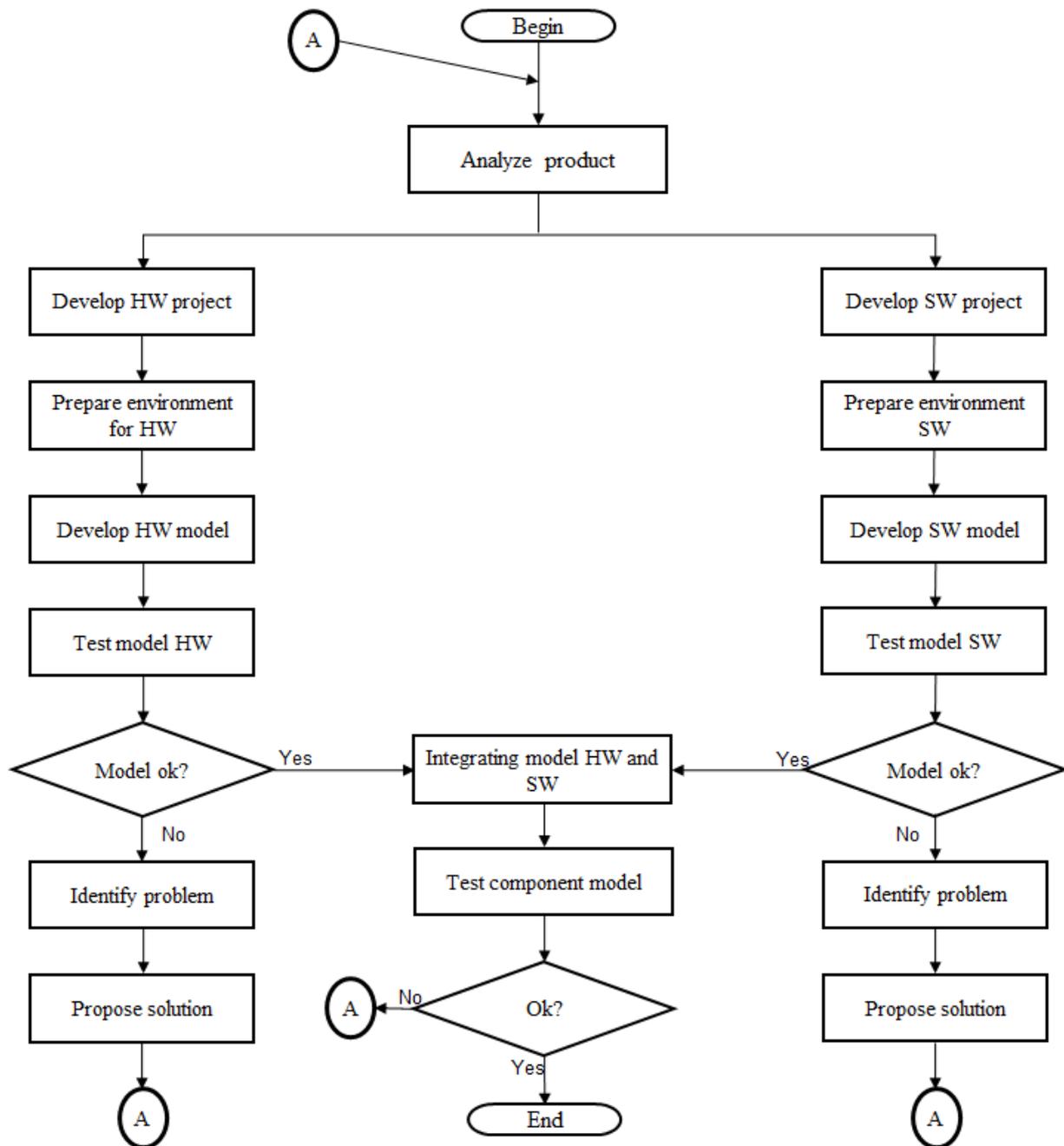


Figure 4.16 - Component design-10.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.3.5. Flowchart activity set control items - 11

The activities of this flowchart are intended to identify the items that should be controlled during the project, and should be considered the component items and artifacts produced as a result of the process.

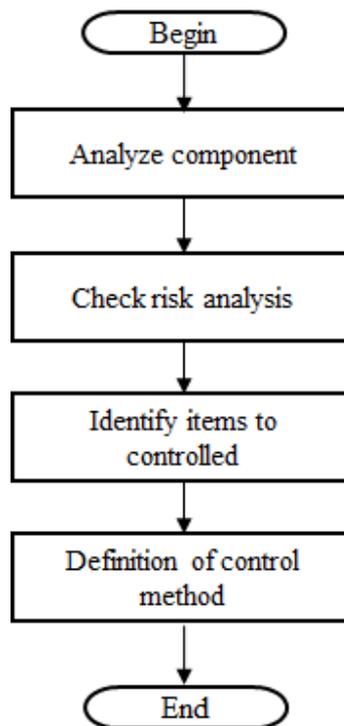


Figure 4.17 - Set control items-11.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4. Phase of the flow chart of analysis and design – F3

This suggests flowchart generally the main activities of phase.

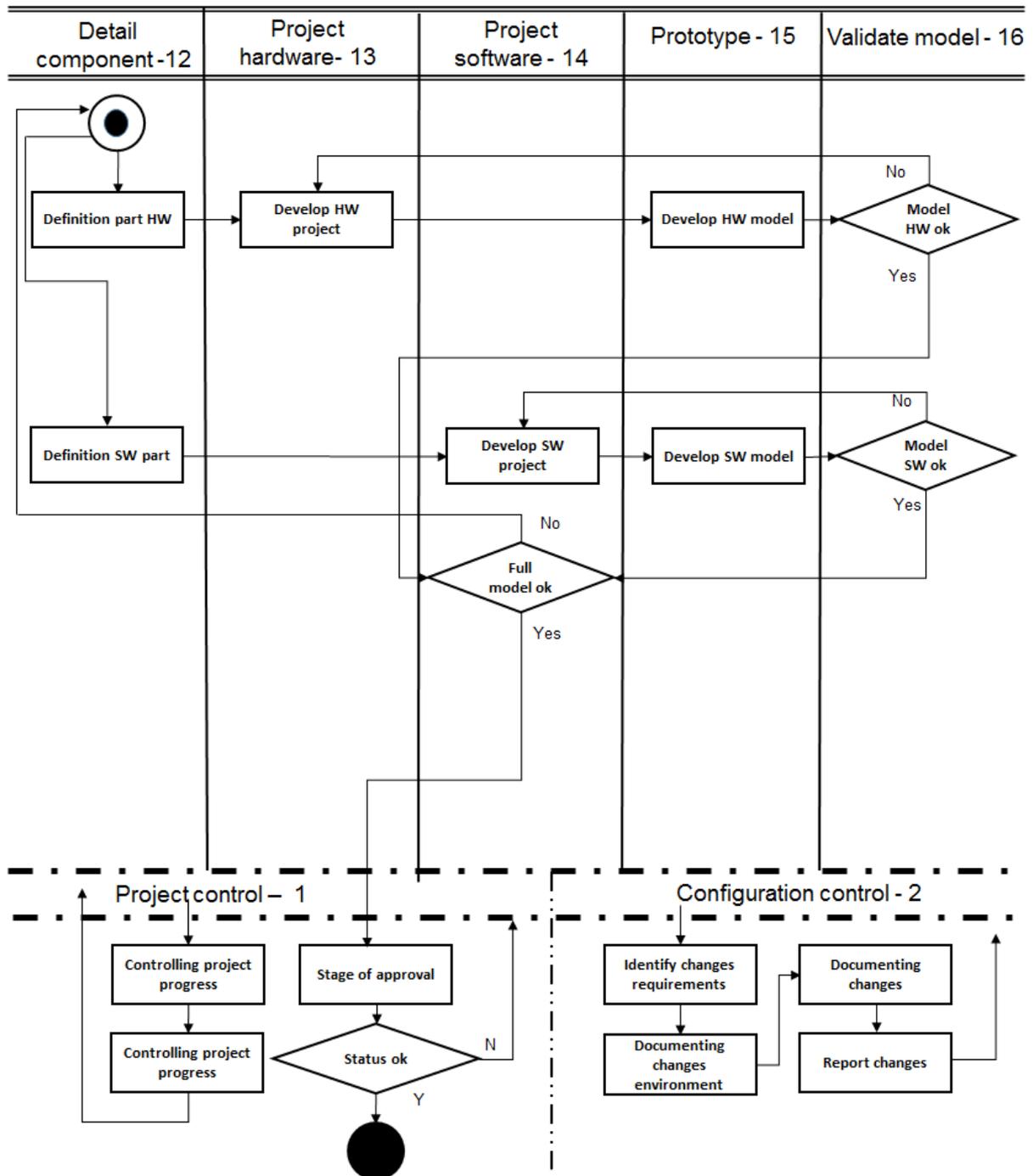


Figure 4.18 - Flowchart analysis and design phase.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.1. Flowchart detail component activity - 12

The activities of this flowchart are designed to detail the component production according to the model.

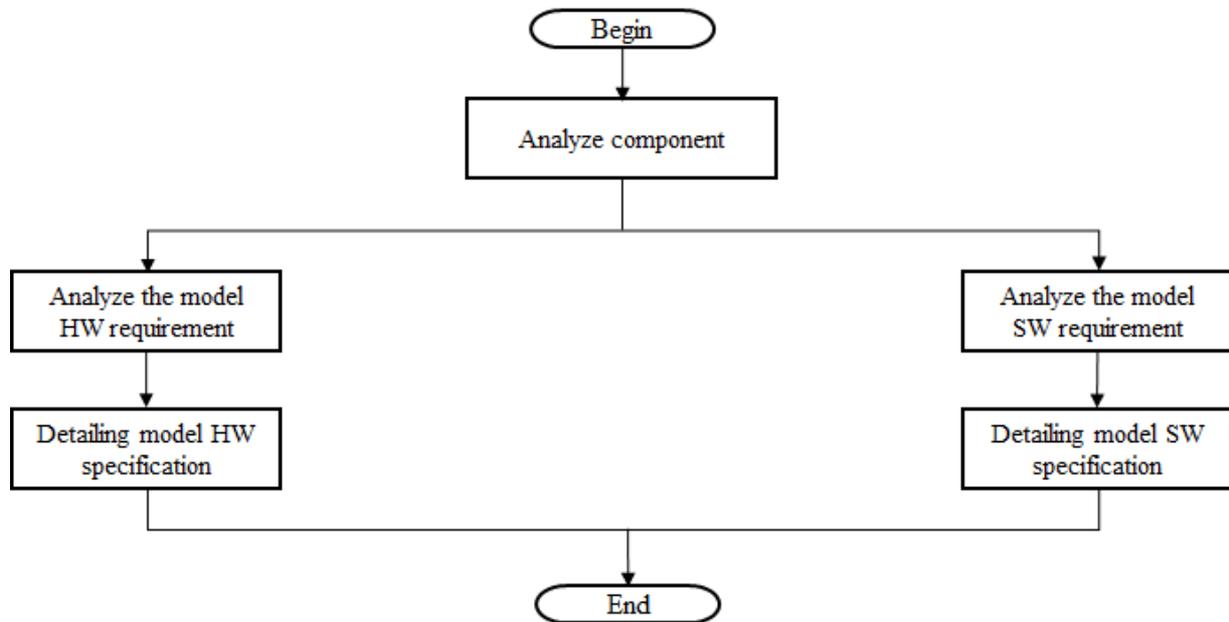


Figure 4.19 - Detail component-12.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.2. Flowchart of project activity hardware - 13

The activities of this flowchart are intended to design hardware models to be validated at the project level.

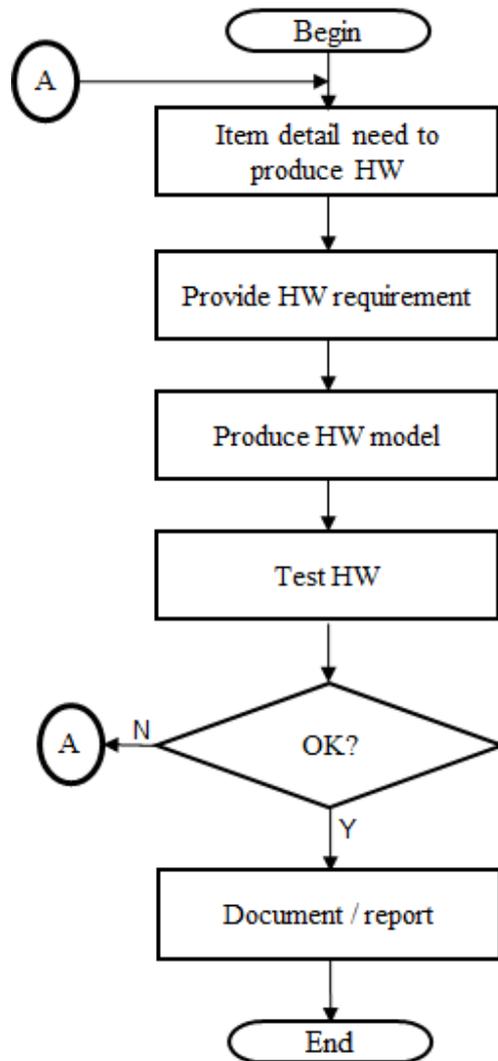


Figure 4.20 - Project hardware-13.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.3. Flowchart software project activity - 14

The activities of this flowchart are intended design software models to be validated at the project level.

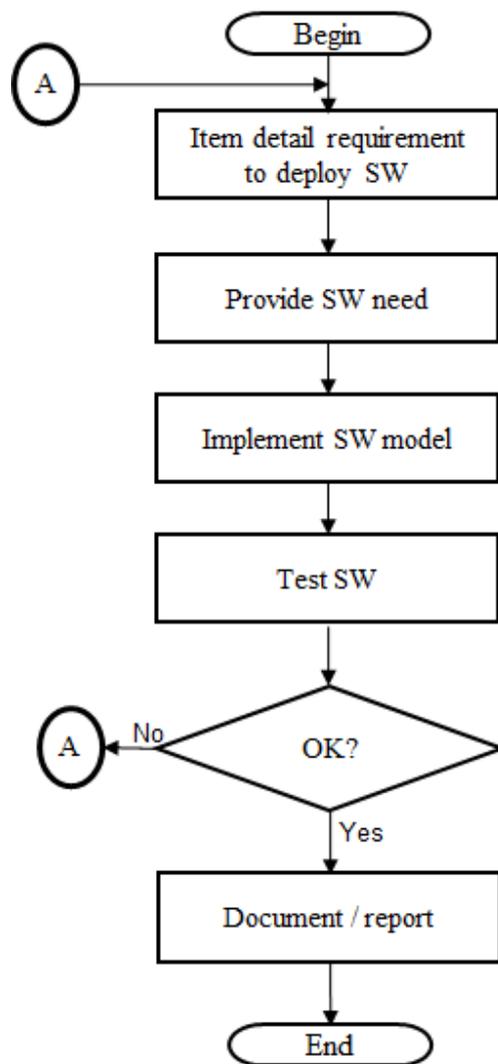


Figure 4.21 - Software design-14.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.4. Flowchart prototype activity - 15

The activities of this flowchart are intended to produce hardware and software prototypes and integrates them at the project level.

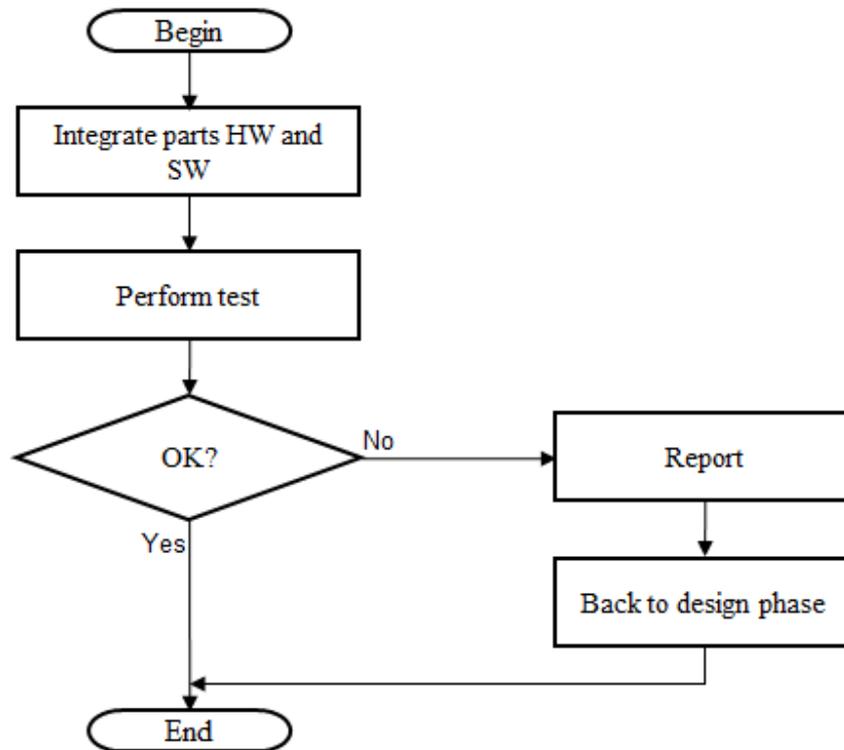


Figure 4.22 - Prototype-15.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.5. Flowchart activity validate model - 16

The activities of this flowchart are intended to validate the prototypes at the project level.

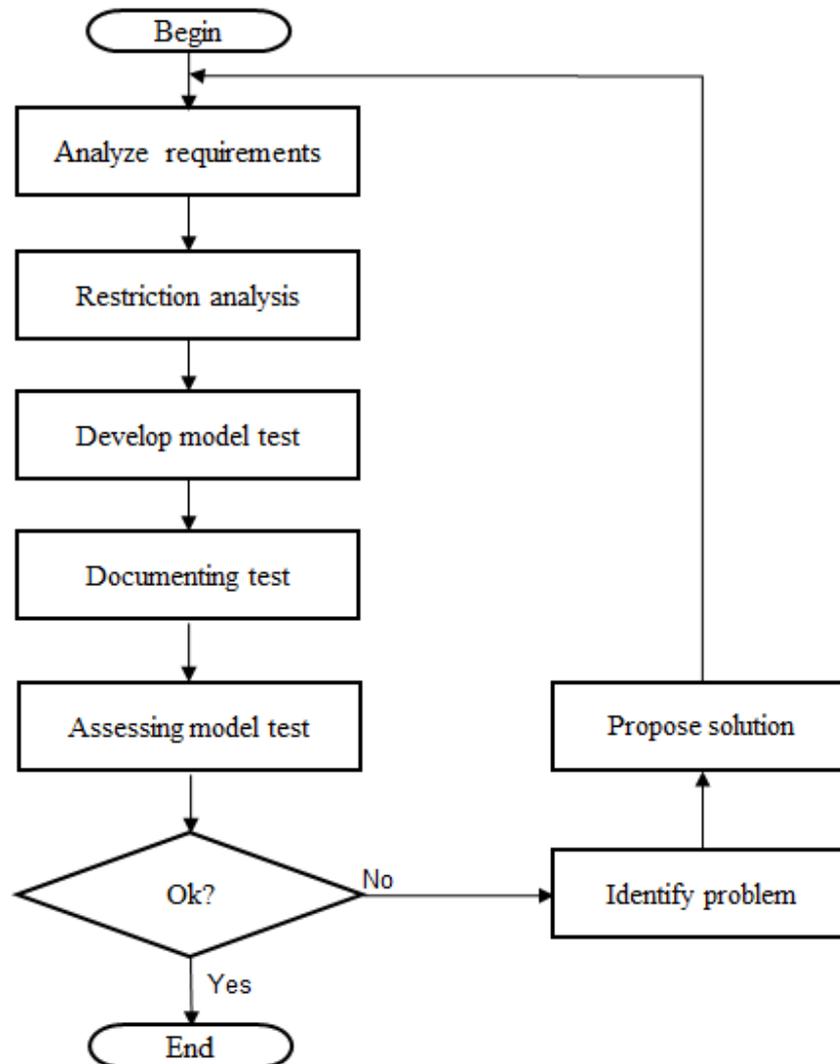


Figure 4.23 - Validate model-16.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

4.3.4.6. Flowchart of the implementation and integration phase – F4

This suggests flowchart generally the main activities of phase.

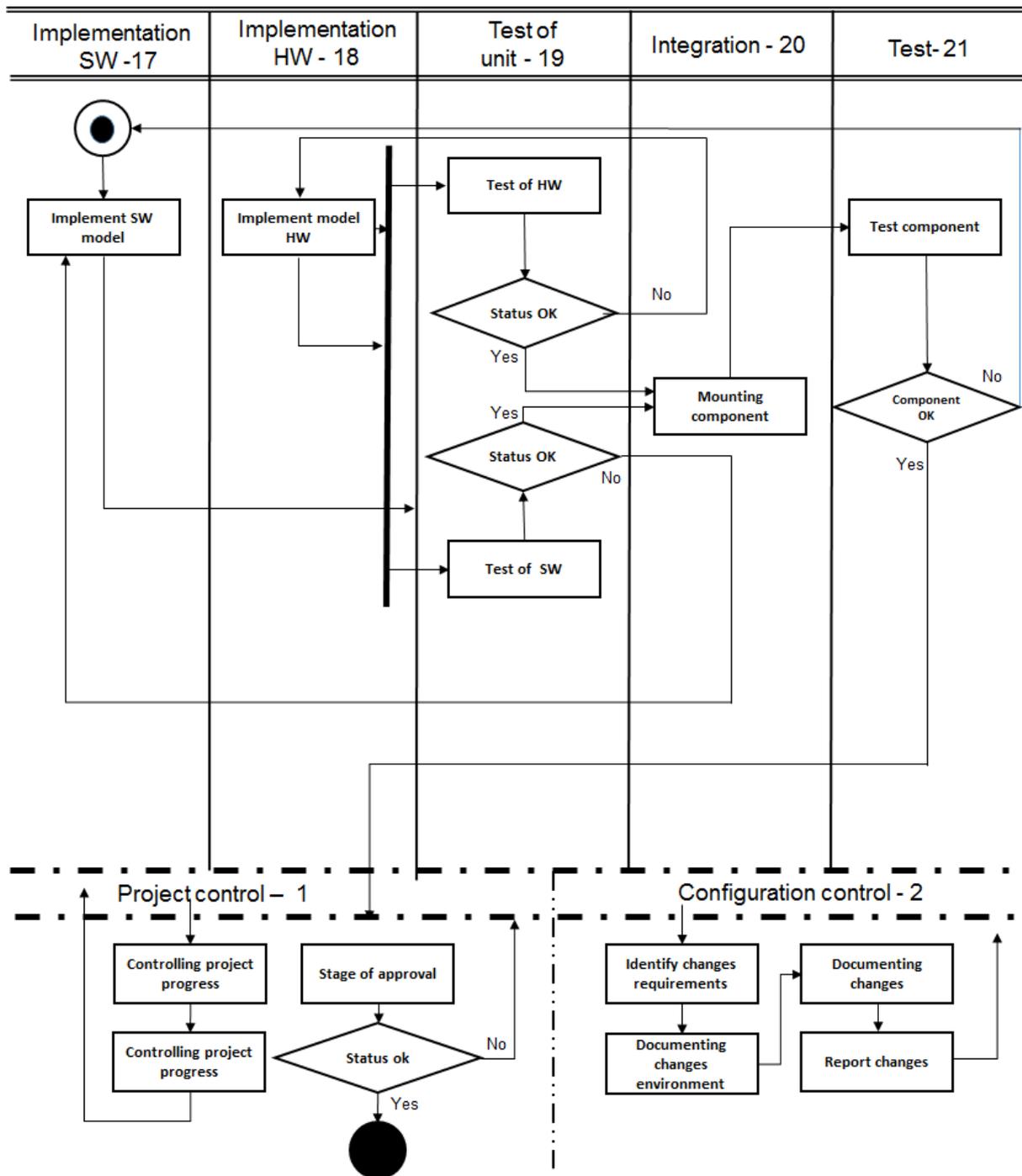


Figure 4.24 - Implementation and integration.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.7. Flowchart activity SW implementation -17

The activities of this flowchart are intended to implement the software according to the model and validated prototype.

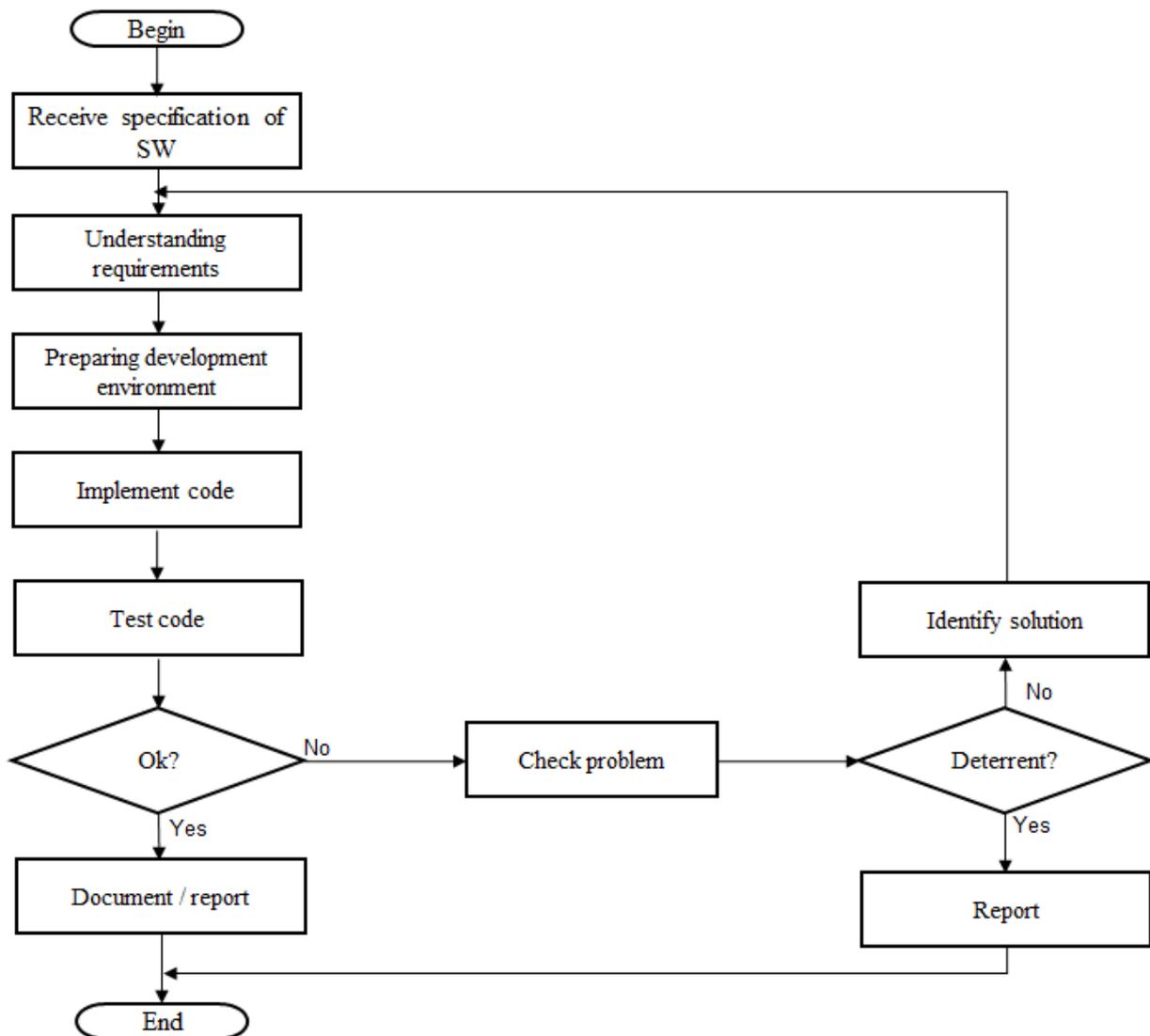


Figure 4.25 - SW Implementation-17.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.8. Flowchart of HW implementation activity - 18.

The activities of this flowchart are designed to produce the hardware according to the model and validated prototype

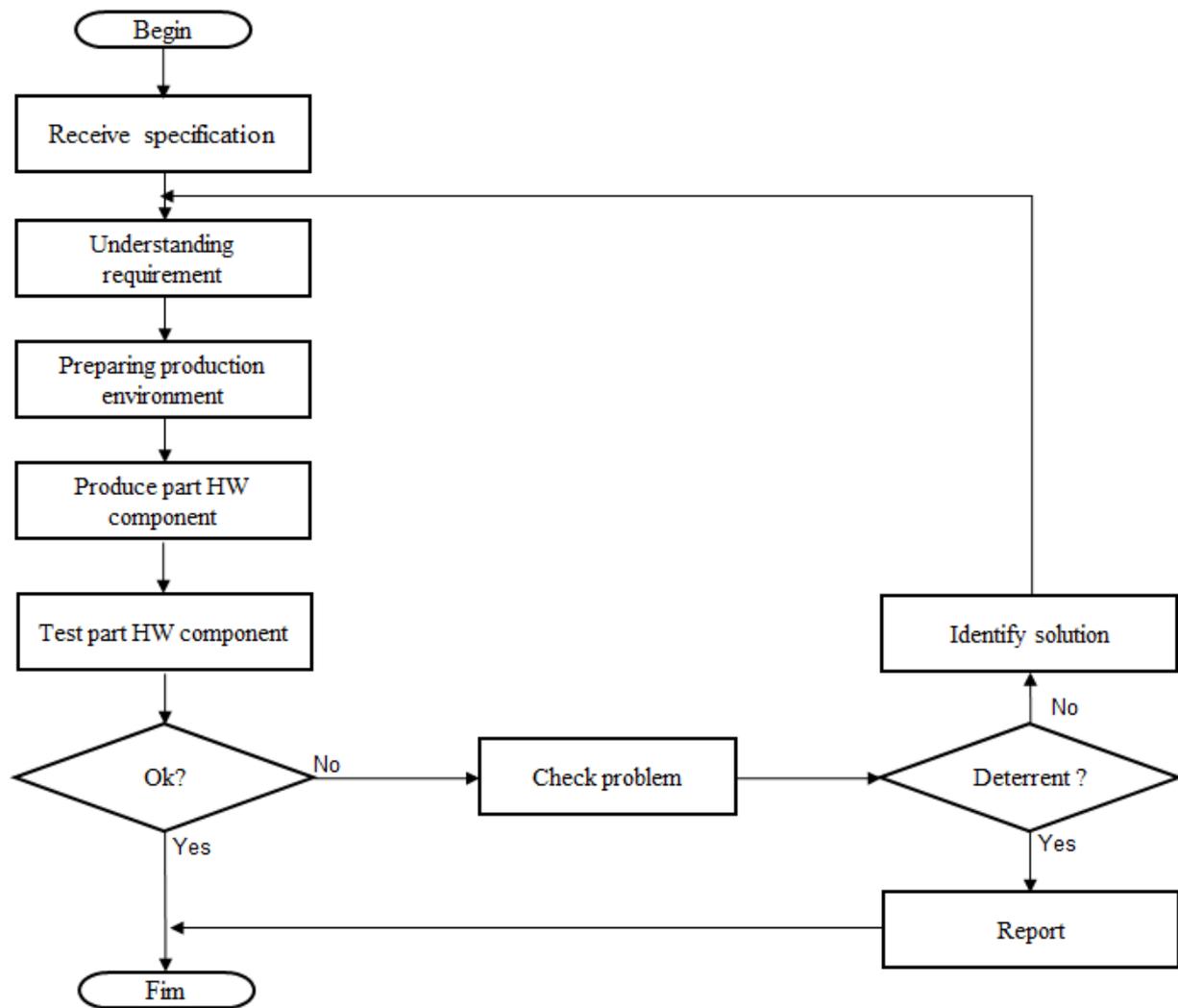


Figure 4.26 - HW Implementation-18.

4.3.4.9. Flowchart of unit testing activity - 19

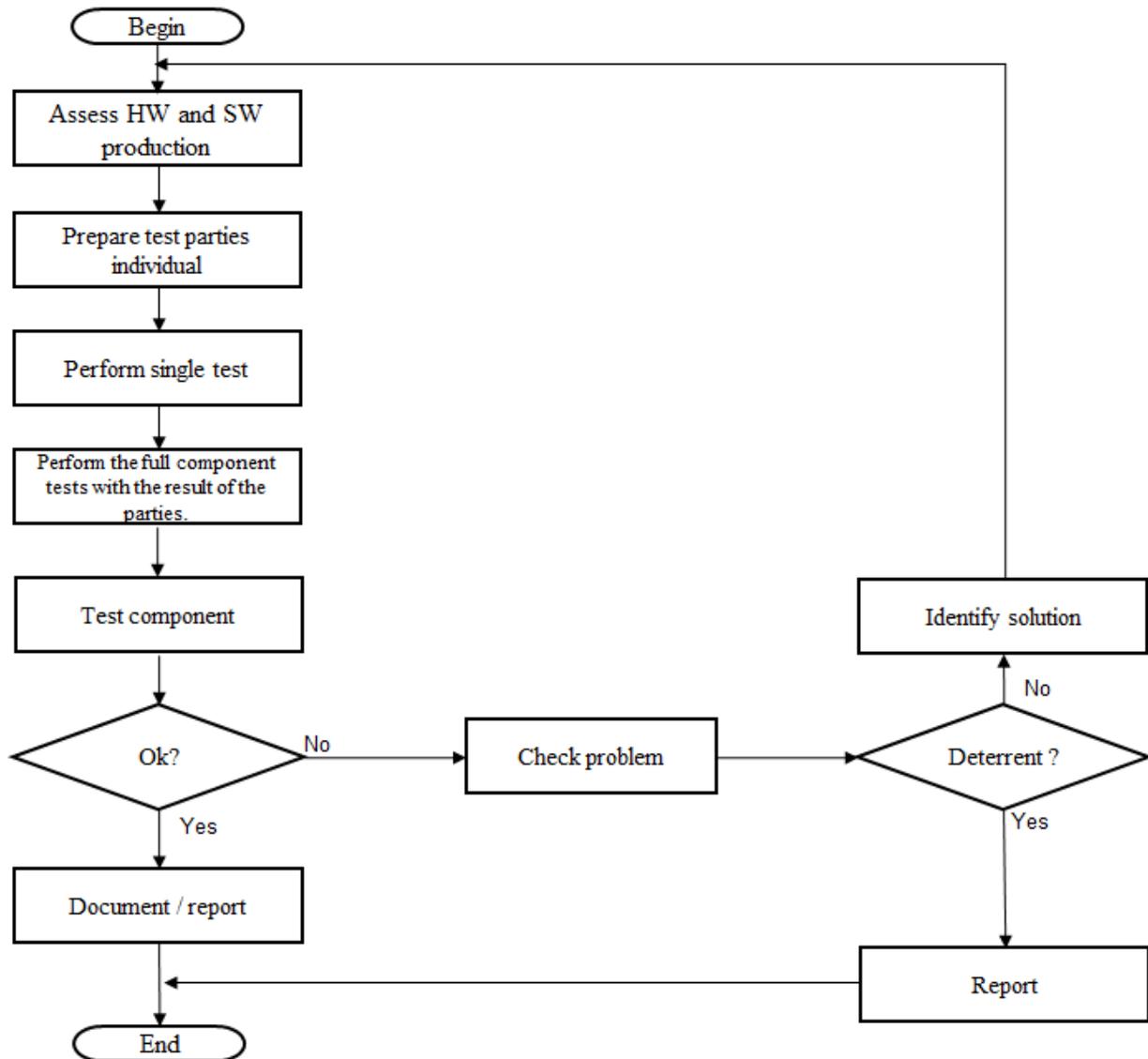


Figure 4.27 - Unit testing-19.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.10. Flowchart integration activity - 20

The activities of this flowchart are intended to lead the integration of software parts and component hardware. Tests conducted in this activity are the basic level of integration.

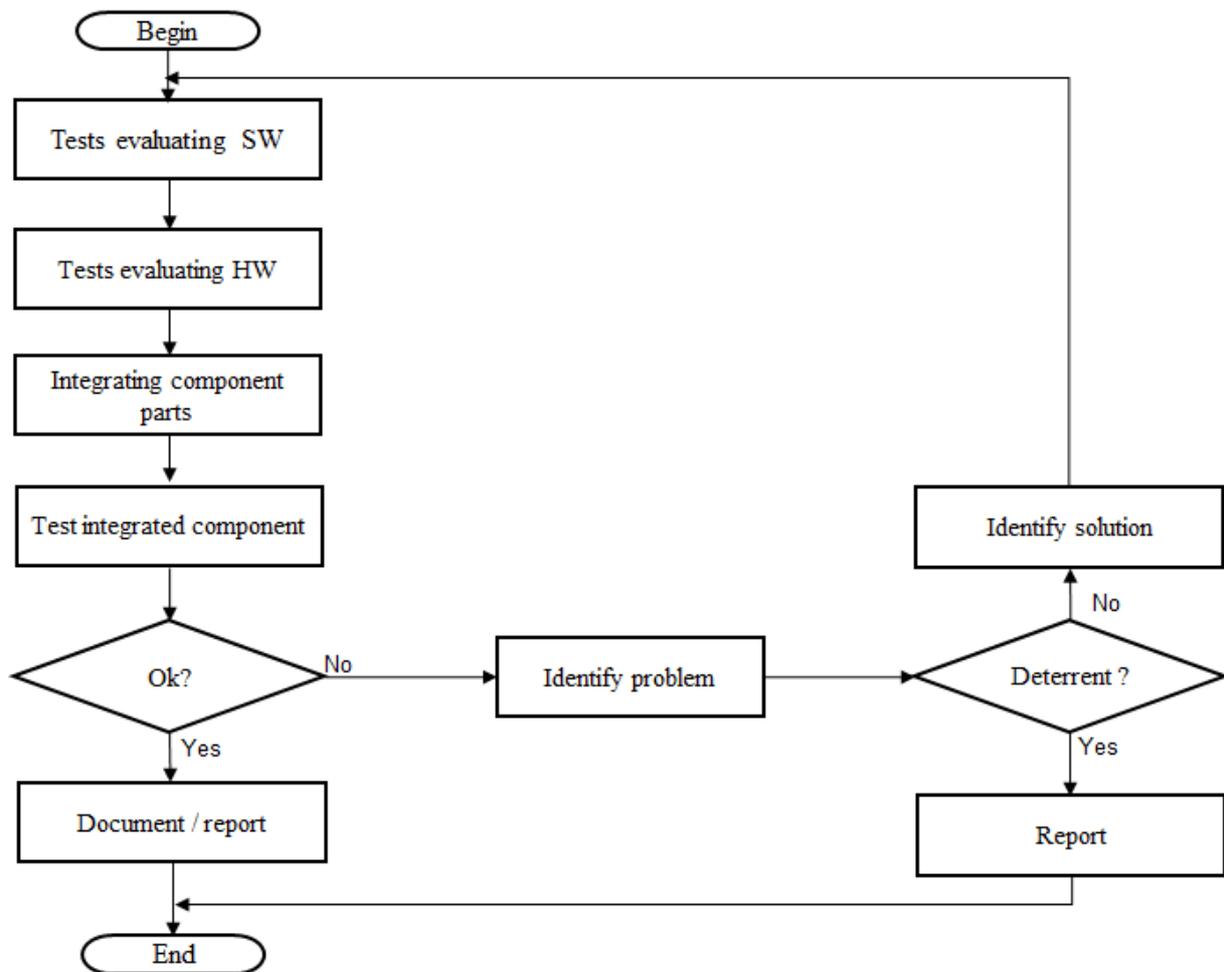


Figure 4.28 - Integration-20.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.4.11. Flowchart component test activity - 21

The activities of this flowchart are designed to perform tests party software, hardware and complete compromises the level of communication between the parties

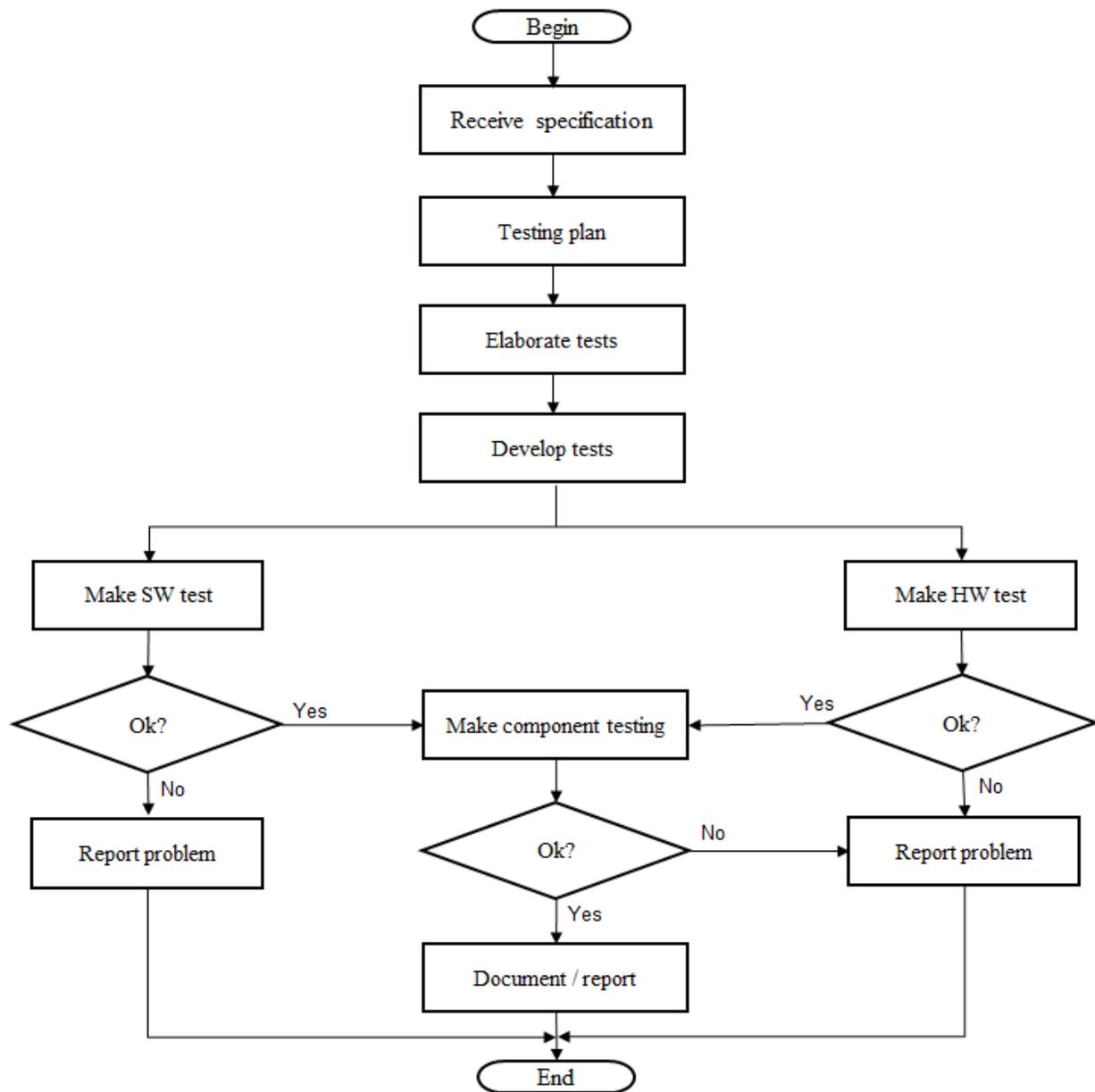


Figure 4.29 - Component test-21.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.5. Flowchart for verification and validation phase – F5

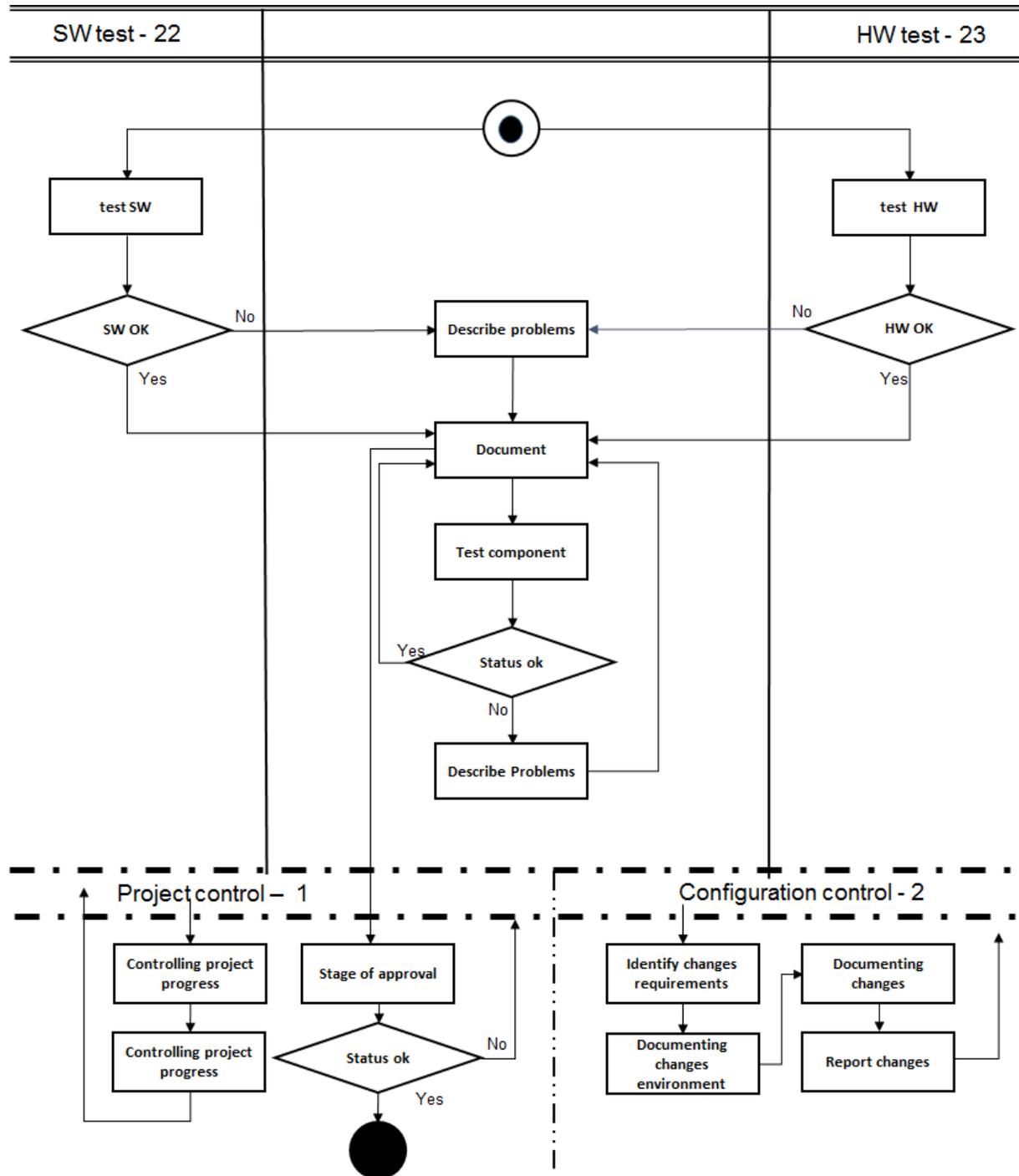


Figure 4.30 - Flowchart of verification and validation phase.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.5.1. Flowchart activity SW test - 22

The activities of this flowchart are designed to perform the test on the part of the software component to component level and their inclusion in the product.

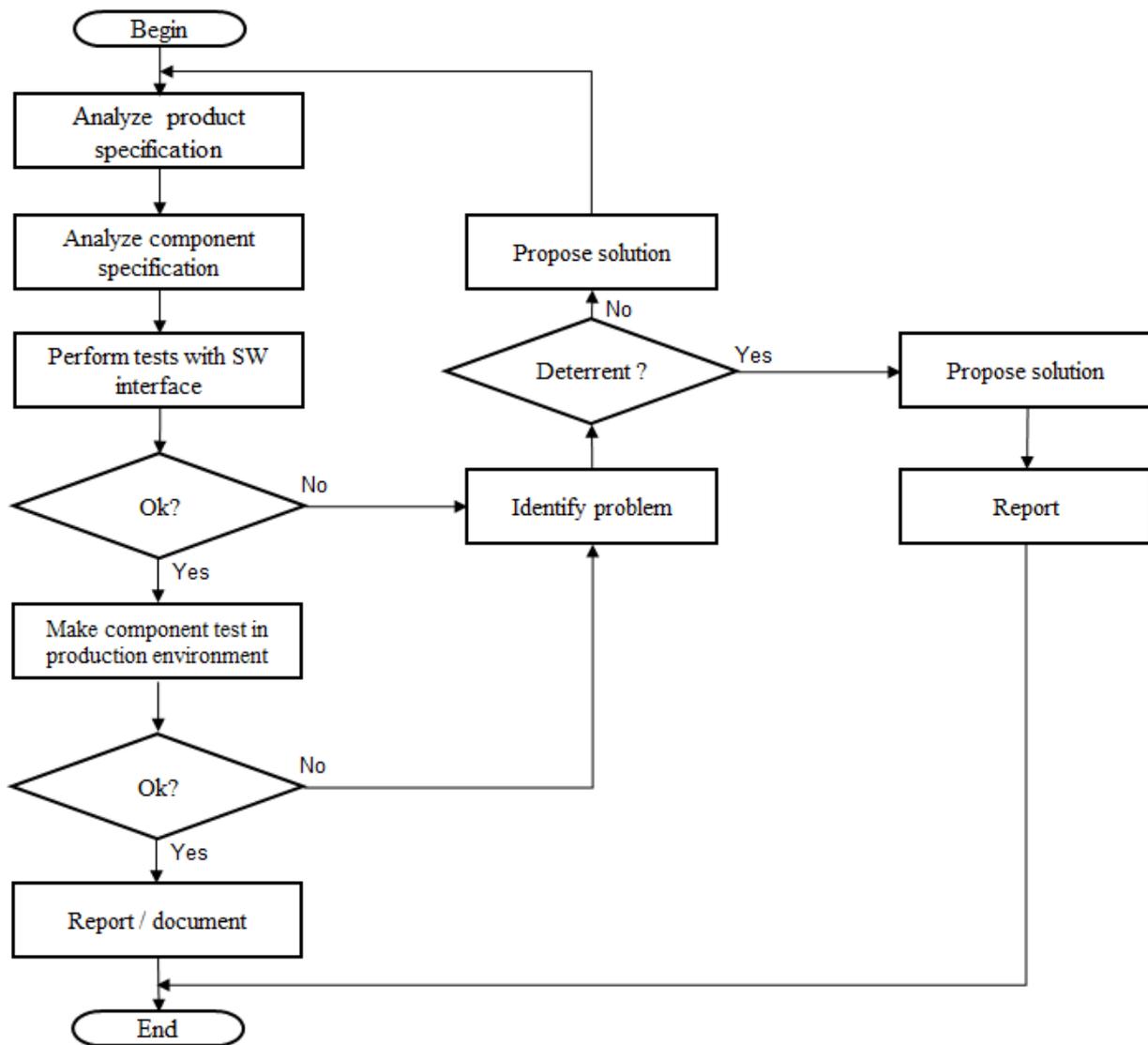


Figure 4.31 - SW Test-22.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.5.2. Flowchart of HW test activity - 23

The activities of this flowchart are designed to make the hardware component of the test the component level and their inclusion in the product

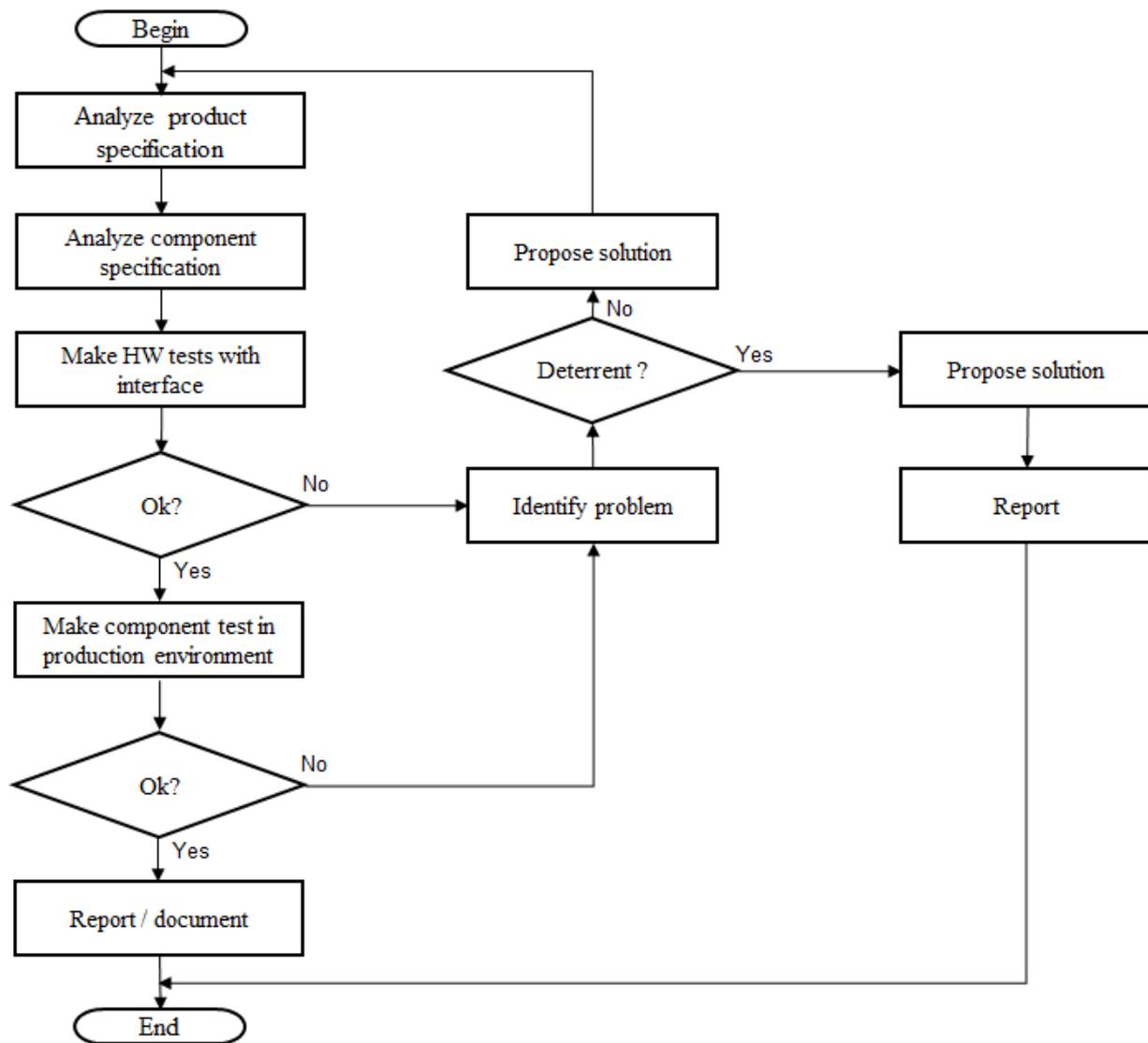


Figure 4.32 - HW test-23.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.6. Phase Flow chart cycle assessment phase – F6

This suggests flowchart generally the main activities of phase.

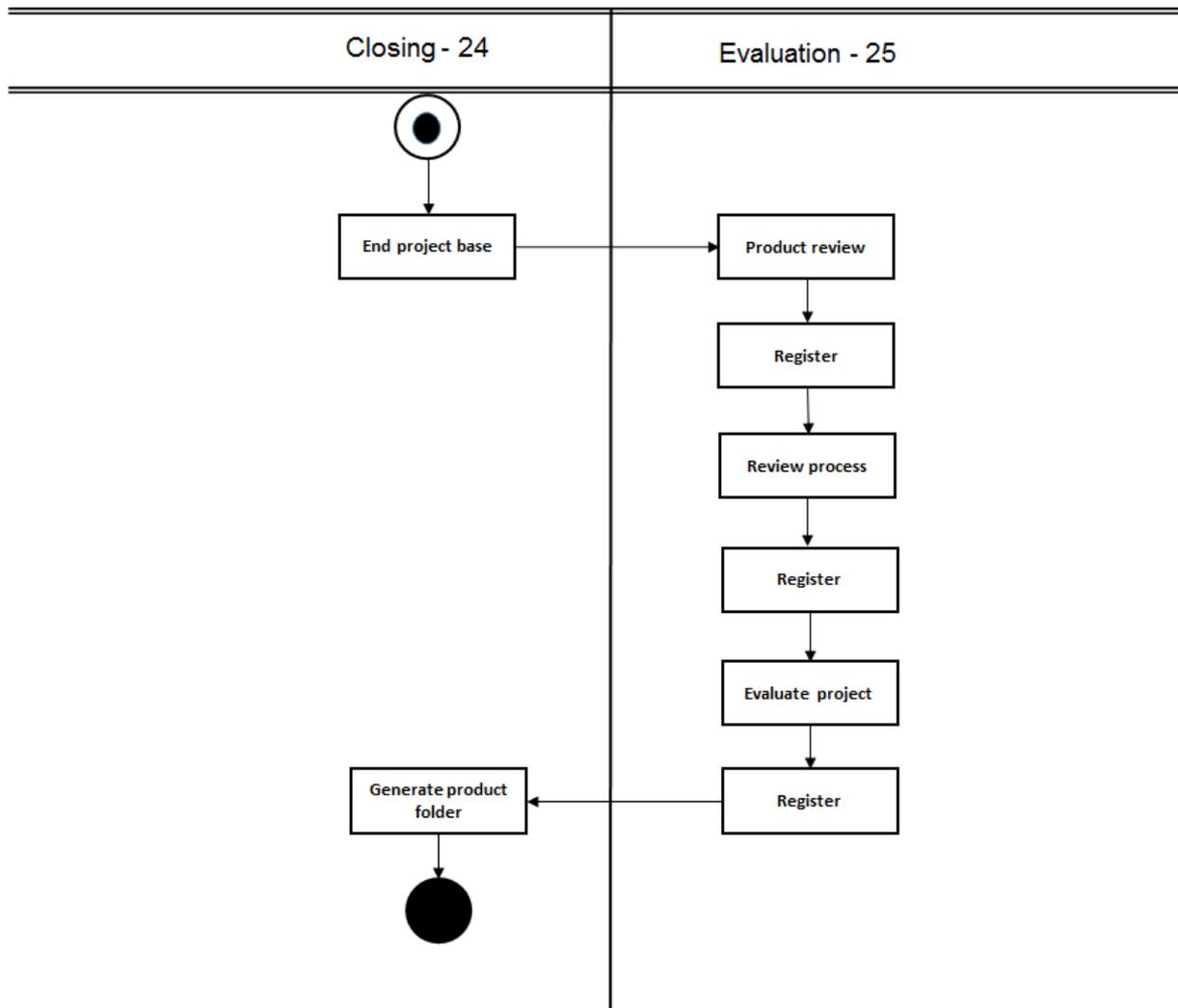


Figure 4.33 - Cycle assessment.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.6.1. Flowchart closing activity - 24

The activities of this flowchart are designed to close the project and develop the product package.

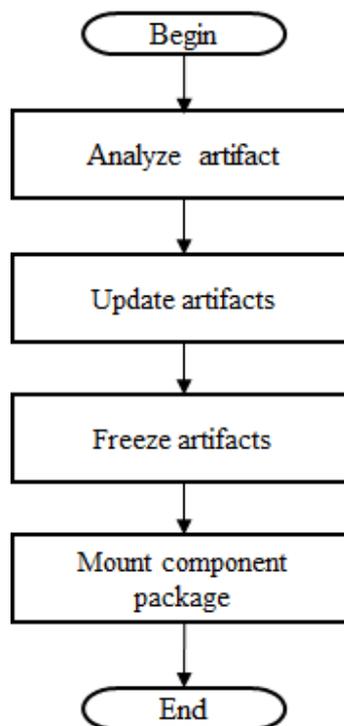


Figure 4.34 - Closing-24.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded Laboratory	Date:

4.3.6.2. Flowchart activity evaluation - 25

The activities of this flowchart have to evaluate the project and produced product and record results and suggestions for improvements for next cycle.

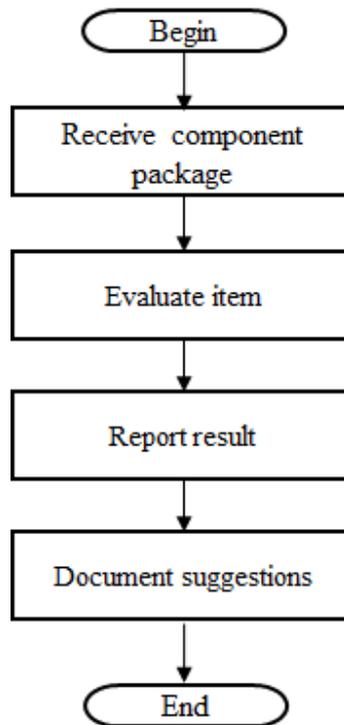


Figure 4.35 - Evaluation-25.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

5. Documents using standard.

5.1. Document property.

Must be filled in the fields of ownership of the documents so that it can be automatically loaded in the document body, as shown below.

The information in these fields will be transported to the Word document.

Document properties must be completed as described.

The screenshot shows a dialog box titled "Propriedades de 0 - Configuration Management-MG01". It has several tabs: "Geral", "Resumo", "Estatísticas", "Conteúdo", and "Personalizar". The "Geral" tab is active. The fields are as follows:

- Título: <Component Name>
- Assunto: <Project Name>
- Autor: <Author Name>
- Gerente: CITAR – INPE
- Empresa: <Facility Name Partner>
- Categoria: Configuration Management
- Palavras-chave: Processo de Desenvolvimento de SE
- Comentários: This document is a result of the proposed Doctoral M.: Magda Ap Silverio Miyashiro at INPE
- Base do hiperlink: (empty)
- Modelo: Configuração
- Salvar Miniaturas de Todos os Documentos do Word

At the bottom, there are "OK" and "Cancelar" buttons.

Figure 5.1 - Word document properties (templates).

5.2. Auto complete document.

The models (templates) used in this process was created with Word template format.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

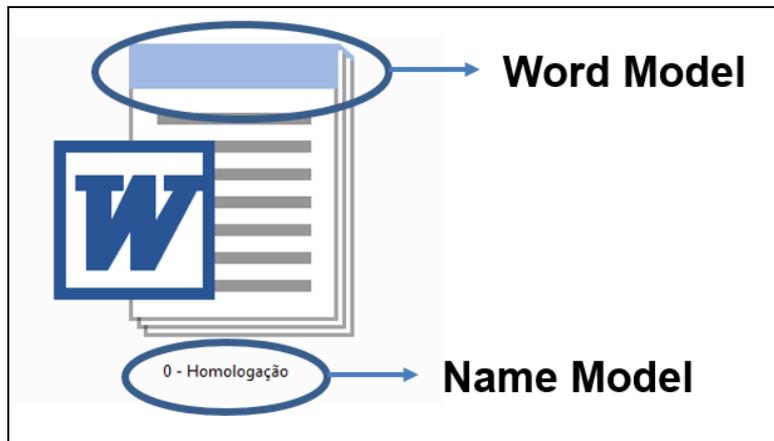


Figure 5.2 - Word template resource representation.

When the template file is edited opens a document image to the edited document (word feature) to be saved with the appropriate name to documents of the cycle, as the figure below item.

The document b and c items of the document are full of in information from documents properties.

To pull this information, after filling in the fields of document properties, select the entire document and access F9, which all document points that have automatic fields of the properties are loaded.

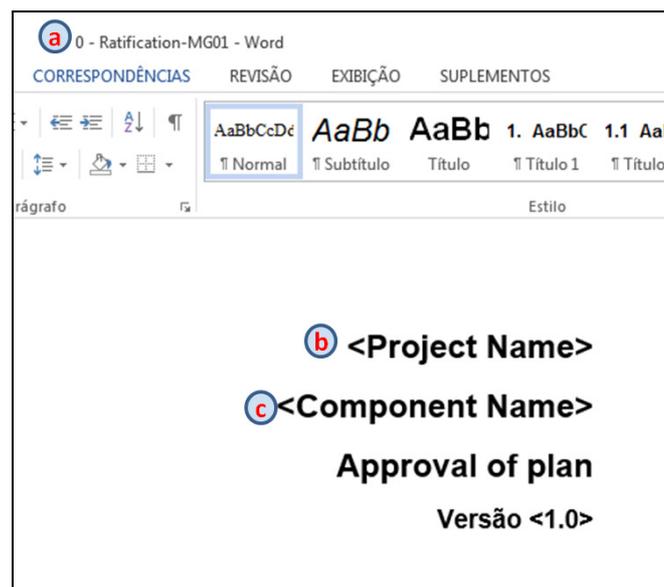


Figure 5.3 - Representation of the word document.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

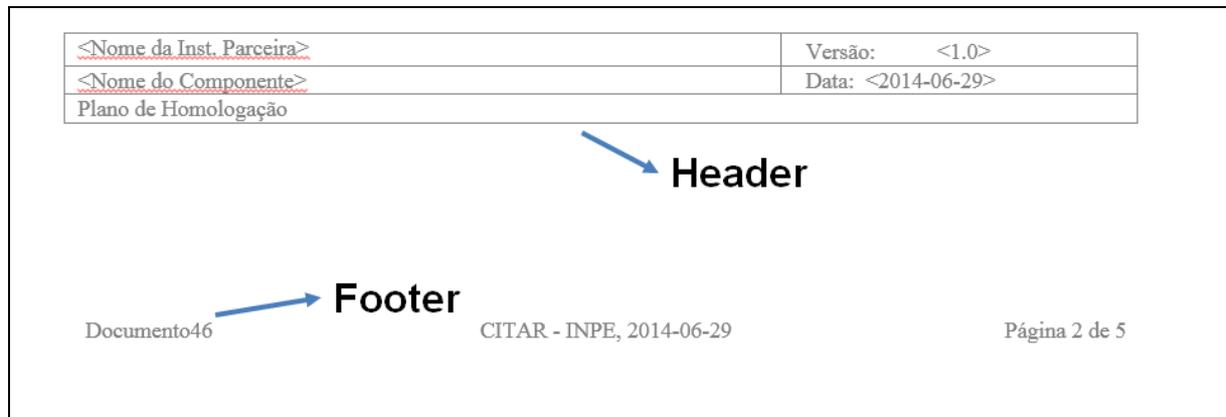


Figure 5.4 - header and footer template.

To load information from the document properties in the header and footer, it uses the same document body concept: open the header, select all and F9, repeat the same for the footer to update automatic footer fields.

5.3. Typing in the Document Interior.

The templates used in the process uses the same resource used by RUP templates and should be used as guidelines to follow and shown in Figure 5.5.

- a) All items of templates have guidance on how to be filled, these guidelines are described in italic blue.
- b) The content to be entered should be entered in automatic color standard document and
 - Position the course at the end of the guideline on the item to be described
 - Keystrokes enter where the cursor changes line.
 - Start typing already be with the source and standardized color.
 - To delete the blue line item completing orientation.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

3.1 **Components**

Component	Description	Responsible
<i>[Tell component]</i>	<i>[Write a brief description of the component]</i>	<i>[Tell the group, sector or organization responsible for developing the component]</i>

Note: You must include rows in the table above as there are components.

3.2 **Environment**
[Detail the environment where the product (or components) will be submitte]

I'm typing the desired contente |

3.3 **Development Environment**
[Detail the development environment as well as products and tools to be used]

3.4 **Use Environment**
[Identify and detail all the items that must be considered]

- *Tasks to be performed?*
- *Running time?*

Figure 5.5 - Typing Office in document.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

6. CMMI and Process

A good process does not have documentation to be filled documents generated in a good process are the result of good practices carried out in a natural way and credibility those who use it.

6.1. Level 2

A process in order to be classified in the maturity level 2, the organization's projects must be planned and executed in accordance with an existing policy, the status is easily identified; the people involved should be trained according to the identified needs to produce the expected outputs; involve all stakeholders; carry out controls and reviews by monitoring and validating. All these activities must be in accordance with the description of the process.

6.2. Service CMMI Practices

Indication of the coverage of CMMI practices in process.

6.2.1. Generic

The general practice assists in the effective implementation of the process, they must follow a framework of action in the context of business, where they are described and implemented. In general this practice guides for capturing the support of the great managers for their achievement.

In the process the generic practices are planned and made official as Table 6.1

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Table 6.1 - Generic practices.

Code	Description	Activity	Document	Phase
GP 2.2	Plan the Process	Planning	Development Plan Process Development External Documents	Engineering, Manages Requirements
GP 2.3	provide Resources	Planning	Development Plan	Engineering, Manages Requirements
GP2.4	assign Responsibilities	Planning	Development Plan	Engineering, Manages Requirements
GP2.5	training People	Planning	Development Plan External Documents	Engineering, Manages Requirements
GP2.6	Controlling Work Products	Project Control	Approval of Plan Development Plan	Common
GP2.7	Identify and Involve Relevant Stakeholders	Planning	Development Plan	Engineering, Manages Requirements
GP2.8	Monitor and Control the Process	Project Control	Development Plan	Common
GP2.9	Objectively Evaluate Adherence	Project Control	Plan Approval Development Plan	Common
GP2.10	Review Status with Higher Level Management	Project Control	Development Plan External Documents	Common

- **GP2.2** - The implementation of this generic practice is recorded in the "development plan" when the project planning follows the development process and must be justified following an institutional and / or product development plan that provides assurance that it is always performed.
- **GP2.3** - The implementation of this generic practice is recorded in the "development plan" in the format of staff specification plan and agreed and made possible by higher levels.
- **GP2.4** - The implementation of this generic practice is recorded in the "development plan" with the specification of the roles and responsibilities, approved and made possible by the higher levels.
- **GP2.5** - The implementation of this generic practice is recorded in the "development plan" with the acquisition planning and staff training following an institutional level of the organization and approved and made possible by the higher levels.
- **GP2.6** - The implementation of this generic practice is recorded in the "approval" document with the product acceptance planning and also in the configuration management plan document " according criteria of control and institutional approval and also the product development plan and approved and made possible by the higher levels.
- **GP2.7** - The implementation of this generic practice is recorded in the "development plan", with the definition of those responsible for the project, both the control and execution as the viability of the project, the latter with authority to make decisions. This activity must be approved and made possible by the higher levels of institutional form.
- **GP2.8** - The implementation of this generic practice is recorded in the "development plan", with the monitoring of the project and should be approved and made possible by the higher levels in accordance with institutional guidelines.
- **GP2.9** - The implementation of this generic practice is recorded in the "Approval of plan" and held in accordance with the development plan, review and approval of the artifacts

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

produced in each phase of the project should be approved and made possible by the higher levels of institutional form.

- **GP2.10** - The implementation of this generic practice is recorded in the "Approval of plan and development plan" at the end of each stage and held in accordance with the development plan must be approved and made possible by the higher levels of institutional form.

6.2.2. Specific Level 2

6.2.2.1. Manages requirements.

The activity requirement collection takes place in its respective phase. However, the alignment and control of activities are carried out in all stages. This activity is a structural basis for all other activities, as table 6.2.

Table 6.2 - Management of requirements.

Code	Description	Activity	Document	Phase
SP1.1	Understanding Requirements	Definition of Scope, Definition of Requirements Definition Component,	Product Specification Component Specification, Development Plan Documents Foreign	Engineering, Management of Requirements
SP1.2	Get commitment to the requirements	Planning	Development Plan	Engineering
SP1.3	Manage changes in requirements	Control Configuration, Control Set Items	Contoles Configuration	Common
SP1.4	Maintain Bidirectional Traceability of Requirements	Configuration control	Contoles configuration (matrix of traceability)	Common, Manages Requirements
SP1.5	Ensure alignment between Work Products, Design Plans and Requirements	Project Control	Approval of plan	Common

- **SP1.1** - The implementation of this specific practice occurs with the understanding of the component requirements and the positioning component requirements in the final product and should be reviewed and validated with those in charge according to plan.
- **SP1.2** - The implementation of this specific practice occurs with identifying the changing needs of the requirements and analysis of their internal and external consequences to the component, and identification of external changes of the occurring requirements and their impact on the component.
- **SP1.3** - The implementation of this specific practice occurs with the strict control of the changing requirements that happen, regardless of stimulus to this change, according to the development plan and configuration control.
- **SP1.4** - The implementation of this specific practice occurs with the mapping requirements and control and its consequences to the last level, where from a point of requirement their peers can be reached, as well as test plans, configuration, documentation etc. The control level should follow what had been set in the development plan.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP1.5** - The implementation of this specific practice occurs with the implementation of follow-up and approval of the artifacts produced in each stage of the process, according to the development plan.

6.2.2.2. Measurement and analysis

The activities of measurement and analysis process area start in the engineering phase and are repeated in all stages by the common activities. The distributions of coverage of their practices occur as provided in Table 6.3.

Table 6.3 - Measurement and analysis.

Code	Description	Activity	Document	Phase
SP1.1	Establish Measurement Objectives	Planning Metrics	Development plan	Engineering, Common
SP1.2	specify measures	Planning Metrics	Development plan	Engineering, Common
SP1.3	Specify Collection Procedures and Data Storage	Planning, Planning Metrics	Development plan	Engineering, Common
SP1.4	Specify Analysis Procedure	Planning, Planning Metrics	Development plan	Engineering, Common
SP2.1	Get Resulting Data Measurement	Planning, Planning Metrics	Development plan	Engineering, Common
SP2.2	Analyze Resulting Data Measurement	Project Control	Development plan	common
SP2.3	Store Data and Results	Project Control	Development plan	common
SP2.4	Report results	Cycle Assessment	Approval Document development plan	Cycle Assessment

- **SP1.1** - The implementation of this specific practice occurs with the implementation of activities Identification of the need for measurement of the items needed for the project and planning.
- **SP1.2** - The implementation of this specific practice occurs with the realization of the defining activities of the measures to be undertaken for the project according to the need.
- **SP1.3** - The implementation of this specific practice occurs with the realization of the defining activities of the measures to be held and specifying the storage media ways.
- **SP1.4** - The implementation of this specific practice occurs with the analysis of the definition of the measured items and communication to higher levels and stakeholders, according to the development plan.
- **SP2.1** - The implementation of this specific practice occurs with the definition of the storage structure and data analysis according to the development plan.
- **SP2.2** - The implementation of this specific practice occurs with the analysis of the stored data in accordance with the development plan and approved with the upper levels.
- **SP2.3** - The implementation of this specific practice occurs with the documentation of test results in accordance with the development plan.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP2.4** - The implementation of this specific practice occurs with the approval of the results of the measured and analyzed data in accordance with the development plan at the end of each phase and the end of the project.

6.2.2.3. Monitoring and project control

The activities of the process area, monitoring and project control, begin in the engineering phase as presented in table 6.4, and extend to ordinary activities.

Table 6.4 - Monitoring and project control.

Code	Description	Activity	Document	Phase
SP1.1	Monitor Project Planning Parameters	Project Control	Development Plan	Engineering
SP1.2	monitoring Commitments	Project Control	Development Plan	Engineering
SP1.3	Monitor Project Risks	Project Control	Development Plan	Engineering
SP1.4	Monitor Data Management	Project Control	Development Plan	Engineering
SP1.5	Monitor Stakeholder Involvement.	Project Control	Development Plan	Engineering
SP1.6	Driving Progress Reviews	Project Control	Development Plan Approval Plan	Engineering
SP1.7	Conduct reviews of Marco	Project Control	Development Plan Approval Plan	Engineering
SP2.1	Analyze Issues	Project Control	Development Plan Approval Plan	Engineering
SP2.2	Implement Corrective Actions	Project Control	Development Plan Approval Plan	Engineering
SP2.3	Manage Corrective Actions	Project Control	Development Plan Approval Plan	Engineering

- **SP1.1** - The implementation of this specific practice occurs with to the study and documentation of the parameters controlled by the project and its results in accordance to the development plan.
- **SP1.2** - The implementation of this specific practice occurs with the approval of the monitoring and the monitoring of measurements taken during the course of the project at each stage, according to the development plan.
- **SP1.3** - The implementation of this specific practice occurs with the monitoring of identified risks, documented in the project plan and reviewed at each phase of the project in accordance to the development plan.
- **SP1.4** - The implementation of this specific practice occurs with the completion of the approval taken at each phase of the project in accordance to the development plan.
- **SP1.5** - The implementation of this specific practice occurs with the completion of the approval activity in each phase of the project in accordance to the development plan.
- **SP1.6** - The implementation of this specific practice occurs with the completion of the approval activity in each phase of the project in accordance with the development plan and should also be carried out whenever necessary.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP1.7** - The implementation of this specific practice occurs with the completion of the approval activity in each phase of the project in accordance to the development plan.
- **SP2.1** - The implementation of this specific practice occurs with the completion of the approval activity in each phase of the project in accordance with the development plan and should also be carried out whenever necessary.
- **SP2.2** - The implementation of this specific practice occurs with the completion of the approval activity in each phase of the project in accordance with the development plan and should also be performed whenever necessary based on the identified issues.
- **SP2.3** - The implementation of this specific practice occurs with the realization of identified activities to address critical issues and should be included in the development plan and perform specific monitoring.

6.2.2.4. Management in accordance with suppliers

The activities of this process area have to direct their practices to all types of suppliers involved in the process, which can be suppliers of component, software or tools for development. In short supplier of all acquisitions that will take place during the project according to table 6.5.

Table 6.5 - Management in accordance with suppliers.

Code	Description	Activity	Document	Phase
SP1.1	Determine Acquisition Type	Planning, System Planning, System Design	Development Plan, Product Specification	Engineering, Management of Requirements
SP1.2	Select Suppliers	Planning	Development Plan, Product Specification	Common, Manages Requirements
SP1.3	Establish contracts with Suppliers	Planning	Development Plan, Product Specification	Common, Manages Requirements
SP2.1	Run Agreement with Supplier	Planning	Development Plan, Product Specification	Common, Manages Requirements
SP2.2	Accept Product Purchased	Project Control	Development Plan, Specification Product Approval Plan, Test Project	Common
SP2.3	Ensure Transition Products	Controle Design	Development Plan	Common

- **SP1.1** - The implementation of this specific practice occurs with the identification of components that can be purchased on the market or reused in other projects, and should be included in the development plan. Documentation must be attached to the documentation produced by the project and enter the normal according to the plan development.
- **SP1.2** - The implementation of this specific practice occurs with the identification of suppliers of components to be purchased on the market or reused in other projects, and should follow the selection criteria of supplier defined in the development plan.
- **SP1.3** - The implementation of this specific practice occurs with the formalization of the supply through the contract and included in the project documentation.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP2.1** - The implementation of this specific practice occurs with the beginning and the monitoring of the activities developed by the vendor and evaluated in the approval of activities at the end of each stage and whenever necessary.
- **SP2.2** - The implementation of this specific practice occurs with the registration and monitoring of the activities developed by the vendor and evaluated in the approval of activities at the end of each stage and whenever necessary.
- **SP2.3** - The implementation of this specific practice occurs with the monitoring of products developed by the vendor and evaluated in the approval of activities at the end of each stage and whenever necessary.

6.2.2.5. Project planning

The activities of the Project Planning process area start in the engineering phase and extend throughout the life cycle. They should be carried out very properly, because they are the basis for the realization of all other activities. The proposed process covers the practices of this process area as table 6.6.

Table 6.6 - Project planning.

Code	Description	Activity	Document	Phase
SP1.1	Estimate the Project Scope	Definition of Scope,	Especificaçãol Product, Component Specification, Development Plan	Engineering
SP1.2	Establish Estimates of Work Product and Task Attributes	Requirements specification	Especificaçãol Product, Component Specification, Development Plan	Manages Requirements
SP1.3	Define Project Life Cycle Stages	Planning	Development Plan	Engineering
SP1.4	Effort and Cost Estimates	Planning, Project Control	Development Plan	Engineering, Common
SP2.1	Establish the Budget and Schedule	Planning, Project Control	Development Plan	Engineering, Common
SP2.2	Identifying Project Risks	Planning, Project Control	Development Plan	Engineering, Common
SP2.3	Data Management Plan	Planning, Project Control	Development Plan	Engineering, Common
SP2.4	Plan Project Resources	Planning, Project Control	Development Plan	Engineering, Common
SP2.5	Plan Skills and Knowledge Required	Planning, Project Control	Development Plan	Engineering, Common
SP2.6	Plan Stakeholder Involvement	Planning, Project Control	Development Plan	Engineering, Common
SP2.7	Establish the Project Plan	Planning, Project Control	Development Plan	Engineering, Common
SP3.1	Review Plans that Affect the Project	Planning, Project Control	Development Plan Approval Plan	Engineering, Common
SP3.2	Reconciling Work Load and Resources	Planning, Project Control	Development Plan	Engineering, Common
SP3.3	Get Commitment to the Plan	Planning, Project Control	Development Plan	Engineering, Common

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP1.1** - The implementation of this specific practice occurs with the identification and definition of the component structure to be developed for its estimate
- **SP1.2** - The implementation of this specific practice occurs with the identification of the component attributes to be developed that will be considered for the project estimate.
- **SP1.3** - The implementation of this specific practice occurs with the organization of the phases and activities to be undertaken during the project.
- **SP1.4** - The implementation of this specific practice occurs with the implementation of project planning activities identifying the cost estimate and time, taking into account the estimates given by the product.
- **SP2.1** - The implementation of this specific practice occurs with the implementation of project planning activities, according to the development plan.
- **SP2.2** - The implementation of this specific practice occurs with the implementation of project planning, assessing the project's risk according to the development plan. This activity is performed focusing on product risks.
- **SP2.3** - The implementation of this specific practice occurs with the implementation of project planning activities by identifying the data to be measured and monitored throughout the project in accordance with the development plan. These data are identified according to the criticality of the component and meet the needs of the product and process.
- **SP2.4** - The implementation of this specific practice occurs with the implementation of identified project planning activities and allocate the necessary human resources for their development as well as the specification and provision of the necessary environment for the development of the project.
- **SP2.5** - The implementation of this specific practice occurs with the identification of technical and admiring features for project execution.
- **SP2.6** - The implementation of this specific practice occurs with the implementation of project planning activities with the allocation of human resources for carrying out activities.
- **SP2.7** - The implementation of this specific practice occurs with the implementation of project planning activities indicating the estimated time for completion of each activity.
- **SP3.1** - The implementation of this specific practice occurs with the monitoring of the project plan and approved at the end of each phase.
- **SP3.2** - The implementation of this specific practice occurs with the implementation of monitoring activities and project redesign to meet any demand or planning not accomplished.

6.2.2.6. Configuration management

The activities of the Configuration Management process area have their defined scope in the engineering phase; at this stage all the settings and all the changes that happen in all activities of all phases are controlled and the controls are not limited to the requirements and versions of

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

programs or components, but rather encompass all the elements that are part of the development environment and project implementation. Distributions of its activities are shown in Table 6.7.

Table 6.7 - Configuration management.

Code	Description	Activity	Document	Phase
SP1.1	Identify Configuration Items	Set Control Items	Configuration Controls	Manages Requirements
SP1.2	Establish a Configuration Management System	Planning	Development plan Configuration Controls Approval Document	Engineering
SP1.3	Create or Release Baselines	Configuration Control	Contoroles Configuration	Common
SP2.1	Track Change Requests	Configuration Control, Project Control	Configuration Contoles, Development Plan	Common
SP2.2	Control Settings Items	Configuration Control	Contoroles Configuration	Common
SP3.1	Establish Configuration Management Records	Configuration Control	Contoroles Configuration	Common
SP3.2	Perform Configuration Audits	Project Control	Approval of plan	Common

- **SP1.1** - The implementation of this specific practice occurs with the identification and planning of the items that will be monitored during the project in accordance with the development plan.
- **SP1.2** - The implementation of this specific practice occurs with the establishment of the change and configuration control mechanism.
- **SP1.3** - The implementation of this specific practice occurs with the record of the changes in accordance with the development plan.
- **SP2.1** - The implementation of this specific practice occurs with the analysis of the request for change and the record of the impact on the component and the development plan.
- **SP2.2** - The implementation of this specific practice occurs with the record of changes and the control of the used configuration.
- **SP3.1** - The implementation of this specific practice occurs with the record of changes, reporting and the used control of the configuration.
- **SP3.2** - The implementation of this specific practice occurs with the maintenance of records and evaluation of configuration data.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

6.2.2.7. Quality assurance

In the proposed process, the activities of the Quality Assurance process area have the responsibility to consolidate the implementation and the results of all activities of the process and the quality of the built product. Its activities are presented in Table 6.8.

Table 6.8 - Process and product quality assurance.

Code	Description	Activity	Document	Phase
SP1.1	Objectively Evaluate Processes	Cycle Assessment	Project Approval Document	Cycle Assessment
SP1.2	Objectively Evaluate Work Products	SW Test , HW Test	Development Plan, Test Plan, Project Approval Document	Verification and Validation
SP2.1	Report and Problem of Default	Closing Cycle Assessment	Project Approval Document	Cycle Assessment
SP2.2	Establish Records	Cycle Assessment	Approval Document	Cycle Assessment

- **SP1.1** - The implementation of this specific practice occurs with the completion of the evaluation activities of project progress against the description of the process and the project plan, as well the assessment of the artifacts produced in relation to the product specification.
- **SP1.2** - The implementation of this specific practice occurs with the performance of activities of the artifacts produced in relation to the product specification.
- **SP2.1** - The implementation of this specific practice occurs with the completion of the approval activities in each phase of the project, recording and communicating noncompliance.
- **SP2.2** - The implementation of this specific practice occurs with the implementation of project planning activities with the record of the evaluations.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

6.2.3. Specific level 3

6.2.3.1. Requirements development

Validation in the required scenarios, and its activities are presented in Table 6.9.

Table 6.9 - Requirements development.

Code	Description	Activity	Document	Phase
SP1.1	Lifting Needs + B3: B10	Definition of Scope, Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP1.2	Transforming the needs of those involved in customer requirements.	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.1	Establish Product Requirements and Product Component	Project Control	Approval of plan	Engineering, Management of Requirements
SP2.2	Allocate Product Component Requirements	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.3	Identify Interface Requirements	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.1	Establish Operational Concepts and Scenarios	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.2	Establish a Definition of Required Functionality and Quality Attributes.	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.3	Analyze Requirements	Definition of Requirements Definition Component, Project Control	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.4	Analyze Requirements In order to Balancing	Definition of Requirements Definition Component , Project Control	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.5	validate Requirements	Project Control	Approval of plan	Common

- **SP1.1** - The implementation of this specific practice occurs with the Survey of preliminary and detailed data of the product / component to be developed in phases of engineering and requirements management respectively.
- **SP1.2** - The implementation of this specific practice occurs by detailing user requirements in general specification of the component.
- **SP2.1** - The implementation of this specific practice occurs as a result of the project monitoring and approval, when it is evaluated the adherence of components to the product throughout the process.
- **SP2.2** - The implementation of this specific practice occurs with the distribution activity of functionality and component modeling.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP2.3** - The implementation of this specific practice occurs with the distribution activity of functionality and component modeling.
- **SP3.1** - The implementation of this specific practice occurs with the identification activity and construction of the operating environment and component development.
- **SP3.2** - The implementation of this specific practice occurs with the identification, definition and specification of the special component requirements.
- **SP3.3** - The implementation of this specific practice occurs with the approval of the activity and verification requirements, according to the plan, throughout the project.
- **SP3.4** - The implementation of this specific practice occurs with identification, definition and specification of the special requirements and component constraints.
- **SP3.5** - The implementation of this specific practice occurs with the component approval activities in each phase of the project according to the planning.

6.2.3.2. Technical solution

The meeting of this process area is done with the study and proposal of solutions for the development or acquisition of the component and its activities are presented in Table 6.10.

Table 6.10 - Technical solution.

Code	Description	Activity	Document	Phase
SP1.1	Develop workarounds and Selection Criteria	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP1.2	Select Product Component Solutions	Definition of Requirements Definition Component,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.1	Develop the Design Product or Product Components	Definition of components,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.2	Establish Technical Data Package	Contolar Configuration	Configuration	Common
SP2.3	Design Interfaces Using Criteria	Definition of components,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.4	Analyze Alternatives: Develop, Buy or Reuse	Definition of components,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.1	Implement Design	Definition of components,	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP3.2	Develop Supporting Documentation to product	NO		

- **SP1.1** - The implementation of this specific practice occurs with the study and proposal of solutions according to the criteria as defined in the development plan.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP1.2** - The implementation of this specific practice occurs with the study and proposal of solutions according to the criteria as defined in the development plan.
- **SP2.1** - The implementation of this specific practice occurs with the realization of the component modeling activity.
- **SP2.2** - The implementation of this specific practice occurs with the setting, control and storage of the configuration of all the artifacts produced during the project.
- **SP2.3** - The implementation of this specific practice occurs with the realization of the component modeling activity.
- **SP2.4** - The implementation of this specific practice is given to occurs with the study and proposal of solutions according to the criteria as defined in the development plan.
- **SP3.1** - The implementation of this specific practice occurs with the component implementation.
- **SP3.2** - The implementation of this specific practice cannot be considered as accomplished it once directs for the preparation of the documentation for the end user, understanding that the end user is the user of the product, and this process focuses on the development component, it is considered to have failed. However, according to the component development view, all documents produced by the project could be considered as documentation for the end user, since the component user is the product or system.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

6.2.3.3. Product integration

This process area is granted with the completion of the integration activities of the proposed process and realized in the integration phase and its activities are presented in Table 6.11.

Table 6.11 - Product integration.

Code	Description	Activity	Document	Phase
SP1.1	Establish an Integration Strategy	Component Project Planning	Product Specification , Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP1.2	Establish Product Integration Environment	Configuration Control	Product Specification , Component Specification, Development Plan External Documents	Management of Requirements Common
SP1.3	Establish Procedures and Criteria for Product Integration	Planning	Product Specification , Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.1	Review Interfaces descriptions to ensure completeness	Project Control	Approval of plan	common
SP2.2	Manage Interfaces	Configuration Control	Component Specification, Development Plan External Documents Configuration	Management of Requirements Common
SP3.1	Confirm that the product components are Ready to be Integrated	Project Control	Approval of plan	Common
SP3.2	Assemble Product Components	Integration	Product Specification, Component Specification, Development Plan External Documents	Implementation and Integration
SP3.3	Evaluate Product Components Mounted	Project Control	Approval of plan	Common
SP3.4	Packaging and Delivering Product or Product Component	Closing	Project	Cycle evaluation

- **SP1.1** - The implementation of this specific practice occurs with the component modeling and planning.
- **SP1.2** - The implementation of this specific practice occurs with the control configuration according to the schedule.
- **SP1.3** - The implementation of this specific practice occurs with the component modeling and planning.
- **SP2.1** - A implementation of this specific practice occurs with the realization of approval activities in accordance with the planning.
- **SP2.2** - A implementation of this specific practice occurs with implementation of configuration control activities in accordance with the planning.
- **SP3.1** - A implementation of this specific practice occurs with the implementation of project control activities and approval in accordance with the planning.
- **SP3.2** - A implementation of this specific practice occurs with the realization of product integration activity.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP3.3** - The implementation of this specific practice is with the realization of project control activity, according to the planning.
- **SP3.4** - The implementation of this specific practice occurs by performing the closing activity in the evaluation phase of the cycle, according to the planning.

6.2.3.4. Verification

Verification activities are planned during the project planning and put in practice during validation phase and its activities are presented in Table 6.12.

Table 6.12 - Verification.

Code	Description	Activity	Document	Phase
SP1.1	Select Work Products for Verification	Planning, Test SW, Test HW	Product Specification, Component Specification, Development Plan, SW, Test Project	Engineering Verification and Validation
SP1.2	Establish Verification Environment	Planning Control Configuration	Component Specification, Development Plan External Documents Configuration	Engineering Common
SP1.3	Establish procedures and verification criteria	Planning	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.1	Prepare for Peer Review	NO		
SP2.2	Driving Peer Review	NO		
SP2.3	Analyze Data Peer Review	NO		
SP3.1	perform Verification	Test SW, Test HW	Product Specification, Component Specification, Development Plan, SW, Test Project	Verification and Validation
SP3.2	Analyze Scan Results	Test SW, Test HW Closing	Product Specification, Component Specification, Development Plan , SW, Test Project	Verification and Validation Cycle evaluation

- **SP1.1** - The implementation of this specific practice starts in the implementation of planning activity and remains in the Software and hardware testing activities.
- **SP1.2** - The implementation of this specific practice occurs with carrying out the activity definition and preparation of the development environment.
- **SP1.3** - The implementation of this specific practice occurs with realization of the planning activity.
- **SP2.1** - This particular practice is considered not since this process can also be used for individual development, however this practice can be planned for in the project planning activity.
- **SP2.2** - This particular practice is considered not met since this process can also be used for individual development, however this practice can be planned in the project planning activity.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP2.3** - This particular practice is considered not met since this process can also be used for individual development, however this practice can be planned in the project planning activity.
- **SP3.1** - The implementation of this practice occurs with the activities of software and hardware testing according to the schedule and the test plan.
- **SP3.2** - The implementation of this practice occurs with the approval of activities and the completion of the evaluation phase of the cycle according to the planning.

6.2.3.5. Validation

The activities of verification and validation phases serve the process area throughout the project following the planning and approval cycle assessment. Its activities are presented in table 6.13.

Table 6.13 - Validation.

Code	Description	Activity	Document	Phase
SP1.1	Select Products for Validation	Test SW, Test HW	Product Specification, Component Specification, Development Plan, SW, Test Project	Verification and Validation
SP1.2	Establish Validation Environment	Configuration Control	Component Specification, Development Plan External Documents	Common
SP1.3	Establish procedures and Validation Criteria	Planning	Product Specification, Component Specification, Development Plan, External Documents	Engineering, Management of Requirements
SP2.1	perform Validation	Test SW, Test HW	Product Specification, Component Specification, Development Plan, SW, Test Project	Verification and Validation
SP2.2	Analyze Validation Results	Test SW, Test HW	Product Specification, Component Specification, Development Plan, SW, Test Project	Verification and Validation

- **SP1.1** - The implementation of this specific practice occurs with realization of the planning activity.
- **SP1.2** - The implementation of this specific practice occurs with carrying out the activity definition and preparation of the development environment.
- **SP1.3** - The implementation of this specific practice occurs with realization of the planning activity.
- **SP2.1** - The implementation of this practice occurs with the activities of software and hardware testing according to the schedule and the test plan.
- **SP2.2** - The implementation of this practice occurs with the approval of activities and the completion of the evaluation phase of the cycle, according to the planning.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

6.2.3.6. Focus on the organization process

This process area is not directly served by the proposed process, because it is made of organizational form activities. The activities of this process area are presented in Table 6.14.

Table 6.14 - Focus Organization Process.

Code	Description	Activity	Document	Phase
SP1.1	Establish Organizational Process Needs	NO		
SP1.2	Assess the Organization's Processes	NO		
SP1.3	Identify improvements to the Organization's Processes	NO		
SP2.1	Establish Process Action Plans	NO		
SP2.2	Implement Process Action Plans	NO		
SP3.1	Deploy Organizational Process Assets	NO		
SP3.2	Deploy Processes standard	NO		
SP3.3	Monitor the Implementation	NO		
SP3.4	Embed Experiences in Organizational Process Assets	NO		

The realization process of the Focus Area Organization Process practices, are not considered to be met, since this process area is related to the implementation of standard processes in the organization. The method in this paper and those practices happen to meet the needs of the individual process, therefore enabling flexible adaptation according to the need of the component to be produced, thus making this process area unfeasible to be serviced.

6.2.3.7. Definition of organization process.

This process area is serviced by defined the proposed project, and follow guidelines belonging to the project in an individual way. and its activities are presented in Table 6.15.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Table 6.15 - Definition of organization process.

Code	Description	Activity	Document	Phase
SP1.1	Establish Processes standard	NO		
SP1.2	Establish Lifecycle Model Descriptions	Planning	Component Specification, Development Plan External Documents	Engineering,
SP1.3	Establish Criteria and Guidelines for Adaptation	Planning	Component Specification, Development Plan External Documents	Engineering,
SP1.4	Establish the Organization's Measurement Repository	Project Control	Development plan	Common
SP1.5	Establish the Organization's Process Asset Library	Project Control	Development plan	Common
SP1.6	Establish Work Environment Standards	Control Configuration Planning	Contoles Configuration Development Plan	Common, Manages Requirements
SP1.7	Establish rules and guidelines for teams.	Planning	Development Plan	Manages Requirements Engineering

- **SP1.1** - This practice is not considered met because the process provides flexibility for adaptation according to the need component of the to be produced, thus not permitting the coverage of this practice.
- **SP1.2** - This practice is fulfilled by in the realization of project planning activity, with the adaptation of the project according to the component to be produced.
- **SP1.3** - Meeting this practice occurs by carrying out the project planning activity, with the adaptation of the project according to the component to be produced.
- **SP1.4** - Meeting this practice occurs by carrying out the design control activity throughout the life cycle according to plan.
- **SP1.5** - Meeting this practice occurs with the realization of the definition of items to be controlled by the activity of control setting.
- **SP1.6** - Meeting this practice occurs with the definition of the configuration environment for the activity of control setting and planning.
- **SP1.7** - Meeting this practice occurs with the implementation of planning activities.

6.2.3.8. Organizational training

This process area is partially met since the training strategies are defined in the proposed process and the training of those involved get on as necessary to meet the project, but does not define organizational guidelines. Its activities are presented in Table 6.16.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Table 6.16 - Organizational training.

Code	Description	Activity	Document	Phase
SP1.1	Establish Strategic Needs Training	Planning	Development Plan	Manages Requirements Engineering
SP1.2	Identify Training Needs in Organization Responsibility	Planning	Development Plan	Manages Requirements Engineering
SP1.3	Establish a Tactical Training Plan in the Organization	NO		
SP1.4	Establish Training Capability	NO		
SP2.1	provide Training	NO		
SP2.2	Establish Training Records	NO		
SP2.3	Evaluate the Effectiveness of Training	NO		

- **SP1.1 and SP1.2** - These practices are accomplished by realizing the project's planning activity, with the team's allocation and preparation.
- **SP1.3 to SP2.3** - These practices are not considered to be met as the practices to be realized in the Organizational Training process have to be held according to organizational guidelines, and the suitable training plan in this process occurs from the project needs, according to the component to produced.

6.2.3.9. Project integrated management

This area is partially answered by the process as it performs integration activities and individual component project control, as well as the entire system along the project development. Therefore it can not meet it as a whole, as it is not combined with the processes the organization. Its activities are presented in Table 6.17.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Table 6.17 - Integrated project management.

Code	Description	Activity	Document	Phase
SP1.1	Establish the Defined Process for the Project	Planning	Development Plan	Manages Requirements Engineering
SP1.2	Using the Organizational Process Assets for Planning the Project Activities	NO		
SP1.3	Establish the Project Desktop	NO		
SP1.4	integrate Plans	Planning	Development Plan	Manages Requirements Engineering
SP1.5	Manage the Project Using Integrated Plans	Planning	Development Plan	Manages Requirements Engineering
SP1.6	establish Teams	Planning	Development Plan	Manages Requirements Engineering
SP1.7	Contribute to the Organizational Process Assets.	Configuration control	Contoles Configuration Development Plan	Common,
SP2.1	Manage Stakeholder Involvement	Planning Control Project Approval	Development Plan	Common Engineering
SP2.2	Manage Dependencies	Definition of Requirements Definition Component, Project Control	Product Specification, Component Specification, Development Plan External Documents	Engineering, Management of Requirements
SP2.3	Resolving Issues Coordinating Reviews	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common

- **SP1.1** - Meeting this practice occurs with the implementation of project planning activities from the life cycle defined for the project.
- **SP1.2** - This practice is not interpreted as not fulfilled, as this activity in the suggested process is related to the component of the project and not following organizational guidelines.
- **SP1.3** - This practice is not interpreted as not fulfilled, as this activity in the suggested process is related to the component of the project and does not follow organizational guidelines.
- **SP1.4 to 1.6** - These practice is carried out in the project planning, with project planning activity.
- **SP1.7** - This practice is used in the control configuration, according to plan.
- **SP2.1 to SP2.3** - These practices are based on the approval activity in project control.

6.2.3.10. Risk management

This process area is carried out by the proposed process starting with the planning and throughout the completion of the project. Its activities are presented in Table 6.18.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Table 6.18 - Risk management.

Code	Description	Activity	Document	Phase
SP1.1	Determine Sources and Risk Categories	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP1.2	Set Parameters for Risk	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP1.3	Establish a Strategy for Risk Management	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP2.1	Identify Risk	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP2.2	Evaluate, Categorize and prioritize risks	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP3.1	Develop Risk Mitigation Plans	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP3.2	Implement Risk Mitigation Plans	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common

- **SP1.1 to SP3.2** - These specific practices are carried out in project planning and are monitored throughout the project's progress in project control.

6.2.3.11. Decision analysis and resolution

The proposed process doesn't offer specific activities for the decision making. However, in an indirect way the process helps in this activity. The activities of this process area are shown in Table 6.19.

Table 6.19 - Decision analysis and resolution.

Code	Description	Activity	Document	Phase
SP1.1	Establish Guidelines for Analysis and Decision	Planning, Approval Cycle evaluation	Development Plan Approval Document	Engineering, Common Cycle Assessment
SP1.2	Establish Evaluation Criteria	Planning, Approval Cycle evaluation	Development Plan Approval Document	Engineering, Common Cycle Assessment
SP1.3	Identify workarounds	Planning, Approval Cycle evaluation	Development Plan Approval Document	Engineering, Common Cycle Assessment
SP1.4	Select Assessment Methods	Planning, Approval Cycle evaluation	Development Plan Approval Document	Engineering, Common Cycle Assessment
SP1.5	Evaluate Workarounds	Planning, Approval Cycle evaluation	Development Plan Approval Document	Engineering, Common Cycle Assessment
SP1.6	select Solutions	Planning, Approval Cycle evaluation	Development Plan Approval Document	Engineering, Common Cycle Assessment

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP1.1, SP1.2, SP1.3, SP1.4, SP1.5 and SP1.6** - The practices in this process area are described in the project planning and held throughout the course of the project in the design control activity and finalized during the closing activity.

6.2.4. Specific level 4

6.2.4.1. Organizational process performance

The process proposed in its planning and the nature of the conduction of activities, partially meets this process area, for good practices result in better process performance. However, their focus is product performance, and its activities are presented in Table 6.20.

Table 6.20 - Organizational process performance.

Code	Description	Activity	Document	Phase
SP1.1	Establish Quality Objectives and Process Performance	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP1.2	select Process	NO		
SP1.3	Establish Process Performance Measures	NO		
SP1.4	Analyze the Process Performance Baselines and Establish Process Performance	NO		
SP1.5	Setting Process Performance Models	NO		

- **SP1.1** - This particular practice is carried out in the project planning and monitored throughout the project's progress in the design of control.
- **SP1.2 to SP1.5** - These specific practices are not considered met, as they refer to organizational activities the performance evaluation activities in this process relate in part to the performance of the process and deeper into the performance of its component products.

6.2.4.2. Quantitative project management

The proposed process partially serves the area of Project Quantitative Management, establishing and defining management activities, and their activities are presented in Table 6.21.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

Table 6.21 - Project quantitative management.

Code	Description	Activity	Document	Phase
SP1.1	Establish the Project Objectives	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP1.2	Compose the Defined Process	Project Control Planning	Product Specification, Component Specification, Development Plan	Engineering, Common
SP1.3	Select Subprocesses and Attributes	NO		
SP1.4	Select Measurements and Analytical Techniques	NO		
SP2.1	Monitor the Performance of Selected Subprocesses	NO		
SP2.2	Manage Project Performance	NO		
SP2.3	Perform root cause analysis.	NO		

- **SP1.1** - This particular practice is carried out in the project planning and monitored throughout the project's progress in the design of control.
- **SP1.2** - This particular practice is carried out in the project planning and monitored throughout the project's progress in the design of control.
- **SP1.3 to SP2.3** - These practices are not considered to be met by the process, since they are related to of process performance and that, the process must be rigorously defined and followed. This is not feasible for the flexibility of the process, according to the component to be produced.

6.2.5. Specific level 5

6.2.5.1. Causal analysis and resolution

The process meets this process area from project planning, indicating points of analysis and monitoring in all its stages in the finalization with the evaluation cycle, and its activities are presented in Table 6.22.

Table 6.22 - Causal analysis and resolution.

Code	Description	Activity	Document	Phase
SP1.1	Select results for Analysis	Project Control	Plan Approval Development plan	Common Engineering
SP1.2	analyze Causes	Project Control	Plan Approval Development plan	Common Engineering
SP2.1	Implement Action Proposals	Project Control	Plan Approval Development plan	Common Engineering
SP2.2	Evaluate Implemented Actions Effects	NO		
SP2.3	Register Data	Closing	Project	Cycle evaluation

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

- **SP1.1 to SP2.1** - These practices are realized during the control of the project, according to planning approval.
- **SP2.2** - This particular practice is considered to have failed, because it is not an explicit action of the process, but it may be included in the project planning to be carried out.
- **SP2.3** - This particular practice is carried out in the evaluation phase of the cycle with a closing activity.

6.2.5.2. Management of organizational process

In the evaluation phase of the cycle, especially in the process of closing activity, are carried out performance evaluation activities throughout the project life cycle, seeking to understand the results achieved and restructure new cycles from successes and failures suffered. Such activities are presented in Table 6.23.

Table 6.23 - Management of organizational process.

Code	Description	Activity	Document	Phase
SP1.1	Keep business objective.	NO		
SP1.2	Analyze process performance data	Closing	Project	Cycle evaluation
SP1.3	Identify potential areas for improvement	Closing	Project	Cycle evaluation
SP2.1	Identificar suggested improvements	Closing	Project	Cycle evaluation
SP2.2	Analyze suggestions for improvements	NO	Project	Cycle evaluation
SP2.3	validate improvements	Closing	Project	Cycle evaluation
SP2.4	Selecting and implementing improvements to	Closing	Project	Cycle evaluation
SP3.1	plan Deployment	Closing	Project	Cycle evaluation
SP3.2	Manage Deployment	Closing	Project	Cycle evaluation
SP3.3	To evaluate the effects Improvements	NO		

- **SP1.1, SP2.2 and SP3.3** - The process does not meet these specific practices, because they are not direct actions that are taken to meet their needs. However, in the process, it is possible to carry them out if included in planning.
- **SP1.2, SP1.3, SP2.1, SP2.3, SP2.4, SP3.1 and SP3.2.** - These specific practices are carried out in the evaluation cycle in the closing activity.

Process for the Development of Embedded Systems	Version:
National Institute for Space Research Systems Development Process Embedded	Date:
Laboratory	

7. References

- European Cooperation for Space Standardization (ECSS). ECSS Secretariat ESA-ESTEC ECSS-E-10 Part 1B, Space engineering – Requirements & Standards Division, Noordwijk, The Netherlands, 2004.
- European Cooperation for Space Standardization (ECSS). ECSS Secretaria - ESA-ESTEC ECSS-E-ST-40C, - Engenharia espacial – ECSS Secretaria - ESA-ESTEC - Requisitos e Divisão de Normas - Noordwijk, Holanda
- Miyashiro, M. A. S, Ferreira, M.G.V, Sannt’ana N. - Viability of Application of the process areas of CMMI-DEV processes Development of Critical Embedded Systems- Science and Information Conference - London (SAI2015) Approved for publication and presentation.
- SOFTWARE ENGINEERING INSTITUTE. - CMMI for Development, Version 1.3 CMMI DEV, V1.3 - CMMI Product Team – Improving processes for developing better products and services - Technical Report - Pittsburgh: Carnegie Mellon University, November 2010.
- Miyashiro, M. A. S., Identificação e melhoria do nível de maturidade de uma organização explorando técnicas de inteligência computacional. 2007. 100 f. Dissertação (Mestrado) - Instituto Nacional de Pesquisas Espaciais, São José Dos Campos, 2007.
- Miyashiro, M. A. S, Ferreira, M.G.V. One approach to the use of the practices of CMMI-DEV V1.3 Level 2 in a process of development of Embedded Systems - The Fifth International Conference on Information, Intelligence, Systems, and Applications – Greece (IISA 2014).
- Miyashiro, M. A. S, Ferreira, M.G.V. - Process for the development of Embedded System following the practices of CMMI Level 2 - Science and Information Conference - London (SAI2014). Nesta conferencia estão previstas reuniões para a formação de um grupo de estudo em engenharia de software para sistemas embarcados.
- Miyashiro, M. A. S, Ferreira, M.G.V. - The Practices of Level 2 (Managed) CMMI-DEV V1.3 in Development of Embedded Systems - International Conference on Electrical and Electronic Engineering – Hong Kong (EEE2014).
- Miyashiro, M. A. S, Ferreira, M.G.V. - Phase Cyclical Process Requirements for the Development Of Embedded Systems - "IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book):2013 III Brazilian Symposium on Computing Systems Engineering" – Brazil (SBESC2013).
- Miyashiro, M. A. S - An Approach to System Development Process Embedded that answers the Level 2 of CMMI-DEV Maturity "Doctoral Thesis. - National Institute for Space Research, São José Dos Campos, with defense date 10/02/2015.
- IBM Rational Unified Process (RUP) - <http://www-01.ibm.com/software/rational/rup> Accessed October 2012.

<Institution Name Partner>

<Name of Project>
<Name of Component>
Development Plan
Versão **<1.0>**

<Author Name>

<Institution Name Partner>	Versão: <1.0>
<Name of Component>	Criado em 25/06/2014
Development Plan	

Revision History

Date	Version	Description	Author
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Institution Name Partner>	Versão: <1.0>
<Name of Component>	Criado em 25/06/2014
Development Plan	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Project Overview	4
2.1	Purpose, Scope and Project Objectives	4
2.2	Assumptions and Constraints	4
2.3	Project Deliverables	4
3.	Project Organization	5
3.1	Organizational Structure	5
3.2	Roles and Responsibilities	5
4.	Development Environment	5
4.1	Tools	5
4.2	Environment	5
4.3	Infrastructure	5
4.4	Development Standard	5
5.	Process Management	6
5.1	Project Estimates	6
5.2	Project Plan (Hardware / Software)	6
5.2.1	Phase Plane	6
5.2.2	Objectives of the cycles	6
5.2.3	Releases	6
5.2.4	Project Schedule	6
5.2.5	Project Resources	6
5.3	Project Monitoring and Control	7
5.3.1	Requirements Management Plan	7
5.3.2	Schedule Control Plan	7
5.3.3	Quality Control Plan	7
5.3.4	Reporting Plan	7
5.3.5	Measurement Plan	7
5.3.6	Configuration Management Plan	7
5.3.7	Risk Management Plan	7
5.4	Product Acceptance Plan	7
5.5	Purchase Plan	7
5.5.1	Identification of Candidate Suppliers	7
5.5.2	Selection Criteria	7
5.5.3	Organization contractual with the supplier	7
5.5.4	Plan of supplier development Summary	7
6.	Attachments	7

<Institution Name Partner>	Versão: <1.0>
<Name of Component>	Criado em 25/06/2014
Development Plan	

<Development Plan>

1. Introduction

[The introduction of the Development Plan should provide an overview of the entire document. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and an overview of this Development Plan]

1.1 Purpose

[Specify the purpose of this Development Plan]

1.2 Scope

[A brief description of the scope of this Development Plan the projects to which it is associated with and anything else that is affected or influenced by this document]

1.3 Definitions, Acronyms and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Development Plan. This information can be provided by reference to the project Glossary]

1.4 References

[This subsection provides a complete list of all documents referenced in the Development Plan. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

In the case of the Development Plan, the referenced articles list includes:

- *Configuration*
- *Component Specification*
- *Product Specification*
- *Test Project*
- *Product Approval*
- *Development Process]*

1.5 Overview

[This subsection describes what the rest of the Development Plan and explain how the document is organized]

2. Project Overview

2.1 Purpose, Scope and Project Objectives

[A brief description of the purpose and objectives of this project and a brief description of the products is expected that the project release]

2.2 Assumptions and Constraints

[A list of the assumptions on which this plan is based and of any restrictions, for example, budget, staff, equipment, schedule, that apply to the proje]

2.3 Project Deliverables

[A table listing the artifacts to be created during the project, including target delivery dates (WBS)]

<Institution Name Partner>	Versão: <1.0>
<Name of Component>	Criado em 25/06/2014
Development Plan	

3. Project Organization

3.1 Organizational Structure

[Describe the organizational structure of the project team, including management authorities and other authorities review]

3.2 Roles and Responsibilities

[Identify the project organizational units that will be responsible for each subject, details of the workflow and support processes as well as those responsible for making decisions.]

4. Development Environment

- *[Describe the computing environment and the component tools to be used to carry out development activities that will be used for the project]*

- *predicted size of product data*

- *Product Team distribution*

- *physical location of servers and clients]*

4.1 Tools

[Describe the tools to be used in the computing environment]

4.2 Environment

[Describe the Environment Required]

4.3 Infrastructure

- *[Describe Infra Structure Required*

- *Tools*

- *Nomenclature*

- *Controls*

- *other]*

4.4 Development Standard

[Describe all items that should be standardized for all products and or components and to be observed in all development cycles]

4.5 Document Name Standard

[Describe the standard names that should be used for documents created during the project, the items that make up the name and the meaning of each item, depending on the model:

NP = Project number, sequence number that identifies the project

NCP = sequential number that identifies the component of the project to be developed in the cycle

PPP = Abbreviation of the Project, the project distinguishing sign in general Abbreviations project name

CD = document ID, ID number of the documents in the file

- *Development Plan Document - 02*

- *Product Specification Document - 03*

- *Component Specification Document - 04*

- *Test Project Document - 05*

<Institution Name Partner>	Versão: <1.0>
<Name of Component>	Criado em 25/06/2014
Development Plan	

- *Configuration document - 06*
- *Product Approval Document - 07*

NNN = sequential document number

VV = Document Version, identification of changes in the document

Document Name

NP-NCP- PPP-CD-NNN-VV]

5. Process Management

5.1 Project Estimates

[The planning component consider building Hardware and Software in distinct and shared activities, where view is clearly the needs, characteristics and actions relevant to each of the component (Hw and Sw). You can develop a project plan for the two parties or develop a project plan for each part]

5.2 Project Plan (Hardware / Software)

[Provide programming and the estimated cost of the project, as well as the basis of these estimates, and the points and circumstances in the project that will be made new estimates]

5.2.1 Phase Plane

[Include the following:

- *Work Breakdown Structure (WBS)*
- *A timeline or a Gantt chart showing the allotted time for the phases and cycles of the project*
- *Identification of major milestones with their success criteria*

Set all statements and important points of release

- *Identification of all validation milestones*

5.2.2 Objectives of the cycles

[List the objectives to be achieved for each of the cycles]

5.2.3 Releases

[A brief description of each component and release is a beta version, demonstration etc..]

5.2.4 Project Schedule

[Diagrams or tables showing target dates for completion of the cycles and phases of release points, demonstrations and other landmarks

- *General Schedule and detailed schedule for all phases of the project]*

5.2.5 Project Resources

5.2.5.1 Staffing Plan

[Identify here the numbers and types of necessary human resources, including all experience or special skills, programmed by phase or project cycle]

5.2.5.2 Plan Resource Acquisition

Describe how you want to find and acquire people to integrate the necessary staff to the project]

5.2.5.3 Training Plan

[List any special training required for the project team members, with target dates identifying when the training must be completed]

<Institution Name Partner>	Versão: <1.0>
<Name of Component>	Criado em 25/06/2014
Development Plan	

5.3 Project Monitoring and Control

[The items below indicate when the document used]

5.3.1 Requirements Management Plan

[Describe the approach adopted to monitor the requirements]

5.3.2 Schedule Control Plan

[Describe the approach taken to monitor progress in view of the planned schedule and how to take corrective action when necessary]

5.3.3 Quality Control Plan

[Describe the progress and the methods to be used to control the quality of the project deliverables and how to take corrective action when necessary]

5.3.4 Reporting Plan

[Describe internal and external reports to be generated, and the frequency and distribution of publication]

5.3.5 Measurement Plan

[Describe the approach taken to set and monitor the metrics]

5.3.6 Configuration Management Plan

[Describe the approach adopted to control the configuration]

5.3.7 Risk Management Plan

[Describe the approach taken to manage risks]

5.4 Product Acceptance Plan

[Describe the approach taken and indicate when the document used]

5.5 Purchase Plan

[Describe the approach taken and indicate when the document used]

5.5.1 Identification of Candidate Suppliers

[Describe the list of suppliers able to meet the requirements]

5.5.2 Selection Criteria

[Describe the approach taken to analyze and identify candidates suppliers]

5.5.3 Organization contractual with the supplier

[Enter the contract that will be used with the supplier (file name, storage form), responsible for the contract (contractor and hired) include a copy of the contract the project documentation]

5.5.4 Plan of supplier development Summary

[Indicate sufficiently the supplier of the work plan for the development of component]

6. Attachments

[Additional material of use to the Development Plan of the reader]

<Institution Name Partner>

<Name of the Project>

Product Specification

Name of the Component >

Version <1.0>

<Author's name>

<Institution Name Partner>	Version: <1.0>
Name of the Component >	Date: 6/25/2014
Product Specification	

Revision History

Date	Version	Description	Author
<dd/mm/aa>	<x.x>	<detalhes>	<nome>

<Institution Name Partner>	Version: <1.0>
Name of the Component >	Date: 6/25/2014
Product Specification	

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 General scope	4
1.3 References	4
1.4 Overview	4
2. Positioning.	4
2.1 Project Overview	4
2.2 Representation Product Graphic	4
3. Identification of Components Involved and Responsible	4
3.1 Components	5
3.2 Environment	5
3.3 Development Environment	5
3.4 Use Environment	5
3.5 Component description to be produced.	5
3.5.1 < Component Name >	5
4. Product Features	5
5. Restrictions	5
6. Quality Ranges	5
7. Precedence and Priority	6
8. Other Product Requirements	6
8.1 Applicable Standards	6
8.2 System Requirements	6
8.3 Performance Requirements	6
8.4 Environmental Requirements	6
9. File Requirements	6
9.1 Legal documents	6
9.2 Other documents	6

<Institution Name Partner>	Version: <1.0>
Name of the Component >	Date: 6/25/2014
Product Specification	

<Product Specification>

1. Introduction

[The introduction of the Vision document provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references and overview of this Vision document]

1.1 Purpose

[Specify the purpose of this Vision document]

1.2 General scope

[A brief description of the scope of this Vision document; what Project (s) it is associated with and anything else that is affected or influenced by this document]

1.3 References

[This subsection provides a complete list of all documents referenced in the Vision document. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information can be provided by an attachment or other docume]

1.4 Overview

[This subsection describes what the rest of the Vision document contains and explains how the document is organized]

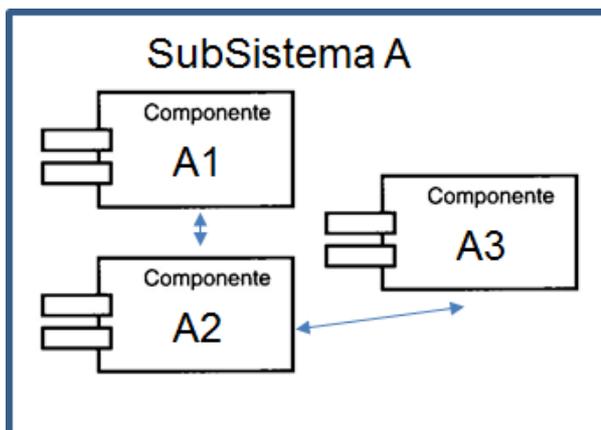
2. Positioning.

2.1 Project Overview

[Write a brief description of the comprehensively Project (system description or complete subsystem to be developed which will be inserted the component to be developed)]

2.2 Representation Product Graphic

[Provide a graphical representation of product being built. The technique / tool to be used should one set in the development plan (scope)]



3. Identification of Components Involved and Responsible

[Describe components that will be developed in the overall project scope]

<Institution Name Partner>	Version: <1.0>
Name of the Component >	Date: 6/25/2014
Product Specification	

3.1 Components

Component	Description	Responsible
<i>[Tell component]</i>	<i>[Write a brief description of the component]</i>	<i>[Tell the group, sector or organization responsible for developing the component]</i>

Note: You must include rows in the table above as there are components.

3.2 Environment

[Detail the environment where the product (or components) will be submitted]

3.3 Development Environment

[Detail the development environment as well as products and tools to be used]

3.4 Use Environment

[Identify and detail all the items that must be considered]

- *Tasks to be performed?*
- *Running time?*
- *Environmental restrictions?*
- *Among others ...*

3.5 Component description to be produced.

[Describe each component to be produced (components and subcomponents)]

3.5.1 < Component Name >

Description	<i>[Description of the component to be developed]</i>
Type	<i>[Qualify the component]</i>
Tasks to be performed by the component	<i>[Describe the functions that will be developed to perform the component]</i>
Interfaces	<i>[Description of interfaces with other components or environment held with the component]</i>
Assistance	<i>[Describe possible assistance of other components to carry out the tasks]</i>
Involvement	<i>[What is the degree component of commitment to the whole.]</i>
Comments	<i>[Relevant observations to be recorded.]</i>

Note: You must include a table for each component.

4. Product Features

[List and briefly describe the product features]

5. Restrictions

[Describes any internal constraints, external constraints or other dependencies]

6. Quality Ranges

[Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set]

<Institution Name Partner>	Version: <1.0>
Name of the Component >	Date: 6/25/2014
Product Specification	

7. Precedence and Priority

[Set the priority of the different system resources]

8. Other Product Requirements

[List applicable standards, hardware or platform requirements, performance requirements and the environmental standards]

8.1 Applicable Standards

[List all standards and regulations with which the product must conform. Among them, can include legal and regulatory (space standards, etc.) and quality and safety standards (ISO, CMMI)]

8.2 System Requirements

[Define any system requirements necessary to support the component. These may be included network platforms and host operating systems supported, configurations, memory, peripherals and external components]

8.3 Performance Requirements

[Use this section to detail performance requirements. Performance issues can include items such as load factors, communication skills, throughput, accuracy, and reliability or response times under a variety of loading conditions]

8.4 Environmental Requirements

[Detail environmental requirements as needed. For hardware-based systems, environmental issues may include temperature, shock, humidity, radiation, etc. For software applications, environmental factors can include usage conditions, the execution environment, resource availability, maintenance issues, and recovery and error handling]

9. File Requirements

[This section describes the documentation that should be developed to support the successful implementation and legal documents required by standards or models for your operation]

9.1 Legal documents

[Describe the documents, the purpose and content of each document to be produced and in which phase it will be produced and configuration control items, if any]

9.2 Other documents

[Describe the documents, the purpose and content of each document to be produced and in which phase it will be produced and configuration control items, if any]

<Institution Name Partner>

<Name of Project>
< Component Specification >
<Name of Component >
Versão <1.0>

<Author Name>

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

Revision History

Date	Verion	Description	Author
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Definitions, Acronyms and Abbreviations	5
1.4	References	5
1.5	Overview	5
2.	Component General Specification.	5
2.1	Description.	5
	2.1.1 Brief Description	5
	2.1.2 Component Graphical Representation	5
2.2	Goals	5
2.3	Metrics	5
	2.3.1 < measurement name of performance1>	6
	2.3.2 < measurement name of performance2>	6
2.4	Workflow	6
	2.4.1 Workflow Basics	6
	< Workflow step name >	6
	2.4.2 Alternative Workflows	6
	< Workflow step name >	6
2.5	Category	6
2.6	Risk	6
2.7	Possibilities	6
2.8	Process Owner	6
2.9	Special Requirements	6
	2.9.1 < Special Requirement Name 1>	6
	2.9.2 < Special Requirement Name 2>	6
2.10	Extension Points	6
	2.10.1 < extension 1 Point Name>	6
	2.10.2 < extension 2 Point Name>	6
2.11	Behavior	6
2.12	Usability	7
2.13	Reliability	7
2.14	Performance.	7
2.15	Scale Problems	7
3.	3. Distribution of Features (Modeling).	7
3.1	Hardware.	7
	3.1.1 Full Description.	7
	3.1.2 Hardware Graphic Representation.	7
	3.1.3 Hardware Model.	7
	3.1.4 Description of used tools.	7
	3.1.5 Detailed Description of Hardware.	7
	3.1.6 < Name artifact 1>	8
	3.1.7 Description of hardware interfaces	8
	3.1.8 Hardware Restrictions	8
3.2	Software.	8
	3.2.1 Full Description	8
	3.2.2 Software Graphic	8
	3.2.3 Software Model	8

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

3.2.4 Description of used tools.	8
3.2.5 Detailed Description of Software	8
3.2.6 < Functionality 1 Name>	8
3.2.7 Description of the Software interfaces	8
3.2.8 Software Restrictions	8
3.3 Integration.	8
3.3.1 Integration Full Description	8
3.3.2 Integration Environment	8
4. Solution Proposal.	8
4.1 NNN Solution Name	9
4.1.1 Software.	9
4.1.2 Part Hardware.	9
4.1.3 Prototype.	9
5. Solution Adopted.	9
6. Acquisition	9
6.1 Identification Supplier	9
6.2 Selection Criteria	9
6.3 Organization contract with the supplier	9

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

<Name of Component >

1. Introduction

[The introduction of the Specification Components provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references and overview of this Specification]

1.1 Purpose

[Specify the purpose of this Component Specification]

1.2 Scope

[A brief description of the scope of this Component Specification, and all that is affected or influenced by this document]

1.3 Definitions, Acronyms and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Component Specification. If there is a project glossary it should be included in this section]

1.4 References

[This subsection provides a complete list of all documents referenced elsewhere in the Component Specification. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information can be provided by an attachment or other documents]

1.5 Overview

[This subsection describes what the rest of the components Specification contains and explains how the document is organized]

2. Component General Specification.

2.1 Description.

2.1.1 Brief Description

[The description briefly discusses the role and the component's purpose]

[Describe the general factors that affect the component and requirements as:

- *Product perspective*
- *Product functions*
- *Restrictions*
- *Assumptions and dependencies*
- *Sets of requirements]*

2.1.2 Component Graphical Representation

[Using diagrams, models and other features that clearly represent the component, its functions, the subcomponents. Representation should allow independent view of the hardware and software component as well as the features that will be held by them]

2.2 Goals

[A statement of goals or measurable component objectives]

2.3 Metrics

[A specification of the metrics relevant to the component and the definition of the goals you want to

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

achieve using these metrics]

2.3.1 < measurement name of performance1>

[A brief description of the performance measure]

2.3.2 < measurement name of performance2>

[A brief description of the performance measure]

2.4 Workflow

[A textual description of the workflow represented by the component]

2.4.1 Workflow Basics

< Workflow step name >

[A brief description of the workflow step]

2.4.2 Alternative Workflows

< Workflow step name >

[A brief description of the workflow step]

2.5 Category

[Indicates the level of responsibility of the functions performed by the component in your environment, the 'central', 'method', 'management' or 'other']

2.6 Risk

[Specifies the operational risks of implementing the component]

2.7 Possibilities

[Describes the potential to improve the estimated component]

2.8 Process Owner

[Defines who the owner of the achieved results is]

2.9 Special Requirements

[Description of special component requirements]

2.9.1 < Special Requirement Name 1>

[A brief description of the special requirement]

2.9.2 < Special Requirement Name 2>

[A brief description of the special requirements 2]

2.10 Extension Points

[Extension points to the requirement]

2.10.1 < extension 1 Point Name>

[A definition of the location of the extension point in the flow]

2.10.2 < extension 2 Point Name>

[A definition of the location of the extension point in the flow]

2.11 Behavior

[This section expresses the general objectives of the behavior of the product that are not specific to a particular component]

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

2.12 Usability

[This section includes all the requirements that affect usability]

2.13 Reliability

[Reliability requirements (from the point of view of the product) should be specified here. Below are some suggestions:

- *Availability - specify the percentage of time that is available (xx.xx%), early hours of use, etc.*
- *Accuracy - specify precision (resolution) and accuracy (by some known standard) that are required in output]*

2.14 Performance.

[Performance characteristics should be summarized in this section. Include specific response times. When applicable, reference to names of related components.

- *Response time for a transaction (average, maximum)*
- *Transfer rate;*
- *Ability;*
- *Use of resources]*

2.15 Scale Problems

[Briefly list and describe briefly any information about the limitations or precautions that need to be considered]

3. 3. Distribution of Features (Modeling).

[Construction of activity models that explain the characteristics and behavior of party software and hardware component parts to be developed. In its construction the models must be used in the preparation and representation of the architecture of the features and functionality that the component must provide and contribute to its planning and construction. In general it implies the construction of graphical models that represent the artifacts of hardware and software components used and their interrelationships. The choice of technique and tool to be used must be made according to the characteristics of the component to be developed and the development of these characteristics and the environment involved it. Can be represented by using flowcharts, graphic languages, tools and techniques directed to the component to be produced.]

3.1 Hardware.

3.1.1 Full Description.

[The description should address all the actions to be resolved by the component hardware part]

3.1.2 Hardware Graphic Representation.

[Represent the hardware and all of its functionality, preferably through models and diagrams including interfaces with the software and other components]

3.1.3 Hardware Model.

[Prototype / hardware model with all the artifacts used]

3.1.4 Description of used tools.

[Description of tools to be used for modeling and prototyping activities of the component.]

3.1.5 Detailed Description of Hardware.

[Description of all items (artifacts) used to produce the hardware and the function of each item in the component]

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

3.1.6 < Name artifact 1>
[Artifact description]

3.1.6.1 Detailed artifact 1
[Description of all items (artifacts) used for hardware production and the function of each item in the component, include as many lines as necessary]

3.1.7 Description of hardware interfaces
[Detailed description of the hardware interfaces].

3.1.8 Hardware Restrictions
[Detailed description of the hardware restrictions and consequences]

3.2 Software.

3.2.1 Full Description
[The description should address all the actions to be resolved by the software part of the component]

3.2.2 Software Graphic
[Represent the software and all of its functionality, preferably through models and diagrams including interfaces with the hardware and other components]

3.2.3 Software Model
[Prototype / software model with all the artifacts used]

3.2.4 Description of used tools.
[Description of tools to be used for modeling and prototyping activity component.]

3.2.5 Detailed Description of Software
[Description of all items (artifacts) used for the production of software and the function of each item in the component]

3.2.6 < Functionality 1 Name>
[Description of feature]

3.2.6.1 Detailed Functionalit 1
[Description of how it should be implemented functionality]

3.2.7 Description of the Software interfaces
[Detailed description of the interfaces that the software does]

3.2.8 Software Restrictions
[Detailed description of the software restrictions and consequences]

3.3 Integration.

3.3.1 Integration Full Description
[The description should address all the actions to be taken for the perfect integration of component parts]

3.3.2 Integration Environment
[Describe The Necessary environment for the integration of component parts]

4. Solution Proposal.

[A description of the component solution proposal and the results of the evaluation of the prototype, for components that can develop different solutions for each proposed solution should be included by adding a

<Institution Name Partner>	Version: <1.0>
<Name of Component >	Date: <26/10/14>
Component Specification	

sub-item in item 4. (4.1, 4.2, 4.3 ..).]

4.1 NNN Solution Name

[Description of the component solution. Each solution create a new item where NNN must be a equencia numbering for the solution and the name should be an assignment for solution identification].

4.1.1 Software.

[Description of part of the software component solution].

4.1.2 Part Hardware.

[Description of the hardware component of the solution]

4.1.3 Prototype.

[Component of the prototype, HW, SW and complete].

5. Solution Adopted.

[A description of the component solution proposal and the results of the evaluation of the prototype, for components can be developed different solutions for each proposed solution should be included plus a sub-item in item 4. (4.1, 4.2, 4.3 ..).]

6. Acquisition

[Describe the approach adopted for acquisition]

6.1 Identification Supplier

[Component Supplier Description]

6.2 Selection Criteria

[Describe the items that were used for the vendor selection]

6.3 Organization contract with the supplier

[Indication and attachment of the supply contract]

<Name Partner Institution>

<Name of Project >

<Name of Component>

Test Project

Versão <1.0>

<Author Name>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Revision History

Date	Version	Description	Author
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Test Target	5
1.4	Terminology and Acronyms Document	5
1.5	References	5
1.6	Document Structure	5
2.	Evaluation Mission and Test Motivation	5
2.1	Detailed Information	5
2.2	Assessment Mission	6
2.3	Motivators of Tests	6
3.	Items target of Tests	6
4.	Outline of Planned Tests	6
4.1	Summary of Test Inclusions	6
4.2	Summary of Test Exclusions	6
5.	Approach to Testing	7
5.1	Technical and Test Types	7
5.1.1	Data Integrity Test and Database	7
5.1.2	Function Testing	8
5.1.3	User Interface Testing	9
5.1.4	Load Testing	9
5.1.5	Stress Testing	10
5.1.6	Volume Testing	12
5.1.7	Security Test	14
5.1.8	Tolerance Test Failures and Recovery	14
5.1.9	Configuration Test	16
5.1.10	Installation Testing	17
6.	Entry and Exit Criteria	18
6.1	Test Plan	18
6.1.1	Criteria Test Plan Entry	18
6.1.2	Criteria Test Plan Output	19
6.1.3	Criteria Suspension and Resumption	19
7.	Deliverables	19
7.1	Summary of Test Evaluation	19
7.2	Reporting of Test Coverage	19
7.3	Quality Reports Noticeable	19
7.4	Incident Logs and Change Requests	19
7.5	Regression Test Suite and Support Test Scripts	19
7.6	Additional Work Products	19
7.6.1	Detailed Results of Tests	19
7.6.2	Test Scripts Automated Functional Additional	19
7.6.3	Test Guidelines	19
8.	Test Workflow	20
9.	Environmental Needs	20

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

9.1	Basic Hardware Component	20
10.	Responsibilities, Staffing and Training Needs	20
10.1	People and Roles	20
10.2	Staffing and Training Needs	20
11.	Test Mark	20
12.	Procedures and Management Processes	21
12.1	Measurement and Evaluation of Test Extension	21
12.2	Evaluation of Deliverables of this Test Plan	21
12.3	Problem Reporting, selection to Solve them and Search for Solutions	21
12.4	Traceability Strategies	21
12.5	Approval and Closure	21

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

<Plan Test >

1. Introduction

1.1 Purpose

[The purpose of the Test Plan is to gather all the information necessary for planning and control of the test effort. It describes the approach to testing the software and is the top-level plan generated and used by managers to direct the test effort.]

This Test Plan related to the <Project Name> supports the following objectives:

- *[Identifies the items that should be targeted by the tests.]*
- *[Identifies the motivation and reasoning behind the test areas to be covered.]*
- *[Outlines the testing approach that will be used.]*
- *[Identifies the necessary resources and provides an estimate of the test efforts.]*
- *[List the deliverable elements of the test project]*

1.2 Scope

[Describe the test levels for example, Unit, Integration and System -- and types of test -- as Functionality, Usability, Reliability, Performance and Supportability -- that will be addressed by this Test Plan. It is also important to provide a general indication of the important areas to be excluded from the scope]

1.3 Test Target

[Provide a brief description of the target for which the Test Plan is being written. This will help document readers to identify whether it is really intended for their use and also help to prevent the document from being used inappropriately.]

Note: Often, the style and content of the document are changed in the target function]

1.4 Terminology and Acronyms Document

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Test Plan. If the project has a glossary, include a reference to the project Glossary in the References section]

1.5 References

[This subsection provides a list of documents referenced elsewhere in the Test Plan. Identify each document by title, version number (or report, if applicable), date, publishing organization or original author. Specify the sources from which the "official versions" of the references can be obtained]

1.6 Document Structure

[This subsection describes what the rest of the Test Plan contains and provides an introduction to how the rest of the document is organized]

2. Evaluation Mission and Test Motivation

[Provide an overview of the mission and motivation of tests to be conducted]

2.1 Detailed Information

[Provide a brief description of the grounds which justify the test effort defined by this Test Plan. Include information such as the solution of, the main problem, the main benefits of the solution, the planned architecture of the solution and a design brief history. When this information is defined in other documents, you can include references to these more detailed documents if appropriate. This section should only be about three to five paragraphs]

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

2.2 Assessment Mission

[Provide a brief statement outlining the mission of the current assessment effort. This statement might incorporate one or more concerns including:

- *find in as many bugs as possible*
- *find in important problems and assess the criticality of the identified risks*
- *list in the identified project risks*
- *ensuring standard*
- *verifying a specification (requirements, design or claims)*
- *listing of product quality and meet those involved*
- *listing of tests*
- *complying with the determinations of the process*
- *and so on*

Each mission provides a different context to the test effort and changes the way in which testing should be approached]

2.3 Motivators of Tests

[Provide a summary of the main elements that motivate the testing effort. Testing will be motivated by a number of factors such as: quality risks, technical risks, project risks, usage cases, functional requirements, non-functional requirements, design elements, fault or suspected bugs, changes request, etc.]

3. Items target of Tests

The list below identifies the items -- software, hardware and support elements of the product -- which have been identified as target of the test. This list represents the items to be tested.

[Provide a top-level list of the major items that will be subject to testing. This list should include items produced directly by the project development team and items that depend on those products; for example, the basic processing hardware, peripheral devices, operating systems, third party products or components, etc. It is recommended to group the list by category and assign the relative importance to each motivator]

4. Outline of Planned Tests

[This section provides a top-level summary of the tests to be performed. The summary provided here is an overview of the top level of the tests that will and will not be run]

4.1 Summary of Test Inclusions

[Provide a high level outline of the main current planned tests. Check what will be included in the plan and record what is not explicitly included in the section entitled Summary of Test Inclusions]

4.2 Summary of Test Exclusions

[Provide a high level outline of the possible tests that could have been conducted, but were explicitly excluded from this plan. If you are not implementing or enforcing a kind of test, clearly inform that the test will not be run or implemented and justify. The following are examples of reasons that may be used:

- *"These tests do not contribute to the evaluation mission"*
- *"There are not enough resources to run these tests"*
- *"These tests are unnecessary because of the tests performed by xxx"*

According to a heuristic point of view, if you think it is perfectly conceivable that one of the members of your audience expects that a certain test aspect is included and if you do not want or can not include it, justify their exclusion. If the team agrees that the exclusion is obvious, you will not probably list it]

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

5. Approach to Testing

[This section presents the recommended strategy to create and implement the necessary tests]

5.1 Technical and Test Types

5.1.1 Data Integrity Test and Database

[When databases and database processes Should be tested as an independent subsystem. This test should test the subsystems without the test objective User Interface to interface with the data. Identify the tools and techniques that may exist to support the tests identified in the following table]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
Technique Objective:	<i>[Try processes and methods of accessing independent of the UI database so you can observe and record incorrect target behaviors or the existence of corrupted data]</i>
Technique:	<ul style="list-style-type: none"> • <i>[Shoot each process and method of access to the database, propagating data requests or valid and invalid data in each.</i> • <i>Inspect the database to ensure that data has been distributed as planned and that all database events occurred properly, or review the returned data to ensure that the correct data was retrieved for the right reasons]</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe, the test results accurately. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> • <i>Test Script Automation Tool</i> • <i>restorative imager and the basic configuration</i> • <i>Backup and recovery tools</i> • <i>installation of monitoring tools (registry, hard disk, CPU, memory, etc.)</i> • <i>tools and SQL database utilities</i> • <i>data generation tools]</i>
Success Criteria:	<i>[The technique supports the testing of all key processes and methods of access to database]</i>
Special Remarks:	<ul style="list-style-type: none"> • <i>[The tests may require drivers or a development environment.</i> • <i>Processes should be invoked manually.</i> • <i>Small or minimum size databases Should be used (with a limited number of records) to increase the visibility of any non-acceptable events]</i>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

5.1.2 Function Testing

[The function test objective should focus on all test requirements that can be directly associated with the functions and business rules. The goal of this test is to verify the proper acceptance, processing and data recovery, and appropriate implementation of business rules. This type of test is based on black box techniques; ie verify the application and its internal processes interacting with the application through the Graphical User Interface (GUI) and analyzing the output or results. The following table identifies a summary of the recommended test for each application]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
Technique Objective:	<i>[Try to test the purpose of functionality, including navigation, data input, processing and recovery of data in order to observe and log target behavior]</i>
Technique:	<p><i>[Run the resources and flows or functions of each use case scenarios, using valid and invalid data to verify that:</i></p> <ul style="list-style-type: none"> <i>• the expected results occur when valid data is used</i> <i>• the appropriate error or warning messages will be displayed when used invalid data</i> <i>• each business rule is properly applied]</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the software and hardware component part]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> <i>• Test Script Automation Tool</i> <i>• restorative imager and the basic configuration</i> <i>• Backup and recovery tools</i> <i>• installation of monitoring tools (registry, hard disk, CPU, memory, etc..)</i> <i>• data generation tools]</i>
Success Criteria:	<p><i>[The technique supports the test:</i></p> <ul style="list-style-type: none"> <i>• all major use case scenarios</i> <i>• all key features]</i>
Special Remarks:	<i>[Identify or describe the items or issues (internal or external) that influence the implementation and execution of the function test]</i>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

5.1.3 User Interface Testing

[The User Interface (UI) testing verifies a user's interaction with the software. The UI test goal is to ensure that the UI provides the user access to adequate and navigation through the test objective functions. In addition, UI testing ensures that the objects within the UI work as expected and conform to corporate or industry standards]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
Technique Objective:	<p><i>[Try the following to observe and record compliance with standards and target behavior:</i></p> <ul style="list-style-type: none"> • <i>Navigating the purpose of testing to verify that it reflects the requirements and business functions, including window to window and field to field, and the use of access methods (tab keys, mouse movements, accelerator keys).</i> • <i>The objects and features may be experienced as, for example, size, position, state, and focus]</i>
Technique:	<i>[Create or modify tests for each function and check the appropriate object states]</i>
Strategies:	<i>[Describe one or more strategies that can be used by the technique to observe, the test results accurately. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination]</i>
Required Tools:	<p><i>[The technique requires the Test Script Automation Tool.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the software and hardware component parts]</i></p>
Success Criteria:	<i>[The technique that supports the testing of each screen or main window that will be widely used by the end user]</i>
Considerações Especiais:	<i>[Identify or describe the items or issues (internal or external) that influence the implementation and test execution]</i>

5.1.4 Load Testing

[The load test is a performance test that reveals the purpose of testing the different workloads to measure and assess skills and behaviors of performance goals, in order to verify if it continues to function properly with those various workloads]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
-----------------	---

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Technique Objective:	<i>[Try functions or transactions referred to in various workload conditions to observe and log target behavior and component performance data.]</i>
Technique:	<ul style="list-style-type: none"> • <i>[Use Transaction Test Scripts developed for Business Cycle Testing or working as a base, but remember to remove the iterations and unnecessary delays.</i> • <i>Modify the data files in order to increase the number of transactions or modify the tests in order to increase the number of times each transaction occurs.</i> • <i>The Workloads should include Peak loads (eg Daily, Weekly, Monthly, etc.).</i> • <i>The Workloads should represent average loads and peak loads.</i> • <i>The Workloads should represent snapshots and Sustained Peaks.</i> • <i>The Workloads must be performed with different Environment Settings Test]</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to accurately observe the test results. The strategy combines the method by Which the observation can be made and the characteristics of specific outcome that Indicate probable success or failure. Ideally, the strategies are self-analysis, Allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the Risks inherent in automated results determination.</i></p> <p><i>The strategy shouldnt take into consideration items and results that will be used for the the inter-relationships between they software and hardware component parts]</i></p>
Required Tools:	<p><i>[The technique requires the Following tools:</i></p> <ul style="list-style-type: none"> • <i>Test Script Automation Tool</i> • <i>Control tool and Transaction Load Scheduling</i> • <i>installation of monitoring tools (registry, hard disk, CPU, memory, etc.)</i> • <i>resource constraint tools (for example, Canned)</i> • <i>data generation tools]</i>
Success Criteria:	<i>[The technique supports the Workload Emulation test, which is the successful emulation of the workload without any failures due to test implementation problems]</i>
Special Considerations:	<ul style="list-style-type: none"> • <i>[Identify or describe the items or issues (internal or external) that influence the implementation and test execution]</i>

5.1.5 Stress Testing

[The stress test is a performance test implemented and executed to understand how component failures occur due to conditions at or outside the limits of the expected tolerances. Typically, this involves low resources or resource competition. The conditions of limited resources reveal how failures occur in the test objectives that are not apparent in normal conditions]

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

[Note: The transactions listed in the following table are logical business transactions]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
Technique Objective:	<p><i>[Try functions of the test focus on the following stress conditions to observe and log target behavior that identifies and documents the conditions that cause the component to stop working properly:</i></p> <ul style="list-style-type: none"> <i>• little or no memory available on the server (RAM and persistent storage space)</i> <i>• maximum physical capacity of real connection or simulation of clients</i> <i>• multiple users performing the same transactions against the same data or accounts</i> <i>• set or volume of transactions that generate "overhead" (see Performance Profiling above)]</i>
Technique:	<ul style="list-style-type: none"> <i>• [Use the Load test and Performance Profiling.</i> <i>• To test limited resources, tests should be run on a single machine, and RAM and persistent storage space on the server should be reduced or limited.</i> <i>• For the remaining stress tests are to be used multiple clients, by performing the same additional tests or tests to produce joint or the volume of transactions in the worst case]</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> <i>• Test Script Automation Tool</i> <i>• Control tool and Transaction Load Scheduling</i> <i>• installation of monitoring tools (registry, hard disk, CPU, memory, etc.)</i> <i>• resource constraint tools (for example, Canned)</i> <i>• data generation tools]</i>
Success Criteria:	<i>[The technique supports the Stress emulation test. The component may be emulated in an effective manner on one or more conditions defined as stress conditions, can be captured and an observation of the state of the component resulting during and after the condition has been emulated]</i>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Special Considerations:	<ul style="list-style-type: none"> • <i>[To force the network, you may need to overload the network tools with messages or packets.</i> • <i>Persistent storage used for the component should be reduced temporarily in order to restrict the space available for the database to develop.</i> • <i>Synchronize the simultaneous clients accessing of the same records or data accounts]</i>
-------------------------	---

5.1.6 Volume Testing

[The volume of test reveals the purpose of large volumes of test data to determine whether limits are reached that will cause the software to crash. This test also identifies the volume or the maximum continuous load test objective can sustain during a given period of time. For example, if the test subject is processing a set of database records to generate a report, a Volume Test will use a large bank of test data and verify that the software behavior was normal and produced the correct report]

Part Component:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Technique Objective:	<p><i>[Try the purpose of the test in the following high volume scenarios to observe and log target behavior:</i></p> <ul style="list-style-type: none"> • <i>The maximum (actual or physically capable) connected clients, or simulated, all performing the same business function (performance) in the worst case, for an extended period of time.</i> • <i>has been reached maximum size of the database (actual or scaled) and multiple queries or report transactions are executed simultaneously]</i>
Technique:	<ul style="list-style-type: none"> • <i>[Use the Load Test and Performance Profiling.</i> • <i>multiple clients shouldnt be used, by performing the same tests or additional tests to produce joint or the volume of transactions in the worst case (see Stress Test) for a long period of time.</i> • <i>Set the maximum size be of the database (real, scale or filled with representative data) and will be used several clients to run queries and report transactions simultaneously for extended periods of time]</i>
Strategies:	<p><i>[Outline one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are autoverificadas, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part]</i></p>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> • <i>Test Script Automation Tool</i> • <i>Control tool and Transaction Load Scheduling</i> • <i>installation of monitoring tools (registry, hard disk, CPU, memory, etc.)</i> • <i>resource constraint tools (for example, Canned)</i> • <i>• Data-generation tools]</i>
Success Criteria:	<p><i>[The technique supports the Volume emulation test. It is possible to emulate efficiently, large amounts of users, data, transactions or other aspects of the system volume and used in an observation can be captured on the component status changes during the test volume]</i></p>
Special Considerations:	<p><i>[What time period would be considered acceptable for high volume conditions, as noted above?]</i></p>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

5.1.7 Security Test

Based on the required security level, perform the functions]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
Technique Objective:	<p><i>[Try the test objective under the following conditions to observe and log target behavior:</i></p> <ul style="list-style-type: none"> • <i>Security at the application level: an actor can access only those functions or data for which their user type is provided permissions.</i> • <i>Security at the component level: only the component features and applications are allowed to access them]</i>
Technique:	<ul style="list-style-type: none"> • <i>[Security at the application level: Identify and list each user type and the functions or data for which each of them has access permission]</i> <ul style="list-style-type: none"> ○ <i>Create tests for each user type and verify each permission by creating transactions specific to each type of user.</i> ○ <i>o Modify the type of user and run the tests again to the same users. In each case, make sure the additional functions or data are correctly available or are denied access.</i> • <i>Access at the component level: [See Special Considerations below]</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part.]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> • <i>Test Script Automation Tool</i> • <i>Tools for research and for breach of security by "hackers"</i> <ul style="list-style-type: none"> • <i>Administration Tools Operating System Security]</i>
Success Criteria:	<i>[Technical support for each type of well-known actor, the test of the appropriate functions or data affected by security settings]</i>
Special Considerations:	<i>[Identify or describe the items or issues (internal or external) that influence the implementation and test execution]</i>

5.1.8 Tolerance Test Failures and Recovery

[The fault-tolerance test and recovery ensures that the purpose of the test can tolerate and recover, effectively,

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

a series of hardware failures, software or network with undue loss of data or data integrity]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
Technique Objective:	<p><i>[Simulate the failure conditions and exercise the recovery processes (manual and automated) to restore the known and desired state of the database, application and component. The following types of conditions are included in the test to observe and log target behavior after recovery:</i></p> <ul style="list-style-type: none"> <i>• power interruption to the customer</i> <i>• power interruption to the server</i> <i>• breakdown in communication through network servers</i> <i>• incomplete cycles (filtering processes interrupted data synchronization processes interrupted data)</i> <i>• pointers or invalid database keys</i> <i>• invalid or corrupted data elements in the database]</i>
Technique:	<p><i>[The functional tests and used as a basis to create a series of transactions to support fault tolerance and recovery tests and especially to define the tests to be performed to determine whether the recovery was successful.</i></p> <ul style="list-style-type: none"> <i>• Power interruption to the client: switch off the PC.</i> <i>• Power interruption to the server: simulate or initiate server shutdown procedures.</i> <i>• Interruption via network servers: simulate or initiate communication loss with the network (physically disconnect communication wires or power down the server or the network routers).</i> <p><i>After the above conditions or simulated conditions are achieved, additional transactions should be executed, and when the state this second test point is reached, recovery procedures should be invoked.</i></p> <p><i>Testing for incomplete cycles uses the same technique described above, except for the actual database processes, which should be aborted or prematurely terminated.</i></p> <p><i>The test of the following conditions is met requires a known state database.</i></p> <p><i>Various fields, pointers and database keys should be corrupted manually and directly in the database (via database tools). Additional transactions should be executed using the Application Function tests and should be performed complete cycles]</i></p>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part.]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> • <i>restorative imager and the basic configuration</i> • <i>installation of monitoring tools (registry, hard disk, CPU, memory, etc.)</i> • <i>Backup and recovery tools]</i>
Success Criteria:	<p><i>[The technique supports the test:</i></p> <ul style="list-style-type: none"> • <i>One or more simulated disasters involving one or more combinations of the application, database and component.</i> • <i>• One or more simulated recoveries involving one or more combinations of the application, database and component in a known state desired]</i>
Special Considerations:	<ul style="list-style-type: none"> • <i>[The impairment test is highly invasive. Procedures to disconnect cabling (simulating power loss or communication) may not be desirable or feasible. Alternative methods may be required, for example, diagnostic software tools.</i> • <i>• It will take Component Resources (or Computer Operations), Database and Network Groups.</i> • <i>• These tests should be performed after working hours or on an isolated machine]</i>

5.1.9 Configuration Test

[Configuration testing verifies the operation of the purpose of the test in different software configurations and hardware]

Part Component:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Technique Objective:	<i>[Try the test focus on hardware and software configurations required in order to observe and log target behavior in different configurations and identify changes in configuration state]</i>

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Technique:	<ul style="list-style-type: none"> • <i>[Use Function Test Scripts.</i> • <i>Open and close various related software other than the test objective (for example, Microsoft Excel and Word applications) as part of the test or before the test.</i> • <i>Execute selected transactions to simulate actors interacting with software that is the purpose of the test and the other than the test objective.</i> • <i>Repeat the above process, minimizing the available conventional memory on the client workstation]</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are self-analysis, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> • <i>restorative imager and the basic configuration</i> • <i>installation of monitoring tools (registry, hard disk, CPU, memory, etc.)]</i>
Success Criteria:	<p><i>[The technique supports the testing of one or more combinations of the test target items que are executed in supported environments and Implementation expected]</i></p>
Special Considerations:	<ul style="list-style-type: none"> • <i>[What software, other than the purpose of the test, it is necessary, is available and accessible on the desktop?</i> • <i>What applications are typically used?</i> • <i>What data are the applications running; for example, a large spreadsheet opened in Excel or a 100-page document in Word?]</i>

5.1.10 Installation Testing

[Installation testing has two purposes]

Part Component:	<i>[Describe which part of the component on which this test is applied, part of hardware or software]</i>
-----------------	---

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Technique Objective:	<p><i>[Try installing the test objective on each hardware configuration required under the following conditions to observe and record the behavior of the installation and changes in the configuration state:</i></p> <ul style="list-style-type: none"> • <i>new installation: a new machine, which has never been previously installed <Project Name></i> • <i>update: a machine that was previously installed <Project Name>, same version</i> • <i>• update: a machine that was previously installed <Project Name> in an older version]</i>
Technique:	<ul style="list-style-type: none"> • <i>[Develop automated or manual scripts to validate the condition of the target machine. the new, never installed the same older version or version already installed</i> • <i>Start or run the installation.</i> • <i>Using a predetermined subset of Function Test scripts, run the transactions</i>
Strategies:	<p><i>[Describe one or more strategies that can be used by the technique to observe accurately, the test results. The strategy combines the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, the strategies are autoverificadas, allowing automated tests to make an initial assessment of success or failure. However, be careful to mitigate the risks inherent in automated results determination.</i></p> <p><i>The strategy should take into consideration items and results that will be used for the inter-relationships of the results to the party software and hardware component part.]</i></p>
Required Tools:	<p><i>[The technique requires the following tools:</i></p> <ul style="list-style-type: none"> • <i>restorative imager and the basic configuration</i> • <i>installation of monitoring tools (registry, hard disk, CPU, memory, etc.)]</i>
Success Criteria:	<p><i>[The technique supports the product installation test developed in one or more installation settings.]</i></p>
Special Considerations:	<p><i>[That transaction <Project Name> should be selected to be a test showing that the application of <Project Name> has been successfully installed and is not missing any part of the main software??]</i></p>

6. Entry and Exit Criteria

6.1 Test Plan

[Specify this section the test plan on the part hardware and part software component may be made together or create a new subsection eg 6.1 Hardware Test Plan, Plan 6.2 Software testing, one can also create a subsection for the complete component already integrated eg 6.3. Component test plan. The subsections need not be the same individual may be assigned to each criterion in accordance with the needs and characteristics]

6.1.1 Criteria Test Plan Entry

[Specify the criteria that will be used to determine whether the execution of the Test Plan may be initiated]

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

6.1.2 Criteria Test Plan Output

[Specify the criteria that will be used to determine whether the implementation of the Test Plan has been completed or if the continued implementation will not be advantageous]

6.1.3 Criteria Suspension and Resumption

[Specify the criteria that will be used to determine whether testing should be prematurely suspended or terminated before the plan has been fully implemented. Also specify what criteria the tests may be restarted]

7. Deliverables

[In this section, list the various artifacts that will be created by the test effort and that will be released products useful to various stakeholders of the test effort. Do not list all work products; list only those who provide tangible direct benefits to those involved and for measuring the success of the test effort.

For this section take consider the item's guidelines 6.1]

7.1 Summary of Test Evaluation

[Please provide a brief summary of the form and content of the test evaluation summaries and indicate how often they are generated]

7.2 Reporting of Test Coverage

[Provide a brief summary of the form and content of the reports used to measure the extent of testing and indicate how often they are generated Provide an indication as to the method and tools used to record, measure and report the extent of testing]

7.3 Quality Reports Noticeable

[Provide a brief summary of the form and content of the reports used to measure the perceived quality of the product and indicate how often they are generated. Provide an indication as to the method and tools used to record, measure and report the perceived quality of the product. You can include analyzes of incidents and change requests throughout the Test Coverage]

7.4 Incident Logs and Change Requests

[Provide a brief outline of the method and tools used to record, track and manage incidents of tests, change requests and their associated status]

7.5 Regression Test Suite and Support Test Scripts

[Please provide a brief summary of the features of the tests that will be distributed to enable continuous regression testing of subsequent builds of the product in order to help detect regressions in product quality]

7.6 Additional Work Products

[In this section, identify the work products that are optional or that should not be used to measure or evaluate the successful execution of the Test Plan]

7.6.1 Detailed Results of Tests

[This is a Microsoft Excel spreadsheet listing the set of results determined for each test case or refers to the repository of test records and results kept for a certain product specialized tes]

7.6.2 Test Scripts Automated Functional Additional

[These scripts consist of a set of source code files for automated test scripts or repository of source code and compiled executables for test scripts maintained by the test automation product]

7.6.3 Test Guidelines

[The Test Guide covers a wide range of categories, including Test-Idea Catalogs, Appropriate Practices Guidelines, Test Patterns, Templates errors and failures, Automation Design Standards, etc.]

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

8. Test Workflow

[Provide a summary of the workflow to be followed by the Test team in the development and execution of this Test Plan. For this section take consider the item's guidelines 6.1]

9. Environmental Needs

[This section presents non-human resources needed to Test Plan (Hardware and Software)]

9.1 Basic Hardware Component

[The following tables show sets the component resources needed to test effort presented in this Test Plan.

This that this section should be completed soon to time as needed]

Component Features		
Resource	Amount	Name and Type

10. Responsibilities, Staffing and Training Needs

[This section provides the necessary resources to address the test effort described in the Test Plan - the main responsibilities and sets of knowledge or skills required of these resources. (Hardware and Software)]

10.1 People and Roles

This table shows the assumptions relating to the profile of the test team effort.

[Note: Add or delete items as needed.]

Human Resources		
Role	Minimum Resources Recommended (number of allocated full-time roles)	Specific Responsibilities or Comments

10.2 Staffing and Training Needs

This section summarizes how to approach the profile of the team and the training of professionals who will fill the test papers in the project.

[The way to approach the profile of the team and the training of professionals varies from project to project. If this section integrate a Master Test Plan, indicate which parts of the project life cycle will require different skills and a different number of team members]

11. Test Mark

[Identify the major milestones of programming that define the Test Effort context. (Hardware and Software)]

<Name Partner Institution>	Version: <1.0>
<Name of Component>	Date: <25/06/2014
Test Project	

Mark	Start Date Planned	Start Date Real	End Date Planned	Date Real End
Test is finished				

12. Procedures and Management Processes

[Summarize the processes and procedures to be used when problems arise in the Test Plan and its implementation (Hardware and Software)]

12.1 Measurement and Evaluation of Test Extension

[Summarize the measurement and evaluation process to be used to track the extent of testing]

12.2 Evaluation of Deliverables of this Test Plan

[Summarize the evaluation process to review and accept the deliverables of this Test Plan]

12.3 Problem Reporting, selection to Solve them and Search for Solutions

[Set to the problems relating to proceedings shall be reported, as people will be selected to solve them and the process to be followed to arrive at a solution]

12.4 Traceability Strategies

[Consider appropriate traceability strategies regarding:

- *Test Coverage in relation to the specifications - enables the measurement of the extent of testing*
- *Test Motivations - enable the evaluation of the relevance of tests to help determine whether they should be kept or not*
- *Software Design Elements - enable the tracking of subsequent design changes that require tests run again or be canceled*
- *Resulting Change Requests - make the tests that discovered the need for change are identified and run again to verify that the change request was made successfully]*

12.5 Approval and Closure

[Summarize the approval process and list the positions (and the names of the current occupants) which will initially approve the plan and finish with the satisfactory implementation of the plan]

<Institution Name Partner>

<Name of Project >

<Name of Component >

Approval plan

Versão <1.0>

<Author Name>

<Institution Name Partner>	Versão: <1.0>
<Name of Component >	Data: <2014-06-25>
Approval plan	

Revision History

Date	Version	Description	Author
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Institution Name Partner>	Versão: <1.0>
<Name of Component >	Data: <2014-06-25>
Approval plan	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Responsibilities	4
3.	Product Acceptance Tasks	4
3.1	Product Acceptance Criteria	4
3.2	Audit of Physical Configuration	4
3.3	Audit Functional Configuration	4
3.4	Programming	4
4.	Resource Requirements	4
4.1	Hardware Requirements	4
4.2	Software Requirements	4
4.3	Documentation Requirements	5
4.4	Personnel Requirements	5
4.5	Test Data Requirements	5
4.6	Other Requirements	5
5.	Troubleshooting and Corrective Action	5
6.	Product Acceptance Environment	5
7.	Identification and Review of the required Artifacts	5
8.	Tools, Techniques and Methodologies	5
9.	Cycle Assessment	5
9.1	Process	5
9.2	Life Cycle Stages	5
9.3	Project Planning	5
9.4	Problems and Resolutions	5
9.5	People	5
9.6	Produced Component	5

<Institution Name Partner>	Versão: <1.0>
<Name of Component >	Data: <2014-06-25>
Approval plan	

Approval plan

1. Introduction

[The introduction of the Certification Plan provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this Approval Plan]

1.1 Purpose

[Specify the purpose of this Approval Plan]

1.2 Scope

[A brief description of the scope of this Plan Approval; Projects to which it is associated with and anything else that is affected or influenced by this document]

1.3 Definitions, Acronyms and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Approval Plan. This information can be provided by referring to the project Glossary]

1.4 References

[This subsection provides a complete list of all documents referenced in the Certification Plan. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information can be provided by an attachment or other document]

1.5 Overview

[This subsection describes what the rest of the Approval Plan and explains how the document is organized]

2. Responsibilities

[Clearly identify the client's and the development team's responsibilities in preparing and implementing the product acceptance activities]

3. Product Acceptance Tasks

3.1 Product Acceptance Criteria

[Identify the objective criteria to determine the acceptability of the products and deliverable artifacts from this project. These criteria must be formally agreed between the customer and the development team]

3.2 Audit of Physical Configuration

[Identify and list the artifacts resulting from this project to be delivered and accepted by the customer.]

3.3 Audit Functional Configuration

[For each artifact identified in the Physical Configuration Audit, identify the methods of evaluation and the level of detail that will be used to determine if the device meets the product acceptance criteria. The methods may include software running tests, product demonstrations, documentation reviews etc.]

3.4 Programming

[A schedule indicating the beginning and end of each product acceptance tasks, including preparation and configuration activities]

4. Resource Requirements

4.1 Hardware Requirements

[For example, hardware, interface equipment, firmware items]

4.2 Software Requirements

[For example, operating systems, compilers, test drivers, test data generators]

<Institution Name Partner>	Versão: <1.0>
<Name of Component >	Data: <2014-06-25>
Approval plan	

4.3 Documentation Requirements

[For example, test documentation, technical references]

4.4 Personnel Requirements

[For example, development of team members, customer representatives, third-party authorities]

4.5 Test Data Requirements

[For example, the size, type and composition of data to support the acceptance tests]

4.6 Other Requirements

[For example, special equipment]

5. Troubleshooting and Corrective Action

[This section describes the procedures to report and resolve problems identified during the Product Acceptance activities. They are usually covered including the artifact Plan Troubleshooting by reference]

6. Product Acceptance Environment

[Describe plans to set up the product acceptance environment]

7. Identification and Review of the required Artifacts

[Based on the description of the Functional Configuration Audit, identify the evaluation of each individual artifact to be held. For each device, list the type of evaluation (test, review etc.) and objectives]

Note: A Test Case artifact for each test identified here will be prepared]

8. Tools, Techniques and Methodologies

[A list of all tools, specific techniques and methodologies to be used in fulfilling the Product Acceptance activities]

9. Cycle Assessment

[Analysis and description of the progress of all phases and cycle activities and suggestions for improvement to other cycles.]

9.1 Process

[Opinion related to the development process]

9.2 Life Cycle Stages

[Opinion related to the activities that were defined in each life cycle phase]

9.3 Project Planning

[Opinion related to the overall project planning.]

9.4 Problems and Resolutions

[Opinion related to the problems that arose, solution flexibility, quality of solutions, etc..]

9.5 People

[Opinion related to the people involved with the project, commitment, initiative, teamwork, training needs, etc..]

9.6 Produced Component

[Opinion related to the component quality produced in compliance with the requirements, outputs throughout the process, etc.]

<Institution Name Partner>

< Name of Project>

< Name of Component>

Configuration Management

Version <1.0>

<Author Name>

<Institution Name Partner>	Version: <1.0>
< Name of Component>	Date: <24/06/2014>
Configuration Management	

Revision History

Date	Version	Description	Author
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

<Institution Name Partner>	Version: <1.0>
< Name of Component>	Date: <24/06/2014>
Configuration Management	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	Referências	4
1.5	Overview	4
2.	Software Configuration Management	4
2.1	Organization, Responsibilities, and Interfaces	4
2.2	Configuration Control and Change	4
2.2.1	Processing and Change Request Approval	4
2.2.2	Responsible for Change Control	4
2.2.3	Storage Mechanism and Change Control.	4
2.2.4	Change Control Reports.	4
2.3	Configuration Items	5
2.3.1	Development Environment	5
2.3.2	Produced Artifacts.	5
2.3.3	Software Items.	5
2.3.4	Hardware Items.	5
3.	Subcontractors and Suppliers Control	5
3.1.1	Software Items.	5
3.1.2	Hardware Items.	5
4.	Traceability Matrix	5

<Institution Name Partner>	Version: <1.0>
< Name of Component>	Date: <24/06/2014>
Configuration Management	

Configuration Management

1. Introduction

[The introduction of Configuration Management provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this Configuration Management.]

1.1 Purpose

[Specify the purpose of Configuration Management]

1.2 Scope

[A brief description of the scope of Configuration Management; the model to which it is associated with and anything else that is affected or influenced by this document]

1.3 Definitions, Acronyms and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Configuration Management. This information can be provided by reference to the project Glossary]

1.4 Referênces

[This subsection provides a complete list of all documents referenced in the Configuration Management. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information can be provided by an attachment or other document]

1.5 Overview

[This subsection describes what the rest of the Configuration Management is and explains how the document is organized.]

Note that the configuration management must provide all phases of the process.]

2. Software Configuration Management

2.1 Organization, Responsibilities, and Interfaces

[Describe the activities and who will be responsible for implementation.]

2.2 Configuration Control and Change

[Describe any point of the project life cycle and product baselines that should be established. The most common baselines would be at the end of each stage of Inception, Elaboration, Construction and Transition. They can also be generated at the end of iterations which have occurred within the various phases or even more frequently]

2.2.1 Processing and Change Request Approval

[Describe the process by which problems and changes are submitted, reviewed and willing]

2.2.2 Responsible for Change Control

[Describe the person in charge of the procedures for processing change requests and approvals to be followed]

2.2.3 Storage Mechanism and Change Control.

[Describe how when and the controls will be maintained, type of file (baseline), support software when]

2.2.4 Change Control Reports.

[Describe the reports to be prepared for analysis and monitoring of changes]

<Institution Name Partner>	Version: <1.0>
< Name of Component>	Date: <24/06/2014>
Configuration Management	

2.3 Configuration Items

[Indications of items to be controlled]

2.3.1 Development Environment

[List all items from development environment that require monitoring]

- *Equipment.*
- *Software Products*
- *Among others...]*

2.3.2 Produced Artifacts.

[List all devices to be controlled by Configuration Management when produced during the project]

- *Product Specification.*
- *Component Specification*
- *Among others...]*

2.3.3 Software Items.

[List all artifacts that can be controlled when produced]

- *Source Program.*
- *Executable.*
- *Software Model*
- *Among other...]*

2.3.4 Hardware Items.

[List all artifacts that can be controlled when produced]

3. Subcontractors and Suppliers Control

[Describes how the component acquired or developed outside the project environment will be incorporated also identifies your configuration control points]

3.1.1 Software Items.

[List all artifacts that can be controlled when produced]

- *Artifacts.*

3.1.2 Hardware Items.

[List all artifacts that can be controlled when produced]

- *Component.*

4. Traceability Matrix

[Describe how the traceability matrix will be organized, addressing the storage medium, addressing. You must define what information should be in the form of matrices and traceability. The array can be organized in configuration control tools in Word or Excel file type, or other means]

Anexo A – Estudio de caso INPE

INPE - Instituto Nacional de Pesquisas Espaciais

Divisão de Desenvolvimento de Sistemas de Solo
Especificação de Componente
Software de Controle - Conjunto de Medida de
Velocidade
Versão <1.0>

Magda Miyashiro

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

Índice Analítico

1.	Introdução	5
1.1	Finalidade	5
1.2	Escopo	5
1.3	Definições, Acrônimos e Abreviações	5
1.4	Referências	6
1.5	Visão Geral	6
2.	Especificação Geral do Componente.	6
2.1	Descrição.	7
2.1.1	Breve Descrição	7
2.1.2	Representação Gráfica do Componente	7
2.2	Metas	8
2.3	Métricas	8
2.3.1	<nome da medida de desempenho1>	8
2.3.2	<nome da medida de desempenho2>	8
2.4	Fluxo de Trabalho	8
2.4.1	Fluxo de Trabalho Básico	8
	<nome do passo do fluxo de trabalho>	8
2.4.2	Fluxos de Trabalho Alternativos	8
	<nome do passo do fluxo de trabalho>	8
2.5	Categoria	8
2.6	Risco	8
2.7	Possibilidades	9
2.8	Proprietário do Processo	9
2.9	Requisitos Especiais	9
2.9.1	<Nome do Requisito Especial 1>	9
2.9.2	<Nome do Requisito Especial 2>	9
2.10	Pontos de Extensão	9
2.10.1	<Nome do Ponto de Extensão1>	9
2.10.2	<Nome do Ponto de Extensão2>	9
2.11	Comportamento	9
2.12	Usabilidade	9
2.13	Confiabilidade	9
2.14	Desempenho.	9
2.15	Problemas de Escala	9
3.	Distribuição das Funcionalidades (Modelagem).	10
3.1	Hardware.	10
3.1.1	Descrição Completa.	10
3.1.2	Representação Gráfica do Hardware.	10
3.1.3	Modelo do Hardware.	10
3.1.4	Descrição das ferramentas utilizadas.	10
3.1.5	Descrição detalhada do Hardware.	10
	REQUISITOS DE HARDWARE	12
	REQHW_01	12
	REQHW_02	12
	REQHW_03	12
	REQHW_04	12
	3.1.6 <Nome do artefato1>	12
	3.1.7 Descrição das interfaces do Hardware	12

<i>INPE - Instituto Nacional de Pesquisas Espaciais</i>	<i>Version:</i> <1.0>
Software de Controle - Conjunto de Medida de Velocidade	<i>Date:</i> <06/08/14>
Especificação do Componente	

3.2	Software.	12
3.2.1	Descrição Completa	12
3.2.2	Representação Gráfica do Software	14
3.2.3	Modelo do Software	17
3.2.4	Descrição das ferramentas utilizadas.	17
3.2.5	Descrição detalhada do Software	18
3.2.6	<Nome da funcionalidade1>	20
3.2.7	Descrição das interfaces do Software	20

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

Software de Controle - Conjunto de Medida de Velocidade

1. Introdução

[A introdução da Especificação Componentes fornece uma visão geral do documento inteiro. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral desta Especificação]

Este documento descreve a elaboração do software de controle para o Conjunto de Medida de Velocidade de satélites, com uma abordagem baseada na Linguagem de Modelagem Unificada (UML) utilizando o Rational Unified Process (RUP) de modo customizado.

1.1 Finalidade

[Especifique a finalidade desta Especificação de Componentes]

1.2 Escopo

[Uma breve descrição do escopo desta Especificação de Componentes, ela e de tudo o que for afetado ou influenciado por este documento]

1.3 Definições, Acrônimos e Abreviações

[Esta subseção fornece as definições de todos os termos, acrônimos e abreviações necessárias para interpretar corretamente a Especificação de Componentes. Em caso de existir um glossário de projeto, indicar nesta seção]

CBERS - China Brazil Earth Resource Satellite, Satélite Sino-Brasileiro de Recursos Terrestres

CCS - Centro de Controle de Satélites

CMV - Conjunto de Medida de Velocidade

DSS - Divisão de Desenvolvimento de Sistemas de Solo.

ESA - European Space Agency, Agência Espacial Européia

ET - Estação Terrena

INPE - Instituto Nacional de Pesquisas Espaciais

FPGA - Field Programmable Gate Array

MMF - Módulo de Medida de Frequência

MSC - Módulo de Software de Controle

PROCOD III - Processador de Coleta de Dados

PTT - Processador de Telemetria e Telecomando

PPS - Pulso por Segundo

RF - Rádio Frequência

SDID - Station Data Interchange Document, Documentação para troca de Dados

TT&C - Tracking, Telemetry, Telecommand and Control

ΔF - Desvio Doppler

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

1.4 Referências

[Esta subseção fornece uma lista completa de todos os documentos mencionados em outra parte da Especificação de Componentes. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento]

ECSS-E-ST-70C. ECSS Space Engineering – Ground systems and operations. July, 2008.

ECSS-E-ST-50-02C. ECSS Space Engineering - Ranging and Doppler tracking

ECSS-E-ST-10-06C ECSS Space Engineering – Technical requirements specification. March, 2009.

[WiWer] Wiley J. Larson and James R Wertz. Space Mission Analysis and Design. Space Technology Series. Microcosm, Inc. 3rd. Edition 1999.

Valcir Orlando e Hélio Koiti Kuga, Rastreo e Controle de satélites do INPE, Capítulo 6, São José dos Campos, 1999

A-EIF-0004 Ground Segment Communication Protocol Specification Instituto Nacional de Pesquisas Espaciais, primeira edição 15/02/1993

JANDL JUNIOR, Peter. Java: guia do programador. -- São Paulo: Novatec Editora, 2007. ISBN 978-85-7522-109-9

ENGHOLM Júnior, Hélio. Engenharia de Software na prática. --São Paulo: Novatec Editora, 2010 ISBN 978-85-7522-217-1

GONÇALVES, Edson. Desenvolvendo Aplicações Web com JSP Servlets, JavaServerFaces, Hibernate, EJB3 Persistence e Ajax. -- Rio de Janeiro: Editora Ciência Moderna Ltda., 2007. ISBN 978-85-7393-572-1

LUCKOW, Décio Heinzemann e MELO, Alexandre Altair. Programação Java para Web. 1. ed. São Paulo: Novatec Editora, 2010. ISBN 978-85-7522-238-6

SOMMERVILLE, Ian. ENGENHARIA DE SOFTWARE 8a Edição São Paulo: PEARSON EDUCATION DO BRASIL, 2007. ISBN 858863928-9

ANDRADE, Thiago. Algaworks dwjsf Desenvolvimento Web com JSF. Uberlândia. Disponível em <<http://www.algaworks.com/treinamentos/apostilas>>. Acesso em 04 mar. 2011.

Gilleanes T. A. Guedes. UML , uma abordagem Prática. – São Paulo: NOVATEC, 2009.

A. C. Pinheiro e A.S. Simão, JPlavisFSM Manual de Instruções,.

1.5 Visão Geral

[Esta subseção descreve o que o restante da Especificação de Componentes contém e explica como o documento está organizado]

2. Especificação Geral do Componente.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

2.1 Descrição.

2.1.1 Breve Descrição

[A descrição aborda brevemente o papel e a finalidade do componente]

[Descrever os fatores gerais que afetam o componente e seus requisitos como:

- *perspectiva do produto*
- *funções do produto*
- *restrições*
- *suposições e dependências*
- *conjuntos de requisitos]*

O Conjunto de Medida de Velocidade - CMV é um sistema que tem como objetivo estimar a velocidade radial de um satélite em órbita, pela medição do desvio Doppler da frequência de uma portadora de RF transmitida para o solo e disponibiliza a massa de dados para processamento da previsão de órbitas.

O CMV realiza a partir de um pedido, em modo local ou remoto, as medidas de desvio Doppler. Estes pedidos são interpretados por um módulo de software que gera uma requisição que é enviada ao CMV através de uma rede de comunicação. O CMV verifica e envia uma mensagem de resposta ao requisitante se os comandos são válidos ou não. Em caso positivo, o CMV realiza as medidas requisitadas, coleta as informações de status do equipamento e envia uma mensagem de resposta ao requisitante.

As medidas obtidas são formatadas em relatórios tipo "XML" e "Excel" conforme o padrão de comunicação do segmento solo e enviadas ao Centro de Controle de Satélites (CCS) para posterior processamento.

2.1.2 Representação Gráfica do Componente

[Utilizar diagramas, modelos entre outros recursos que representem claramente o componente, suas funções, os subcomponentes. A representação deve permitir a visão independente do hardware e do software do componente, bem como as funcionalidades que serão realizadas por eles]

O CMV é formado pelos seguintes módulos:

- Módulo de Medida de Frequência é o hardware responsável pela medida do desvio de frequência e Módulo de Datação, é o hardware que fornece a datação das mensagens.
- Módulo Software de Controle (MSC) é responsável pelo controle de processos, interface com usuários, configuração, monitoração do CMV e tratamento de medidas.

A figura 1 mostra o contexto de desenvolvimento do CMV.

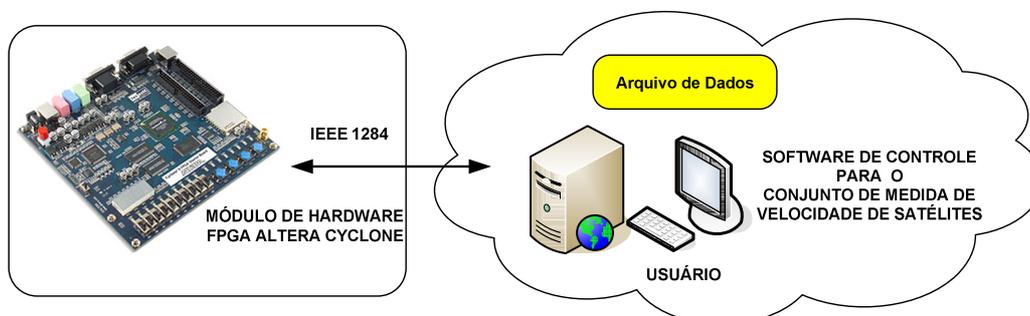


Figura 1- Contexto de Desenvolvimento (CASSIANO e KONO, 2012).

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

2.2 Metas

[Uma especificação das metas ou dos objetivos mensuráveis do componente]

O desenvolvimento do Software de Controle para o Conjunto de Medida de Velocidade de Satélites apresenta inovação tecnológica caracterizada pela implementação de hardware programável, FPGA, a elaboração da lógica de controle baseada em VHDL e a elaboração do software de aplicação com técnicas de orientação a objetos na plataforma Java com acesso via WEB.

A versão de protótipo do Conjunto de Medida de Velocidade de Satélite tem como foco principal o atendimento aos requisitos de acesso ao hardware e aquisição de dados em tempo real utilizado a plataforma Java. Esta versão é utilizada, no laboratório da Divisão de Desenvolvimento de Sistemas de Solo, como referência em soluções de hardware.

O projeto contempla a elaboração de nova versão com abordagem baseada na Linguagem de Modelagem Unificada (UML) utilizando um processo de desenvolvimento RUP customizado.

A metodologia de engenharia de software é descrita em ENGHOLM Júnior (2010). São abordados os processos de software, gerenciamento de requisitos e mudanças, o controle de qualidade, os processos de análise, arquitetura e design de software, os processos de testes e implantação com tecnologia orientada a objetos.

Para efeito da aplicação da abordagem UML e o RUP customizado, serão considerados alguns casos de uso descritos no item diagrama de caso de uso, conforme lista a seguir.

- Diagrama de caso de uso - Contexto
- Diagrama de caso de uso - Configuração de Requisição

2.3 Métricas

[Uma especificação das métricas relevantes ao componente e uma definição das metas que você deseja atingir usando essas métricas]

2.3.1 <nome da medida de desempenho1>

[Uma breve descrição da medida de desempenho]

2.3.2 <nome da medida de desempenho2>

[Uma breve descrição da medida de desempenho]

2.4 Fluxo de Trabalho

[Uma descrição textual do fluxo de trabalho representado pelo componente]

2.4.1 Fluxo de Trabalho Básico

<nome do passo do fluxo de trabalho>

[Uma breve descrição do passo do fluxo de trabalho]

2.4.2 Fluxos de Trabalho Alternativos

<nome do passo do fluxo de trabalho>

[Uma breve descrição do passo do fluxo de trabalho]

2.5 Categoria

[Indica o nível de responsabilidade das funções realizadas pelo componente em seu ambiente, se o 'central', 'de suporte', 'de gerenciamento' ou 'outras']

2.6 Risco

[Especifica os riscos de funcionamento e implementar do Componente]

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

2.7 Possibilidades

[Descreve o potencial de aprimoramento estimado ao componente]

2.8 Proprietário do Processo

[Define quem é o proprietário dos resultados alcançados]

2.9 Requisitos Especiais

[Descrição dos requisitos especiais do componente]

2.9.1 <Nome do Requisito Especial 1>

[Uma breve descrição do requisito especial]

2.9.2 <Nome do Requisito Especial 2>

[Uma breve descrição do requisito especial]

2.10 Pontos de Extensão

[Pontos de extensão ao requisito]

2.10.1 <Nome do Ponto de Extensão1>

[Uma definição do local do ponto de extensão no fluxo]

2.10.2 <Nome do Ponto de Extensão2>

[Uma definição do local do ponto de extensão no fluxo]

2.11 Comportamento

[Esta seção expressa os objetivos gerais do comportamento do produto que não são específicos de um determinado componente]

2.12 Usabilidade

[Esta seção inclui todos os requisitos que afetam a usabilidade]

2.13 Confiabilidade

[Os requisitos de confiabilidade (do ponto de vista do produto) devem ser especificados aqui. Abaixo, algumas sugestões:

- *Disponibilidade — especifique o percentual de tempo disponível (xx.xx%), horas antecipadas de uso, etc.*
- *Exatidão — especifique a precisão (resolução) e a exatidão (por algum padrão conhecido) que são necessárias na saída]*

2.14 Desempenho.

[As características de desempenho devem ser resumidas nesta seção. Inclua tempos de resposta específicos. Quando aplicável, faça referência a nomes dos componentes relacionados.

- *Tempo de resposta de uma transação (médio, máximo)*
- *Taxa de transferência;*
- *Capacidade;*
- *Uso de recursos]*

2.15 Problemas de Escala

[Liste e descreva em poucas palavras qualquer informação sobre as limitações ou precauções que precisam ser consideradas]

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

3. Distribuição das Funcionalidades (Modelagem).

[E a atividade de construção de modelos que explique as características e o comportamento da parte software e da parte hardware do componente a ser desenvolvido. Em sua construção os modelos devem ser usados na elaboração e representação da arquitetura das características e funcionalidades que o componente deve prover e contribuir para o seu planejamento e na sua construção. Em geral implica na construção de modelos gráficos que simbolizam os artefatos dos componentes de software e hardware utilizados e os seus inter-relacionamentos. A escolha da técnica e da ferramenta a ser utilizada, deve ser escolhida de acordo com as características do componente a ser desenvolvido com as características do ambiente de desenvolvimento e dos envolvidos. Pode-se utilizar representação através de fluxogramas, linguagens gráficas, ferramentas e técnicas direcionadas para o componente a ser produzido.]

3.1 Hardware.

3.1.1 Descrição Completa.

Módulo de Medida de Frequência é o hardware responsável pela medida do desvio de frequência e Módulo de Datação, é o hardware que fornece a datação das mensagens.

3.1.2 Representação Gráfica do Hardware.

[Representar preferencialmente através de modelos e diagramas o hardware e toda a sua funcionalidade, incluindo interfaces com o software e outros componentes]

3.1.3 Modelo do Hardware.

[Protótipo/ modelo de hardware com todos os artefatos utilizados]

3.1.4 Descrição das ferramentas utilizadas.

[Descrição das ferramentas a serem utilizadas para as atividades de modelagem e prototipação do componente.]

3.1.5 Descrição detalhada do Hardware.

O MMF efetua a medida da variação de frequência devido ao efeito Doppler. A medida de frequência é feita em uma janela de tempo definida pelo operador, síncrona com a referência interna de 1 PPS (um pulso por segundo).

A medida de frequência é obtida através da leitura de 02 grupos de contadores:

- O Contador Totalizador: formado basicamente por um contador binário de 32 bits que tem a função de determinar o número de ciclos inteiros da frequência de entrada (FIN).
- O Contador de Intervalo: formado por um contador binário de 8 bits que tem a função de determinar o número de ciclos do sinal de 180 MHz, (FREF) dentro de uma janela de ajuste.

O diagrama de tempo simplificado, figura 2, mostra o processo básico de aquisição de dados e o cálculo da frequência e o diagrama de tempo detalhado do MMF, figura 3, mostra os instantes para aquisição dos dados.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

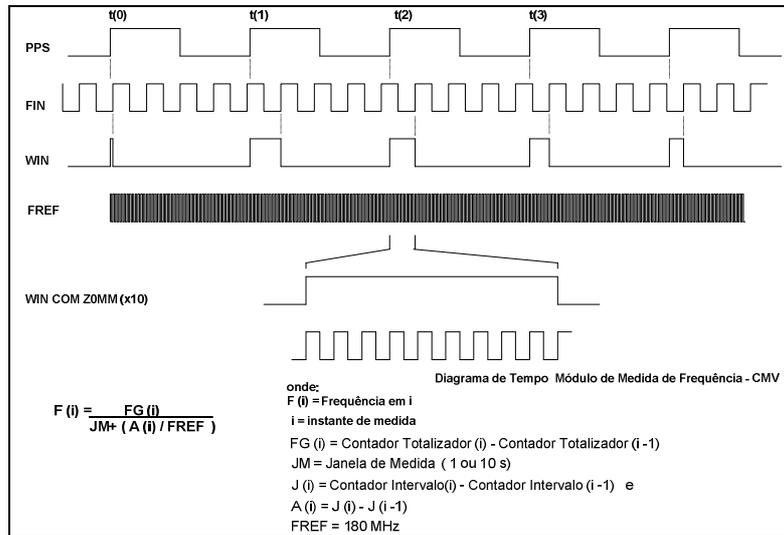


Figura 2 - Diagrama de tempo Simplificado

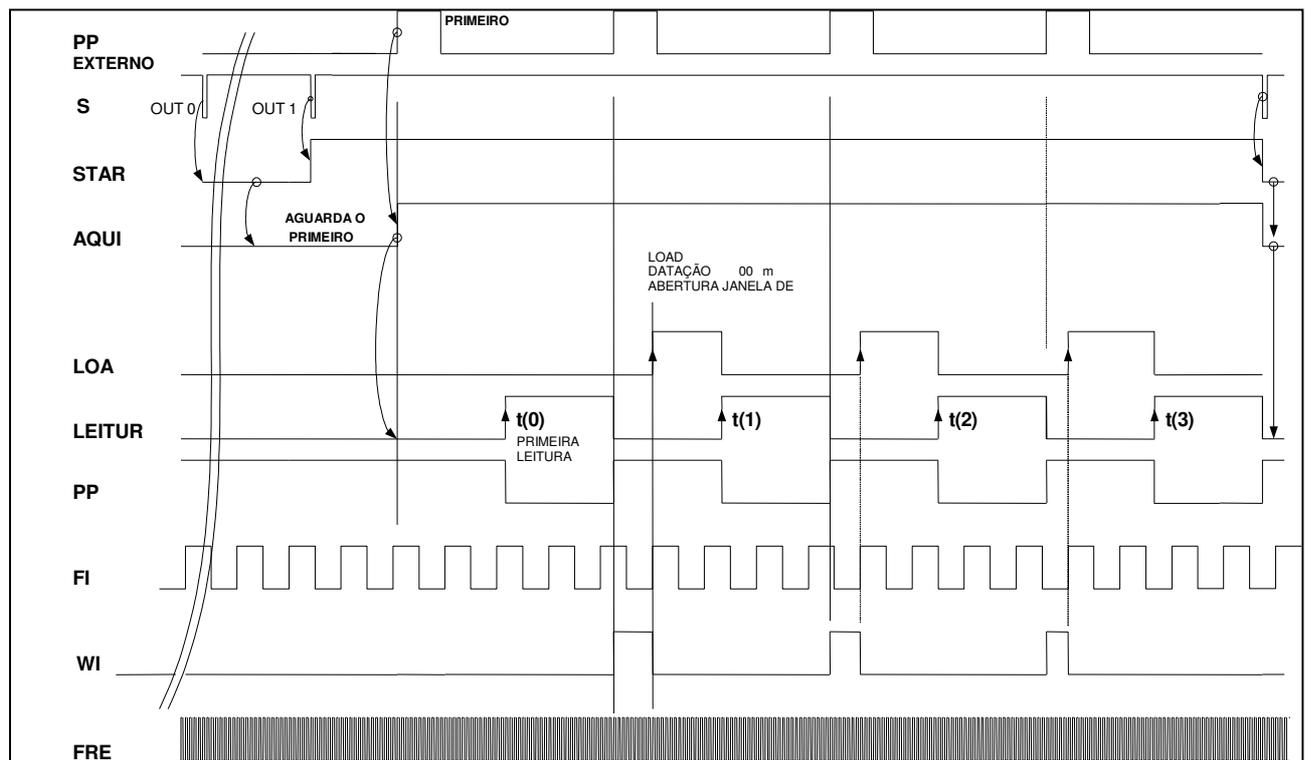


Figura 3 - Diagrama de Tempo Detalhado do MMF

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

REQUISITOS DE HARDWARE

Estes requisitos são atendidos pelo módulo de hardware e não fazem parte do escopo de desenvolvimento desta aplicação de software.

REQHW_01

O hardware deve medir a frequência de 1 MHz $\pm \Delta F$ (Desvio Doppler), com ΔF de 150 kHz.

REQHW_02

O hardware deve permitir a seleção entre 2 janelas de medidas de 1 ou 10 segundos.

REQHW_03

O hardware deve apresentar precisão da medida melhor que 180 mHz

REQHW_04

O hardware deve ser compatível com a interface para microcomputador EPP, IEEE 1284.

3.1.6 <Nome do artefato1>

[Descrição do artefato]

3.1.6.1 Descrição detalhada da Artefato1

[Descrição de todos os itens (artefatos) utilizados para a produção do hardware, bem como a função de cada item no componente, incluir quantas linhas forem necessárias]

3.1.7 Descrição das interfaces do Hardware

[Descrição detalhada das interfaces que o hardware faz].

3.2 Software.

3.2.1 Descrição Completa

Módulo Software de Controle (MSC) é responsável pelo controle de processos, interface com usuários, configuração, monitoração do CMV e tratamento de medidas.

A figura 4 apresenta o sistema Módulo de Software de Controle (MSC) baseado na arquitetura Web-Centric N Camadas, de acordo com ENGHOLM Júnior (2010), onde a camada cliente é realizada por browsers da web e o servidor web é o provedor de lógicas de negócio e de apresentação.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

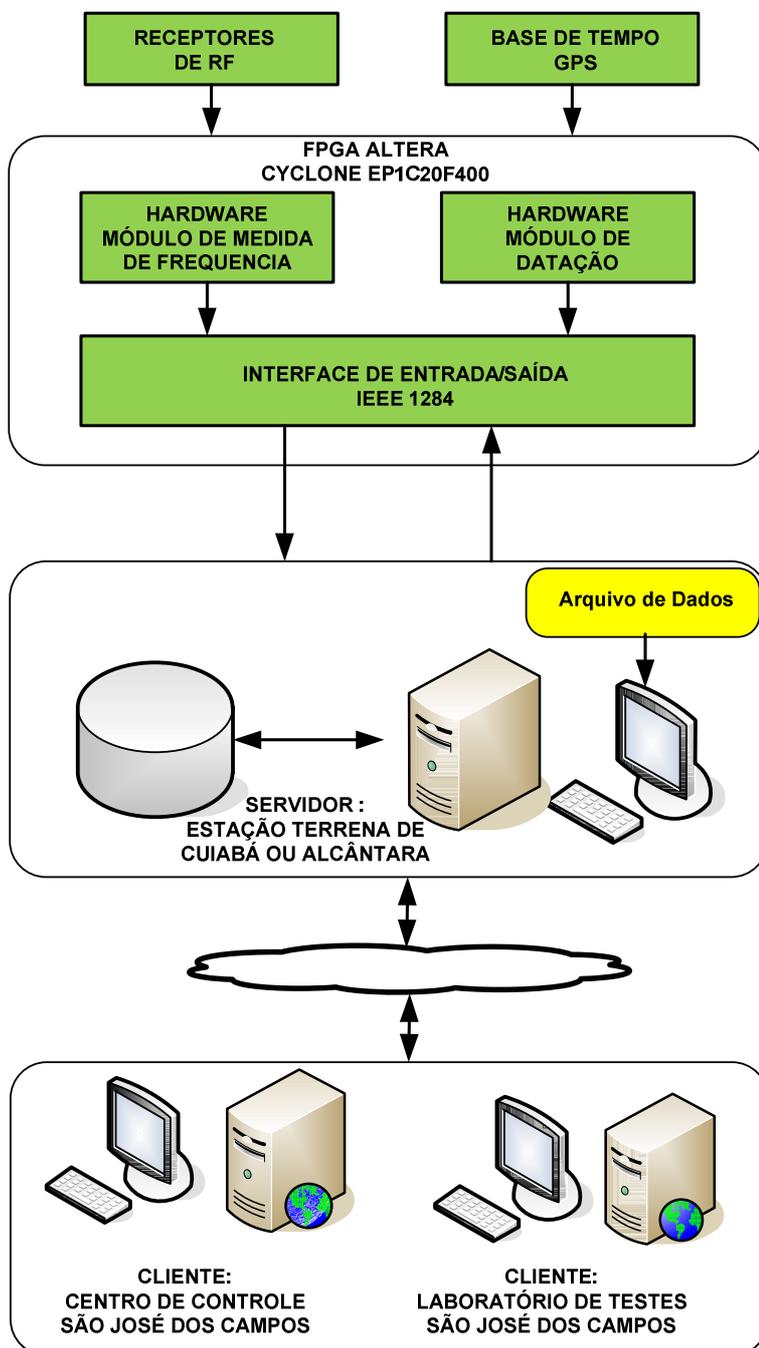


Figura 4 - Visão do sistema Software de Controle do CMV (CASSIANO e KONO, 2012).

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

3.2.2 Representação Gráfica do Software

[Representar preferencialmente através de modelos e diagramas o software e toda a sua funcionalidade, incluindo interfaces com o hardware e outros componentes]

Apresentamos os principais diagramas de caso de uso da aplicação da abordagem UML e o RUP customizado, serão considerados os casos de uso descritos conforme lista a seguir.

- o Diagrama de caso de uso - Contexto
- o Diagrama de caso de uso - Configuração de Requisição.

A figura 5 mostra os casos de uso para contexto e a figura 6 mostra o caso de uso para configuração da requisição.

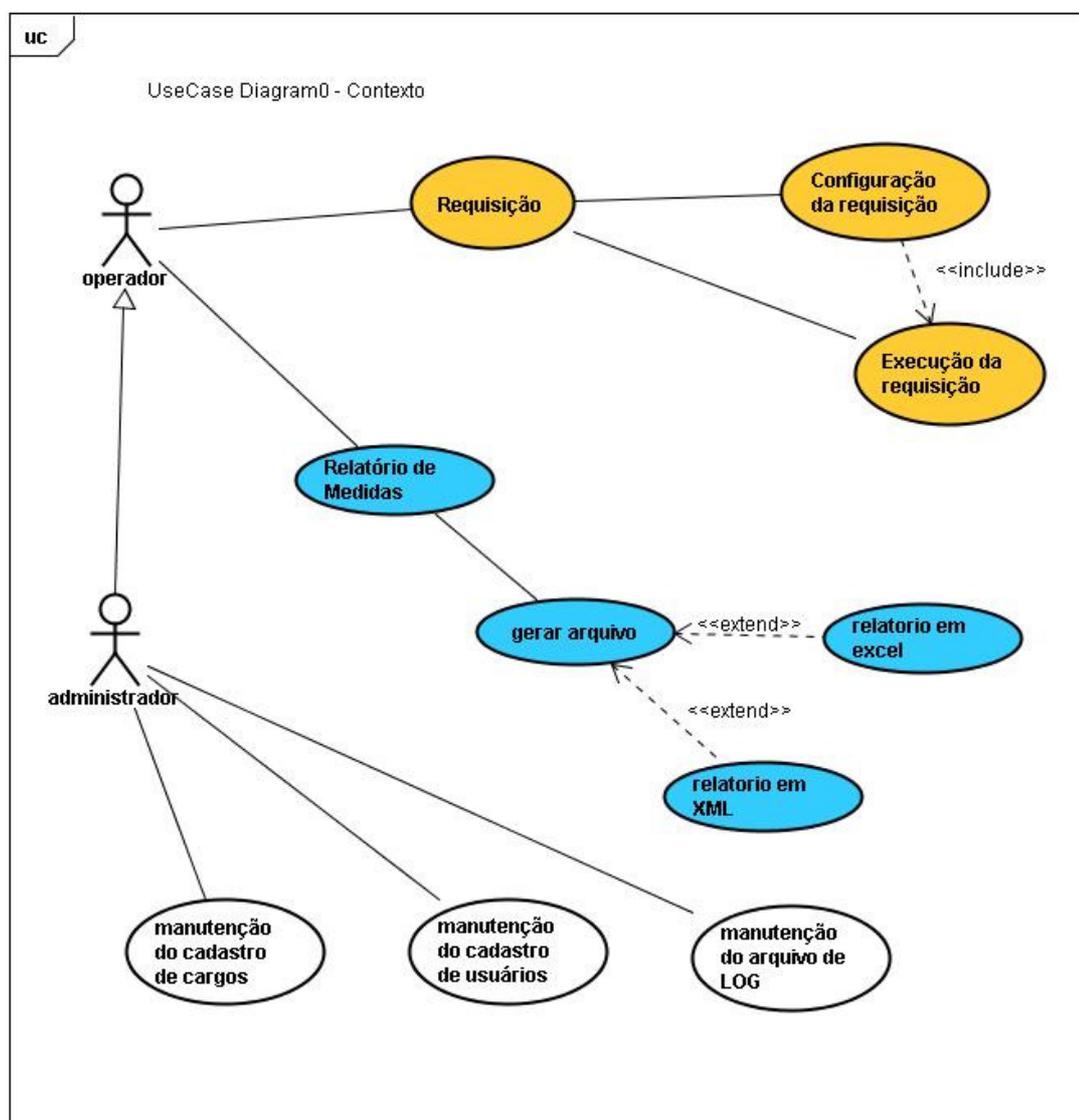


Figura 5 - Caso de uso – Contexto (CASSIANO e KONO, 2012)

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

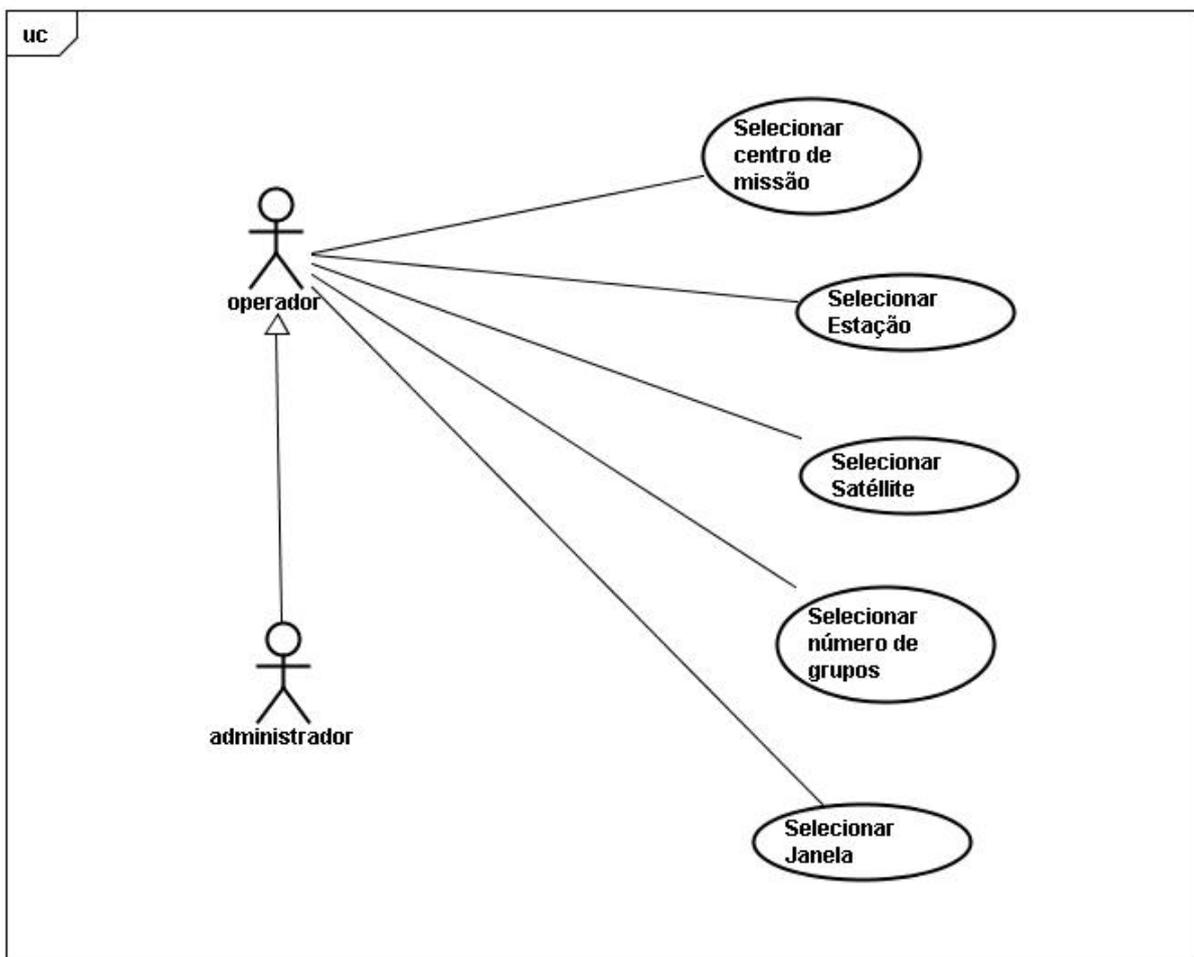


Figura 6 - Caso de uso - Configuração de Requisição (CASSIANO e KONO, 2012)

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

DESCRIÇÃO DOS CASOS DE USO

São relacionados os casos de uso e sua descrição.

Caso de Uso	Descrição
Requisição	Realizar a requisição apresentando uma tela que permite configurar a requisição. Apresenta opção para executar a requisição.
Configuração da requisição	Possibilita informar os parâmetros da requisição: centro de missão, estação receptora, satélite, número de grupos e janela de aquisição (tempo). Apresenta opção para executar a requisição.
Execução da requisição	Executar a requisição de medidas. Apresenta como resultado de saída, os dados das medidas processadas em 3 interfaces:requisições, resposta e grupos de medidas.
Relatório de medidas	Apresenta os dados gerados do histórico de requisições e dos grupos de medidas. Possibilita exportar/gravar o relatório em Excel e/ou em XML
Gerar arquivo	Exporta /grava o relatório em Excel e/ou em XML
Relatório em Excel	Exporta /grava o relatório em arquivo Excel.
Relatório em XML	Exporta /grava o relatório em arquivo com padrão XML.
Selecionar centro de missão	Permite selecionar o centro de missão para a execução das medidas.
Selecionar estação	Permite selecionar a estação de onde serão realizados as transmissões e recepção das medidas.
Selecionar satélite	Possibilita escolher o satélite alvo das medidas.
Selecionar número de grupos	Possibilita selecionar o número de grupos de medidas a serem realizadas.
Selecionar janela	Permite escolher a janela de tempo entre as medidas.
Manutenção do cadastro de cargos	Possibilita realizar a manutenção do cadastro de cargos utilizado no cadastro de usuários. Não será abordado.
Manutenção do cadastro de usuários	Permite realizar a manutenção do cadastro de usuários que podem acessar o sistema. Não será abordado.
Manutenção do arquivo de LOG	Possibilita realizar a manutenção do arquivo de LOG ou histórico do sistema. Não será abordado.

Tabela 1 - Descrição dos Casos de Uso

DIAGRAMA DE ATIVIDADES

Apresentamos o principal diagrama de atividades, Requisição de Medida, para efeito da aplicação da abordagem UML e o RUP customizado.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

Este diagrama, figura 7, mostra as atividades para uma requisição de medida de acordo com o caso de uso requisição de medida.

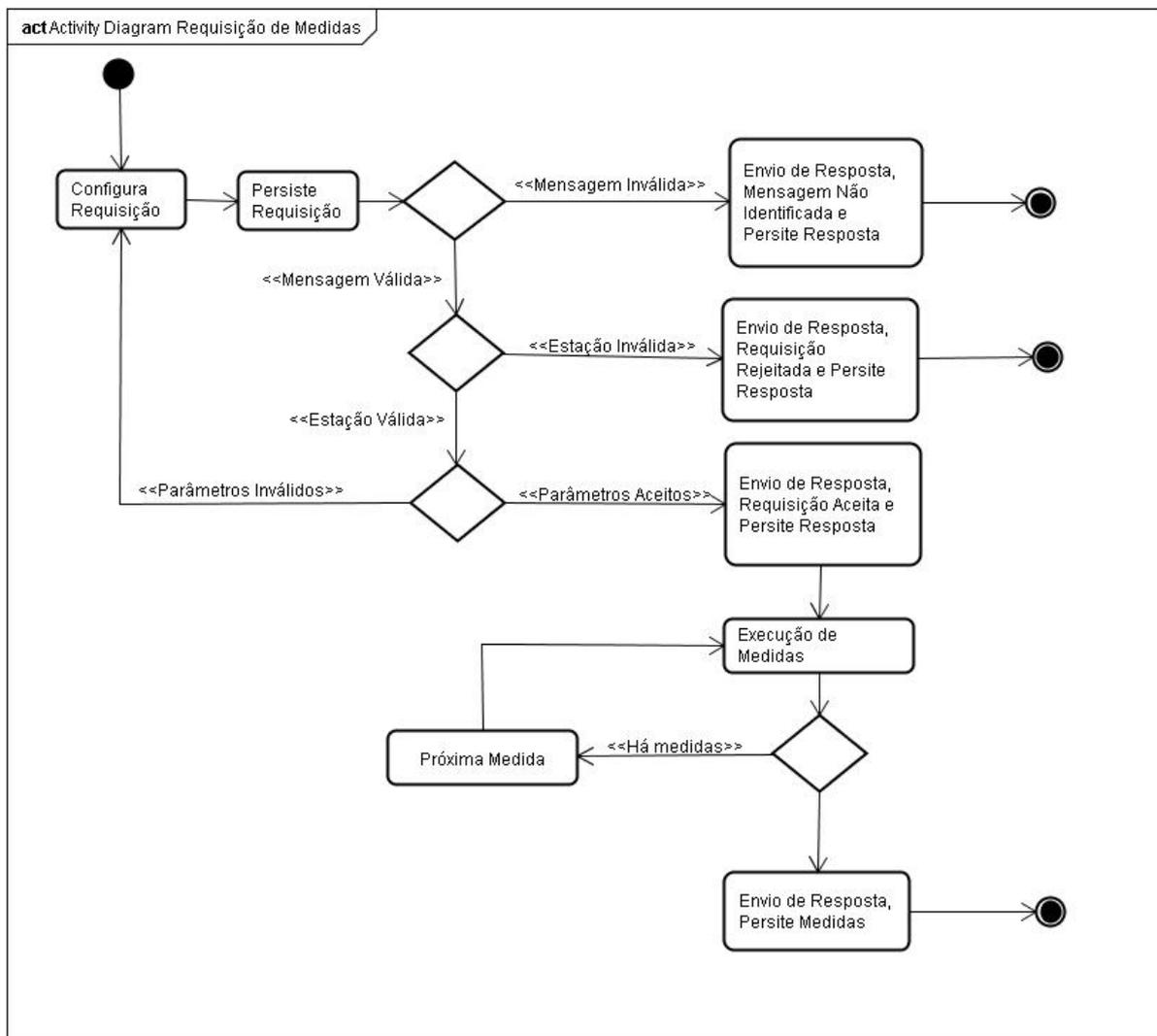


Figura 7 - Diagrama de Atividade - Requisição de Medida (CASSIANO e KONO, 2012)

3.2.3 Modelo do Software

[Protótipo/ modelo de software com todos os artefatos utilizados]

3.2.4 Descrição das ferramentas utilizadas.

[Descrição das ferramentas a serem utilizadas para as atividades de modelagem e prototipação do componente.]

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

3.2.5 Descrição detalhada do Software

O MSC gerencia todos os processos necessários para a execução das funções do CMV, estes processos são: interface web, configuração e monitoração de parâmetros, o tratamento das medidas e acesso ao sistema através de uma conexão Internet.

O contexto de nossa aplicação tem foco nos requisitos de software, para atender esta delimitação os dados de entrada serão provenientes de um arquivo no formato compatível e disponibilizado pelo hardware de aquisição.

REQUISITOS FUNCIONAIS

REQUISITOS PARA ACESSO AO SISTEMA

REQFUN_01

O sistema deve permitir o acesso ao usuário, com validação de login e senha.

REQFUN_02

O sistema deve apresentar um menu com as seguintes opções:

Administrador > Usuário, Cargos e Log

Operação > Requisição.

Relatórios > Relatório de Medidas

Sistema > Logoff

Ajuda > Projeto

REQFUN_03

O sistema deve permitir o registro de acesso de usuários ao sistema.

REQUISITOS PARA CONFIGURAÇÃO E OPERAÇÃO

REQFUN_04

O sistema deve apresentar uma página para a configuração dos parâmetros de uma requisição de medida, estes parâmetros são:

- Centro de Missão: Natal ou Teste
- Estação Receptora: Cuiabá ou Alcântara
- Satélite: SCD1 ou CBERS
- Número de Grupos de medidas
- Janela de Medida

REQFUN_05

O sistema deve permitir a seleção da janela de medida entre os valores de 1 ou 10 s.

REQFUN_06

O sistema deve formar um grupo de medida a cada 10 medidas.

REQFUN_07

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

O sistema deve permitir a programação do número de grupos de medidas, sendo o valor mínimo de 1 e o máximo de 255 para janela de 1s.

REQFUN_08

O sistema deve permitir a programação do número de grupos de medidas, sendo o valor mínimo de 1 e o máximo de 25 para janela de 10s.

REQFUN_09

O sistema deve tratar as medidas de frequência de acordo com tamanho da janela de medida e o número de grupo de medidas.

REQFUN_10

O sistema deve efetuar a aquisição da datação a cada janela de medida.

REQFUN_11

O sistema deve indicar, ao usuário, o status de medida em andamento.

REQFUN_12

O sistema deve enviar uma mensagem de resposta à requisição de medida, as respostas são:

- Resposta Primária: mensagem indicando que a transação foi aceita.
- Resposta para a transação: mensagem com as medidas requisitadas.
- Resposta de mensagem não identificada: indicando que a transação não foi identificada.

REQFUN_13

O sistema deve registrar um histórico das medidas realizadas.

REQFUN_14

O sistema deve apresentar uma página com as seguintes tabelas de resultados:

- Requisições Solicitadas, de acordo com o REQFUN_15.
- Respostas às requisições, de acordo com o REQFUN_16.
- Grupo de Medidas de acordo com o REQFUN_17.

REQFUN_15

O sistema deve permitir, após o término da medidas requisitadas, a visualização das 10 últimas Requisições Solicitada, em ordem decrescente, listando os seguintes parâmetros: Número da Requisição, Tipo de Comando, Satélite, Estação, Missão, Usuário, Número de Grupos e Datação.

REQFUN_16

INPE - Instituto Nacional de Pesquisas Espaciais	Version: <1.0>
Software de Controle - Conjunto de Medida de Velocidade	Date: <06/08/14>
Especificação do Componente	

O sistema deve permitir, a visualização das Respostas, em ordem decrescente, às requisições de medidas, listando os seguintes parâmetros: Número da resposta, Número da requisição, Resultado, Status do CMV, Status do HW, Mensagem e Amostras.

REQFUN_17

O sistema deve permitir a visualização um relatório dos 10 últimos Grupos de Medidas listando os seguintes parâmetros: Número da medida, Número da resposta, Datação UTC, Número da Mensagem, Contador Totalizador, Contador de Intervalo e Frequência, com unidade em Hertz.

3.2.6 <Nome da funcionalidade1>

[Descrição da funcionalidade]

3.2.6.1 Descrição detalhada da Funcionalidade1

[Descrição de como deve ser implementada a funcionalidade]

3.2.7 Descrição das interfaces do Software

[Descrição detalhada das interfaces que o software faz]

REQFUN_18

O sistema deve apresentar uma página com as seguintes tabelas:

Histórico de Requisições, de acordo com o REQFUN_19.

Grupo de Medidas, de acordo com o REQFUN_20.

REQFUN_19

O sistema deve permitir a visualização do Histórico de Requisições com todas as Requisições Solicitadas, em ordem decrescente, listando os seguintes parâmetros: Número da Resposta, Número da Requisição, Resultado, Tipo de Comando, Satélite, Estação, Usuário, Número de Grupos e Datação.

REQFUN_20

O sistema deve permitir a visualização dos 10 últimos Grupos de Medidas listando os seguintes parâmetros: Número da medida, Número da resposta, Datação UTC, Número da Mensagem, Contador Totalizador, Contador de Intervalo e Frequência, com unidade em Hertz.

REQFUN_21

O sistema deve permitir a geração de um arquivo com os 10 últimos Grupos de Medidas nos formato XML ou Excel, listando os seguintes parâmetros: Número da medida, Número da Resposta, Datação UTC, Número da Mensagem, Contador Totalizador, Contador de Intervalo e Frequência, em Hertz.

INPE - Instituto Nacional de Pesquisas Espaciais

Divisão de Desenvolvimento de Sistemas de Solo
Software de Controle - Conjunto de Medida de
Velocidade
Plano de Teste
Versão <1.0>

Magda Miyashiro

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mm/aa>	<x.x>	<detalhes>	<nome>

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Índice Analítico

1.	Introdução	5
1.1	Finalidade	5
1.2	Escopo	5
1.3	Alvo do Teste	5
1.4	Terminologia e Acrônimos do Documento	5
1.5	Referências	5
1.6	Estrutura do Documento	6
2.	Missão de Avaliação e Motivação dos Testes	6
2.1	Informações Detalhadas	6
2.2	Missão de Avaliação	6
2.3	Motivadores dos Testes	6
3.	Itens-alvo dos Testes	6
4.	Resumo dos Testes Planejados	7
4.1	Resumo das Inclusões dos Testes	7
4.2	Resumo das Inclusões dos Testes	7
5.	Abordagem dos Testes	8
5.1	Técnicas e Tipos de Teste	8
5.1.1	Teste de Integridade de Dados e de Banco de Dados	8
5.1.2	Teste de Função	8
5.1.3	Teste da Interface do Usuário	10
5.1.4	Teste de Carga	11
5.1.5	Teste de Stress	11
5.1.6	Teste de Volume	11
6.	Critérios de Entrada e de Saída	12
6.1	Plano de Teste	12
6.1.1	MEF COMPLETA PARA SUT	13
6.1.1.1	ANÁLISE DAS PROPRIEDADES	13
6.1.1.2	TABELA DE ESTADOS E EVENTOS PARA MEF COMPLETA	14
6.1.1.3	GERAÇÃO DE CASOS DE TESTES PARA MEF COMPLETA	14
6.1.1.3.1	CASOS DE TESTE – MÉTODO W	14
6.1.1.3.2	CASOS DE TESTE – MÉTODO HSI	14
6.1.1.3.3	CASOS DE TESTE – MÉTODO SPY	15
6.1.1.3.4	7.3.4 CASOS DE TESTE – MÉTODO UIO	15
6.1.1.3.5	7.3.5 CASOS DE TESTE – MEF COMPLETA MÉTODO MANUAL E MUTANTES	15
6.1.1.3.6	7.3.6 CASOS DE TESTE – MÉTODO MANUAL E IMPLEMENTAÇÃO	16
6.1.1.4	COMPARAÇÃO ENTRE MÉTODOS MEF COMPLETA	16
6.1.1.5	COMPARAÇÃO ENTRE MÉTODOS W, HSI, SPY, UIO E MANUAL, MEF COMPLETA	16
6.1.2	CONCLUSÃO MEF COMPLETA	17
6.1.3	CONCLUSÃO GERAL DE CASOS DE TESTES	17
6.1.4	CONCLUSÃO GERAL	18
6.1.5	Critérios de Entrada de Plano de Teste	18
6.1.6	Critérios de Saída de Plano de Teste	18
6.1.7	Critérios de Suspensão e de Reinício	18
7.	Produtos Liberados	18

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

7.1	Sumários de Avaliação de Testes	18
7.2	Relatórios da Cobertura de Teste	18
7.3	Relatórios da Qualidade Perceptível	19
7.4	Registros de Incidentes e Solicitações de Mudança	19
7.5	Conjunto de Testes de Regressão e Scripts de Teste de Suporte	19
7.6	Produtos de Trabalho Adicionais	19
7.6.1	Resultados Detalhados dos Testes	19
7.6.2	Scripts de Teste Funcionais Automatizados Adicionais	19
7.6.3	Guia de Teste	19
8.	Fluxo de Trabalho de Teste	19
9.	Necessidades Ambientais	19
9.1	Hardware Básico do Componente	19
10.	Responsabilidades, Perfil da Equipe e Necessidades de Treinamento	20
10.1	Pessoas e Papéis	20
10.2	Perfil da Equipe e Necessidades de Treinamento	20
11.	Marcos do teste	20
12.	Procedimentos e Processos de Gerenciamento	21
12.1	Medição e Avaliação da Extensão do Teste	21
12.2	Avaliação dos Produtos Liberados deste Plano de Teste	21
12.3	Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções	21
12.4	Estratégias de Rastreabilidade	21
12.5	Aprovação e Encerramento	21

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Plano de Teste

1. Introdução

1.1 Finalidade

[A finalidade do Plano de Teste é reunir todas as informações necessárias ao planejamento e ao controle do esforço de teste. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.

Este Plano de Teste referente ao Divisão de Desenvolvimento de Sistemas de Solo suporta os seguintes objetivos:

- *[Identifica os itens que devem ser inspecionados pelos testes.*
- *Identifica a motivação e as ideias subjacentes às áreas de teste a serem abrangidas.*
- *Descreve a abordagem de teste que será usada.*
- *Identifica os recursos necessários e fornece uma estimativa dos esforços de teste.*
- *Lista os elementos liberados do projeto de teste]*

1.2 Escopo

O processo de validação envolve o Módulo Software de Controle (MSC) onde o mesmo será validado de forma unitária. As validações são direcionadas aos requisitos requeridos que são divididos em três grupos:

Requisitos de Acesso ao Sistema;

Requisitos para a Configuração do Sistema e Operação e ;

Requisitos para a Geração de Relatórios.

As validações integradas não fazem parte do escopo.

1.3 Alvo do Teste

O objeto do teste é o System Under Test (SUT) que é um Módulo Software de Controle (MSC) que gerencia os processos necessários para a execução das funções do CMV

1.4 Terminologia e Acrônimos do Documento

*[Esta subseção fornece as definições de todos os termos, acrônimos e abreviações necessárias à adequada interpretação do **Plano de Teste**. Caso o projeto tenha um glossário inclua uma referência ao Glossário do projeto na seção Referências]*

1.5 Referências

ECSS-E-ST-10-02C. Space Engineering – Verification. March, 2009.

Wieringa, R. J Design Methods for Reactive Systems: Yourdon, Statemate, and the UML..

Binder, R. V, Testing Object-oriented Systems: Models, Patterns and Tools. Addison-Wesley, 1999,.

A. C. Pinheiro e A.S. Simão, JPlavisFSM Manual de Instruções,.

A.S. Simão, Contribuição para Teste de Software, Janeiro de 2011, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, São Paulo, Brazil.

Chow, T. S. Testing software design modeled by finite-state machines. IEEE Transactions on Software Engineering, v. 4, p. 178 -187, 1978.

Delamaro, M. E.; Maldonado, J. C.; Jino, M. Conceitos básicos. In: Delamaro, M. E.; Maldonado, J. C.; Jino, M., eds. Introdução ao Teste de Software, Elsevier, p. 1 - 7, 2007b.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Petrenko, A.; Yevtushenko, N.; Lebedev, A.; Das, A. Nondeterministic state machines in protocol conformance testing. In: Proceedings of the 6th International Workshop on Protocol Test systems VI (IFIP TC6/WG6.1), Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1993, p. 363-378.

1.6 Estrutura do Documento

*[Esta subseção descreve o que o restante do **Plano de Teste** contém e fornece uma introdução de como o restante do documento está organizado]*

2. Missão de Avaliação e Motivação dos Testes

[Forneça uma visão geral da missão e da motivação dos testes que serão conduzidos]

2.1 Informações Detalhadas

*[Forneça uma breve descrição dos fundamentos que justificam o esforço de teste definido neste **Plano de Teste**. Inclua informações como, por exemplo, o problema principal que está sendo resolvido, os principais benefícios da solução, a arquitetura planejada da solução e um breve histórico do projeto. Quando essas informações estiverem definidas em outros documentos, você poderá incluir referências a esses documentos mais detalhados caso seja apropriado. Esta seção só deverá conter de três a cinco parágrafos]*

2.2 Missão de Avaliação

[Forneça uma breve sentença que defina a missão do esforço de avaliação atual. Essa sentença poderá incorporar uma ou mais preocupações incluindo:

localizar o maior número de erros possível

localizar problemas importantes e avaliar os riscos da qualidade perceptível

advertir sobre os riscos de projeto percebidos

certificar um padrão

verificar uma especificação (requisitos, design ou alegações)

advertir sobre a qualidade do produto e satisfazer os envolvidos

advertir sobre os testes

cumprir as determinações do processo

e assim por diante

Cada missão fornece um contexto diferente para o esforço de teste e altera a maneira como o teste deverá ser abordado]

2.3 Motivadores dos Testes

[Forneça um resumo dos principais elementos que motivarão o esforço de teste. Os testes poderão ser motivados por uma série de fatores como, por exemplo, riscos de qualidade, riscos técnicos, riscos do projeto, casos de uso, requisitos funcionais, requisitos não funcionais, elementos de design, falhas ou erros suspeitos, solicitações de mudança, etc.]

3. Itens-alvo dos Testes

A listagem abaixo identifica os itens — software, hardware e elementos de suporte do produto — que foram identificados como alvo dos testes. Essa lista representa os itens que serão testados.

[Forneça uma lista de nível superior dos principais itens que estarão sujeitos a testes. Essa lista deve incluir itens produzidos diretamente pela equipe de desenvolvimento do projeto e itens de que dependem esses

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

produtos; por exemplo, o hardware de processamento básico, dispositivos periféricos, sistemas operacionais, produtos ou componentes de terceiros, etc. É recomendável agrupar a lista por categoria e atribuir importância relativa a cada motivador]

4. Resumo dos Testes Planejados

[Esta seção fornece um resumo de nível superior dos testes que serão executados. O resumo fornecido aqui representa uma visão geral de nível superior dos testes que serão e dos que não serão executados]

4.1 Resumo das Inclusões dos Testes

*[Forneça um resumo de nível superior dos principais testes planejados atual. Observe o que será incluído no plano e registre o que **não** será incluído explicitamente na seção denominada Resumo das Inclusões dos Testes]*

4.2 Resumo das Exclusões dos Testes

*[Forneça um resumo de nível superior dos possíveis testes que poderiam ter sido conduzidos, mas que foram **explicitamente excluídos** deste plano. Se você não for implementar ou executar um tipo de teste, informe claramente que o teste não será executado ou implementado e justifique. A seguir, há exemplos de justificativas que poderão ser usadas:*

“Esses testes não contribuem para alcançar a missão de avaliação”

“Não há recursos suficientes para executar esses testes”

“Esses testes são desnecessários devido aos testes executados por xxx”

Segundo um prisma heurístico, se você achar que é perfeitamente concebível que um dos membros de seu público espere que um determinado aspecto de teste seja incluído e se você não pretender ou não puder incluí-lo, justifique sua exclusão. Se a equipe concordar que a exclusão é óbvia, você provavelmente não precisará listá-la]

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

5. Abordagem dos Testes

[Esta seção apresenta a estratégia recomendada para criar e implementar os testes necessários]

5.1 Técnicas e Tipos de Teste

5.1.1 Teste de Integridade de Dados e de Banco de Dados

Não se aplica.

5.1.2 Teste de Função

O Teste de Função se baseia nos requerimentos dos três grupos abaixo relacionados:

A) Requisitos de Acesso ao Sistema:

REQFUN_01

O sistema deve apresentar ao usuário um menu de operação com as seguintes opções:

- Requisição
- Relatório de Medidas
- Logoff

B) Requisitos para a Configuração do Sistema e Operação:

REQFUN_02

O sistema deve apresentar uma página para a configuração dos parâmetros de uma requisição de medida, estes parâmetros são:

- Centro de Missão: Natal ou Teste
- Estação Receptora: Cuiabá ou Alcântara
- Satélite: SCD1 ou CBERS
- Número de Grupos de medidas
- Janela de Medida

REQFUN_03

O sistema deve permitir a seleção da janela de medida entre os valores de 1 ou 10 s.

REQFUN_04

O sistema deve formar um grupo de medida a cada 10 medidas.

REQFUN_05

O sistema deve permitir a programação do número de grupos de medidas para janela de 1s, podendo ser o valor mínimo de 1 e o máximo de 255.

REQFUN_06

O sistema deve permitir a programação do número de grupos de medidas para janela de 10s, podendo ser o valor mínimo de 1 e o máximo de 25.

REQFUN_07

O sistema deve tratar as medidas de frequência de acordo com tamanho da janela de medida e o número de grupo de medidas.

REQFUN_08

O sistema deve efetuar a aquisição da datação a cada janela de medida.

REQFUN_09

O sistema deve enviar uma mensagem de resposta à requisição de medida, as respostas são:

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

- Resposta Primária: mensagem indicando que a transação foi aceita.
- Resposta para a transação: mensagem com as medidas requisitadas.
- Resposta de mensagem não identificada: indicando que a transação não foi identificada.

REQFUN_10

O sistema deve registrar um histórico das medidas realizadas.

REQFUN_11

O sistema deve apresentar uma página com as seguintes tabelas de resultados:

- Tabela 1, Requisições Solicitadas,
- Tabela 2, Respostas às requisições,
- Tabela 3, Grupo de Medidas.

REQFUN_12

O sistema deve permitir, após o término das medidas requisitadas, a visualização das 10 últimas Requisições Solicitadas, em ordem decrescente, listando os seguintes parâmetros: Número da Requisição, Tipo de Comando, Satélite, Estação, Missão, Usuário, Número de Grupos e Datação.

REQFUN_13

O sistema deve permitir a visualização das Respostas, em ordem decrescente, às requisições de medidas, listando os seguintes parâmetros: Número da resposta, Número da requisição, Resultado, Status do CMV, Status do HW, Mensagem e Amostras.

REQFUN_14

O sistema deve permitir a visualização um relatório dos 10 últimos Grupos de Medidas listando os seguintes parâmetros: Número da medida, Número da resposta, Datação UTC, Número da Mensagem, Contador Totalizador, Contador de Intervalo e Frequência, com unidade em Hertz.

C) Requisitos para a Geração de Relatórios:

REQFUN_15

O sistema deve apresentar uma página com as seguintes tabelas:

Tabela 4, Histórico de Requisições.

Tabela 5, Grupo de Medidas. O sistema deve registrar um histórico das medidas realizadas.

REQFUN_16

O sistema deve permitir a visualização do Histórico de Requisições com todas as Requisições Solicitadas, em ordem decrescente, listando os seguintes parâmetros: Número da Resposta, Número da Requisição, Resultado, Tipo de Comando, Satélite, Estação, Usuário, Número de Grupos e Datação.

REQFUN_17

O sistema deve permitir a visualização dos 10 últimos Grupos de Medidas listando os seguintes parâmetros: Número da medida, Número da resposta, Datação UTC, Número da Mensagem, Contador Totalizador, Contador de Intervalo e Frequência, com unidade em Hertz.

REQFUN_18

O sistema deve permitir a geração de um arquivo com os 10 últimos Grupos de Medidas no formato XML, listando os seguintes parâmetros: Número da medida, Número da Resposta, Datação UTC, Número da Mensagem, Contador Totalizador, Contador de Intervalo e Frequência, em Hertz.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Tabela 1. Teste de Função.

Parte do Componente:	System Under Test é o componente na parte de Software onde aplicaremos o teste.
Objetivo da Técnica:	Validar a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.
Técnica:	<p>Executar as funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se:</p> <ul style="list-style-type: none"> os resultados esperados ocorrerão quando forem usados dados válidos as mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos cada regra de negócio será aplicada de forma adequada.
Estratégias:	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p> <p><i>A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i></p>
Ferramentas Necessárias:	<p>A técnica exige a seguinte ferramenta:</p> <p>A JPlavisFSM que é uma nova versão da antiga plataforma PLAVIS.</p>
Critérios de Êxito:	<p>A técnica suporta o teste de:</p> <ul style="list-style-type: none"> todos os principais cenários de caso de uso; todos os principais recursos.
Considerações Especiais:	-

5.1.3 Teste da Interface do Usuário

[O Teste da Interface do Usuário (UI) verifica a interação do usuário com o software. A meta do teste de UI é assegurar que a UI forneça ao usuário o acesso e a navegação adequados através das funções do objetivo do teste. Além disso, o teste de UI assegura que os objetos contidos na UI funcionem conforme o esperado e estejam em conformidade com padrões corporativos ou da indústria]

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Tabela 2. Teste de Interface do Usuário.

Parte do Componente:	<i>[Descrever qual a parte do componente no qual este teste será aplicado, parte hardware ou parte software]</i>
Objetivo da Técnica:	<i>[Experimentar o seguinte para observar e registrar a conformidade com padrões e o comportamento-alvo: A navegação pelo objetivo do teste para verificar se reflete os requisitos e as funções de negócios, incluindo a navegação janela a janela e campo a campo, e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras). Os objetos e as características poderão ser experimentados como, por exemplo, tamanho, posição, estado e foco]</i>
Técnica:	<i>[Crie ou modifique testes para cada função e verificar e os estados de objeto apropriados]</i>
Estratégias:	<i>[Descreva uma ou mais estratégias que possam ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina o método através do qual a observação pode ser feita e as características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto-verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados]</i>
Ferramentas Necessárias:	<i>[A técnica necessita da Ferramenta de Automação de Scripts de Teste. A estratégia deve levar em consideração itens e resultados que serão utilizados para o inter-relacionamentos de seus resultados entre a parte software e a parte hardware do componente]</i>
Critérios de Êxito:	<i>[A técnica suporta o teste de cada tela ou janela principal que será muito usada pelo usuário final]</i>
Considerações Especiais:	<i>[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste]</i>

5.1.4 Teste de Carga

Não se aplica.

5.1.5 Teste de Stress

Não se aplica.

5.1.6 Teste de Volume

Não se aplica.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

6. Critérios de Entrada e de Saída

6.1 Plano de Teste

Base às funcionalidades requeridas e listadas, elaboramos as listas dos eventos e ações (com código e descrição) e detalhamento das ações e requisitos funcionais a serem validados são apresentados na Tabela 3.

Tabela 3. Listas de eventos e ações com descrição da ação e número do requisito.

Lista de Eventos		Lista de Ações		Descrição Detalhada da ação e Número do Requisito	
Código	Descrição do Evento	Código	Descrição da ação	Descrição Detalhada	Número REQFUN
bReqMed	Botão "Requisição" pressionado	configMed	Configurar parâmetros de Medidas	Configurar os parâmetros: Centro de Missão Estação Receptora Satélite Número de Grupos Janela	2, 3, 4, 5, 6 e 7
bExeMed	Botão "Executar Requisição" pressionado	resultMed	Apresentar Resultados das Medidas	Apresentar a página com resultados das medidas: As 10 últimas requisições Respostas as requisições Grupos de medidas Registrar o histórico de medidas no Banco de Dados	8, 9, 10, 11,12,13 e 14
bRelMed	Botão "Relatório Medidas" pressionado	relatoMed	Apresentar Relatórios de Medidas	Apresentar a página com Relatórios: Relatório de Medidas Grupos de medidas Permitir a geração de arquivo XML	15, 16,17
bXML	Botão "XML" pressionado	gerRelXML	Gerar Relatório XML	Gera o relatório no formato XML e permitir o download	18
-	-	idle	nenhuma ação	Não há possibilidade desta ação, pois não existe evento disponibilizado no menu	-

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

A Tabela 4 apresenta os estados do SUT do MSC.

Tabela 4. Estados do MSC.

Lista de Estados	
s0	Página de Operação
s1	Página de Requisição de Medida
s2	Página de Resultados das Medidas
s3	Página de Relatório de Medidas
s4	Página de Geração Relatório

Apresentamos a MEF Completa para SUT, que através da ferramenta gerou alguns grupos de casos de uso.

6.1.1 MEF COMPLETA PARA SUT

6.1.1.1 ANÁLISE DAS PROPRIEDADES

A avaliação das propriedades da MEF completa, conforme Figura 1, determina as seguintes características: Determinística, Totalmente Especificada, Reduzida, inicialmente conexa e fortemente conexa.

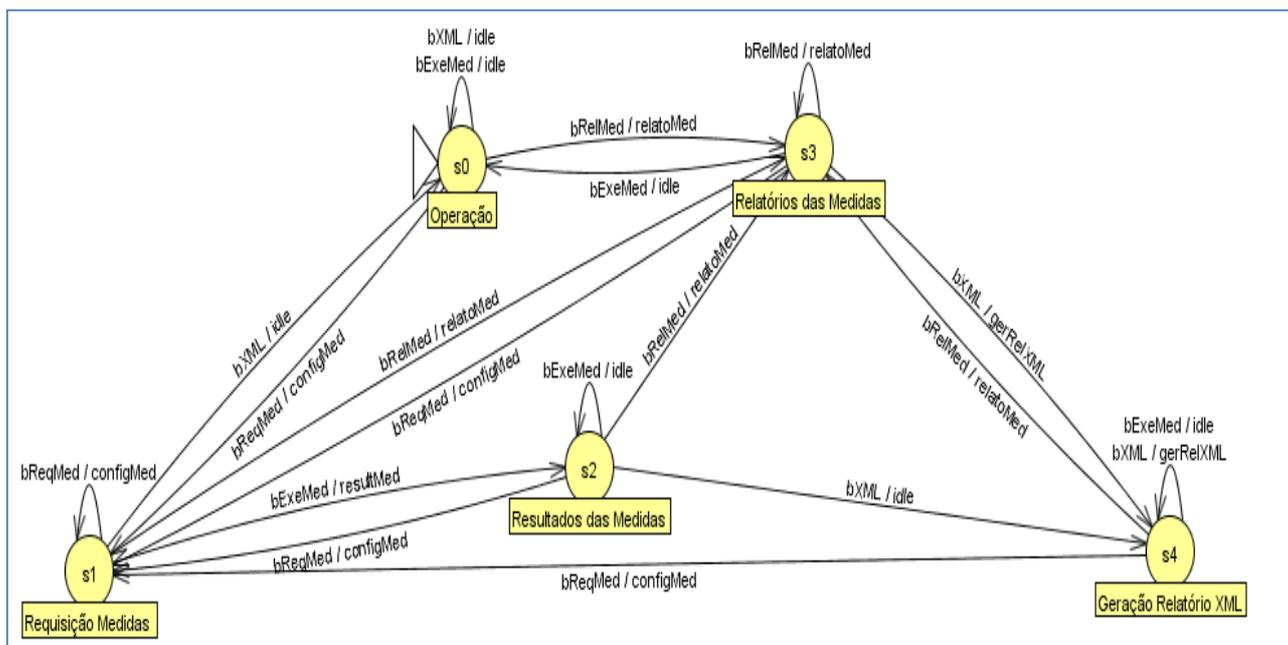


Figura 1 – MEF Completa.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

6.1.1.2 TABELA DE ESTADOS E EVENTOS PARA MEF COMPLETA

A Tabela 5 abaixo apresenta a lista de estados e eventos para MEF completa do SUT, foram criadas as transições e saídas necessárias para que a MEF apresente as propriedades para a geração dos casos de teste e continue a representar o modelo sob análise.

Tabela 5. MEF completa do MSC.

Estado	s0	s1	s2	s3	s4
Evento					
bReqMed	configMed	configMed	configMed	configMed	configMed
	s1	s1	s1	s1	s1
bRelMed	relatoMed	relatoMed	relatoMed	relatoMed	relatoMed
	s3	s3	s3	s3	s3
bExeMed	idle	resultMed	idle	idle	idle
	s0	s2	s2	s0	s4
bXML	idle	idle	idle	gerRelXML	gerRelXML
	s0	s0	s4	s4	s4

6.1.1.3 GERAÇÃO DE CASOS DE TESTES PARA MEF COMPLETA

6.1.1.3.1 CASOS DE TESTE – MÉTODO W

Neste método a JPlavisFSMFSM gerou o conjunto de casos de teste com entradas e saídas apresentado no apêndice B e com os resultados apresentados na Tabela 6

Tabela 6. Método W da MEF completa

Método	W
Nº de Casos de Teste	32
Nº de Mutantes	188
Nº de mutantes Mortos	188
Nº de mutantes Vivos	0
Nº de equivalentes	0
Score	1.0

6.1.1.3.2 CASOS DE TESTE – MÉTODO HSI

Neste método a JPlavisFSM gerou o conjunto de casos de teste com entradas e saídas apresentado no apêndice C e com os resultados apresentados na Tabela 7.

Tabela 7. Método HSI da MEF completa

Método	HSI
Nº de Casos de Teste	32
Nº de Mutantes	188
Nº de mutantes Mortos	188
Nº de mutantes Vivos	0
Nº de equivalentes	0
Score	1.0

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

6.1.1.3.3 CASOS DE TESTE – MÉTODO SPY

Neste método a JPlavisFSM gerou o conjunto de casos de teste com entradas e saídas apresentado no apêndice D e com os resultados apresentados na Tabela 8.

Tabela 8. Método SPY da MEF completa

Método	SPY
Nº de Casos de Teste	17
Nº de Mutantes	188
Nº de mutantes Mortos	188
Nº de mutantes Vivos	0
Nº de equivalentes	0
Score	1.0

6.1.1.3.4 7.3.4 CASOS DE TESTE – MÉTODO UIO

Neste método a JPlavisFSM gerou o conjunto de casos de teste com entradas e saídas apresentado no apêndice E e com os resultados apresentados na Tabela 9.

Tabela 9. Método UIO da MEF completa

Método	UIO
Nº de Casos de Teste	7
Nº de Mutantes	188
Nº de mutantes Mortos	97
Nº de mutantes Vivos	91
Nº de equivalentes	0
Score	0.51

6.1.1.3.5 7.3.5 CASOS DE TESTE – MEF COMPLETA MÉTODO MANUAL E MUTANTES

O conjunto de casos de teste, método manual, contra aos mutantes é apresentado no apêndice F.

Este conjunto de casos é derivado dos casos de teste do método W, que apresenta 32 casos de testes. O critério de seleção dos casos de teste cobrem as funcionalidades do modelo, eliminando as saídas “idle”, portanto temos 05 casos de testes aplicados contra os mutantes com os resultados na tabela 10.

Tabela 10. Método Manual da MEF contra os mutantes.

Método	Manual
Nº de Casos de Teste	5
Nº de Mutantes	188
Nº de mutantes Dead	81
Nº de mutantes Alive	107
Nº de equivalentes	0
Score	0.43

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

6.1.1.3.6 7.3.6 CASOS DE TESTE – MÉTODO MANUAL E IMPLEMENTAÇÃO

O conjunto de casos de teste, método manual contra a implementação é apresentado no apêndice G e com os resultados apresentados na Tabela 11.

No anexo G também são apresentadas as páginas da aplicação real de acordo com os requisitos e tabela de estado.

Tabela 11. Método Manual da MEF contra implementação.

Método	Manual
Nº de Casos de Teste	2
Nº de Mutantes	188
Nº de mutantes Dead	39
Nº de mutantes Alive	149
Nº de equivalentes	0
Score	0.20

6.1.1.4 COMPARAÇÃO ENTRE MÉTODOS MEF COMPLETA

A Tabela 12 apresenta a síntese dos métodos utilizados para a MEF completa com o número de casos de testes gerados, número de mutantes, mutantes vivos e mortos, número de equivalentes e scores.

Tabela 12. Comparação entre os métodos para a MEF completa.

Método	W	HSI	SPY	UIO	W**	Manual I
Nº de Casos de Teste	32	32	17	7	5	2
Nº de Mutantes	188	188	188	188	188	188
Nº de mutantes Dead	188	188	188	97	81	39
Nº de mutantes Alive	0	0	0	91	107	149
Nº de equivalentes	0	0	0	0	0	0
Score	1.0	1.0	1.0	0.51	0.43	0.20

**Retiramos os casos de testes com a saída "idle", cuja as entradas não tinham significado prático.

6.1.1.5 COMPARAÇÃO ENTRE MÉTODOS W, HSI, SPY, UIO E MANUAL, MEF COMPLETA

A Figura 6 apresenta o gráfico com o resultado do score para os métodos W, HSI, SPY, UIO e manual para a MEF completa.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

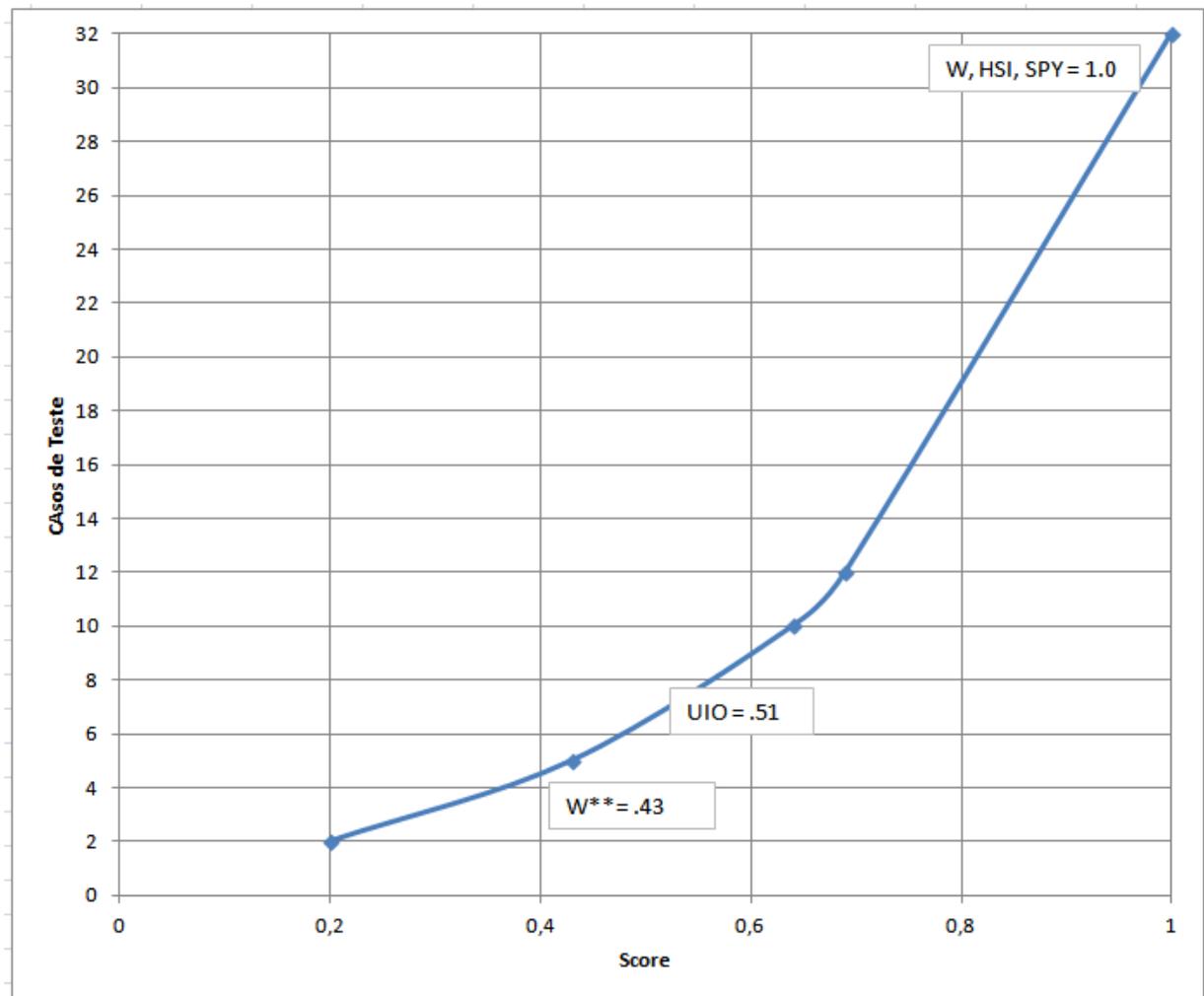


Figura 2 Comparação entre os métodos W, HSI, SPY, UIO e manual

6.1.2 CONCLUSÃO MEF COMPLETA

A MEF, do SUT em análise contempla as seguintes propriedades: determinística, completa, inicialmente conexa e fortemente conexa e mínima. Esta MEF permite a execução direta na JPlavisFSM dos métodos W, HSI, SPY e UIO e a conseqüentemente a comparação entre os resultados de cada método.

6.1.3 CONCLUSÃO GERAL DE CASOS DE TESTES

Os métodos de geração automática de testes consistem em uma ferramenta útil para diminuir o custo com o projeto de teste. Embora os métodos possuam um objetivo comum de verificar se uma implementação está correta com sua especificação, eles diferem com relação ao custo da geração das sequências de teste, tamanho do conjunto de teste e eficácia, sendo a eficácia a capacidade de detecção máxima de defeitos existentes na implementação a partir de testes gerados que devem ser relativamente pequenos para que seja possível sua aplicação na prática.

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

O método W (Chow, 1978), garante que se a implementação da MEF gerar saídas corretas a partir das sequências de entrada geradas pelo método W, esta máquina está correta, pois o método é confiável para testar estruturas de controle modeladas por uma MEF. Entretanto, o método W produz muitas sequências de entradas, o que promove um alto custo para a aplicação dos testes.

O método HSI (Petrenko et al., 1993), uma modificação do método W, garante a cobertura de defeitos existentes, sendo aplicável em qualquer especificação reduzida (mínimal), seja ela completa ou parcial.

O método SPY (Simão et al. 2009c), reduz o tamanho dos conjuntos de teste. A completude do conjunto é garantida pela verificação de que as várias sequências levam a um mesmo estado na implementação, assim, o método SPY evita que muitas sequências sejam adicionadas ao conjunto, aumentando seu tamanho.

Para que o testador selecione um método de geração com o objetivo de aplicá-lo em especificações baseadas em MEF, é necessário atender as exigências de cada método em relação as propriedades, ao tamanho da sequências e o custo de aplicação do conjunto de testes.

No SUT em análise, conforme tabela 12 - Comparação entre os métodos para a MEF completa, o método mais adequado para uma aplicação prática é o SPY, considerando o score de 1.0 e o menor número de casos de testes gerados, 17. Deve-se considerar os resultados dos métodos W e HSI como referência teórica dos casos de teste, pois garantem a correta implementação da máquina.

Outra maneira de orientar o testador é a avaliação do gráfico da figura 2, por exemplo se for aceitável um score de 50% teremos 8 casos de testes para o método UIO ou ainda para método W**(W com exclusão dos casos de testes com a saída "idle", cuja as entradas não tinham significado prático).

Uma aplicação direta de casos de testes com MEF, durante a criação do modelo, é realimentação de requisitos na fase de concepção de um projeto.

6.1.4 CONCLUSÃO GERAL

O documento de especificação representa o comportamento de um sistema, um modelo. A modelagem permite identificar o que um sistema deveria fazer e qual o resultado

6.1.5 Critérios de Entrada de Plano de Teste

6.1.6 Critérios de Saída de Plano de Teste

[Especifique os critérios que serão usados para determinar se a execução do Plano de Teste foi concluída ou se a continuação da execução não será vantajosa]

6.1.7 Critérios de Suspensão e de Reinício

[Especifique os critérios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos antes que o plano tenha sido totalmente executado. Especifique também segundo que critérios os testes poderão ser reiniciados]

7. Produtos Liberados

[Nesta seção, liste os vários artefatos que serão criados pelo esforço de teste e que serão produtos liberados úteis aos vários envolvidos do esforço de teste. Não liste todos os produtos do trabalho; liste apenas os que propiciam benefícios diretos tangíveis aos envolvidos e os que permitem medir o êxito do esforço de teste.

[Para esta seção levar considerar as orientações do item 6.1]

7.1 Sumários de Avaliação de Testes

[Forneça um breve resumo da forma e do conteúdo dos sumários de avaliação de testes e indique com que frequência eles serão gerados]

7.2 Relatórios da Cobertura de Teste

[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a extensão do teste e

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

indique com que frequência eles serão gerados Forneça uma indicação referente ao método e às ferramentas usadas para registrar, medir e reportar a extensão do teste]

7.3 Relatórios da Qualidade Perceptível

[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a qualidade perceptível do produto e indique com que frequência eles serão gerados. Forneça uma indicação referente ao método e às ferramentas usadas para registrar, medir e reportar a qualidade perceptível do produto. Você poderá incluir análises dos Incidentes e Solicitações de Mudança ao longo da Cobertura de Teste]

7.4 Registros de Incidentes e Solicitações de Mudança

[Forneça um breve resumo do método e das ferramentas usadas para registrar, rastrear e gerenciar incidentes dos testes, as solicitações de mudança associadas e seus status]

7.5 Conjunto de Testes de Regressão e Scripts de Teste de Suporte

[Forneça um breve resumo dos recursos dos testes que serão distribuídos para permitir testes de regressão contínuos dos builds subsequentes do produto, a fim de ajudar a detectar as regressões na qualidade do produto]

7.6 Produtos de Trabalho Adicionais

*[Nesta seção, identifique os produtos de trabalho que são opcionais ou os que não deverão ser usados para medir ou avaliar a execução bem-sucedida do **Plano de Teste**]*

7.6.1 Resultados Detalhados dos Testes

[Trata-se de um conjunto de planilhas do Microsoft Excel relacionando os resultados determinados para cada caso de teste ou refere-se ao repositório dos registros de testes e dos resultados determinados mantidos por um produto de teste especializado]

7.6.2 Scripts de Teste Funcionais Automatizados Adicionais

[Estes scripts consistem em um conjunto de arquivos de código-fonte para scripts de teste automatizados ou no repositório do código-fonte e dos executáveis compilados para scripts de teste mantidos pelo produto de automação de testes]

7.6.3 Guia de Teste

[O Guia de Teste abrange um amplo conjunto de categorias, incluindo Catálogos de Idéias de Testes, Orientações de Práticas Adequadas, Padrões de Teste, Modelos de Erros e de Falhas, Padrões de Design de Automação, etc.]

8. Fluxo de Trabalho de Teste

*[Forneça um resumo do fluxo de trabalho a ser seguido pela equipe de Teste no desenvolvimento e na execução deste **Plano de Teste**. Para esta seção levar considerar as orientações do item 6.1]*

9. Necessidades Ambientais

*[Esta seção apresenta os recursos não humanos necessários ao **Plano de Teste** (Hardware e Software)]*

9.1 Hardware Básico do Componente

[Os conjuntos de tabelas a seguir apresentam os recursos do componente necessários ao esforço de teste descrito neste Plano de Teste.

Esta que esta seção deve ser preenchida ao logo do tempo conforme o necessário]

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

Recursos do Componente		
Recurso	Quantidade	Nome e Tipo

10. Responsabilidades, Perfil da Equipe e Necessidades de Treinamento

[Esta seção apresenta os recursos necessários para abordar o esforço de teste descrito no Plano de Teste — as principais responsabilidades e os conjuntos de conhecimentos ou de habilidades exigidos desses recursos. (Hardware e Software)]

10.1 Pessoas e Papéis

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

[Observação: Adicione ou exclua itens conforme o necessário.]

Recursos Humanos		
Papel	Recursos Mínimos Recomendáveis (número de papéis alocados em tempo integral)	Responsabilidades ou Comentários Específicos

10.2 Perfil da Equipe e Necessidades de Treinamento

[Esta seção resume como abordar o perfil da equipe e o treinamento dos profissionais que ocuparão os papéis de teste no projeto.]

[O modo como abordar o perfil da equipe e o treinamento dos profissionais varia de projeto para projeto. Se esta seção integrar um Plano de Teste Mestre, indique em que pontos do ciclo de vida do projeto serão necessárias diferentes habilidades e um número diferente de integrantes da equipe]

11. Marcos do teste

[Identifique os principais marcos da programação que definem o contexto do Esforço de Teste. (Hardware e Software)]

Marco	Data de Início Planejada	Data de Início Real	Data de Término Planejada	Data de Término Real
Término do teste				

INPE - Instituto Nacional de Pesquisas Espaciais	Version: 1.0
Software de Controle - Conjunto de Medida de Velocidade	Date: 06/08/2014
Plano de Teste	

12. Procedimentos e Processos de Gerenciamento

*[Resuma os processos e os procedimentos que deverão ser usados quando surgirem problemas no **Plano de Teste** e em sua execução (Hardware e Software)]*

12.1 Medição e Avaliação da Extensão do Teste

[Resuma o processo de medição e avaliação a ser usado para rastrear a extensão do teste]

12.2 Avaliação dos Produtos Liberados deste Plano de Teste

*[Resuma o processo de avaliação para revisar e aceitar os produtos liberados deste **Plano de Teste**]*

12.3 Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções

[Defina como os problemas referentes a processos serão relatados, como serão selecionadas pessoas para resolvê-los e o processo a ser seguido para se chegar a uma solução]

12.4 Estratégias de Rastreabilidade

[Refleta sobre estratégias de rastreabilidade adequadas referentes a:

Cobertura de Teste em relação às Especificações — possibilita a medição da extensão do teste

Motivações de Teste — possibilitam a avaliação da relevância dos testes a fim de ajudar a determinar se eles deverão ser mantidos ou não

Elementos de Design de Software — possibilitam o rastreamento das mudanças de design subseqüentes que exigirão que os testes sejam executados novamente ou sejam cancelados

Solicitações de Mudança Resultantes — fazem com que os testes que descobriam a necessidade da mudança sejam identificados e executados novamente para verificar se a solicitação de mudança foi efetuada com êxito]

12.5 Aprovação e Encerramento

[Resuma o processo de aprovação e liste os cargos (e os nomes dos ocupantes atuais) que deverão aprovar inicialmente o plano e encerre com a execução satisfatória do plano]

Anexo B – Estudo de caso Mauá

Simulador Spacewire

Citar – projeto II

Projeto Citar
Linha 4

Head: Vanderlei C. Parro
E-Mail: vparro@ieee.org
Phone: +55 11 4239 3167
Office: Praça Mauá, 01

Resumo

Desenvolvimento de um sistema de acionamento para motores sem escovas (Brushless) . O motor utilizado como motivação de estudos terá como diretriz um as condições de contorno impostas pela Plataforma Multi Missão (PMM) do INPE para rodas de reação.

Metas

Elaboração de placa protótipo para controle de motor sem escovas.

Encaminhar a equipe do CITAR sistema desenvolvido em linguagem VHDL.

Recursos para Meta 1

Material de consume: R\$ 10.000,00 (incluindo motor)

Material permanente: R\$ 10.000,00

Serviço de terceiros: R\$5.000,00

Bolsas: 3 engenheiros (recém formados) + 3 bolsas de IC + 1 bolsa de Segundo grau + especialista visitante com experiência em projetos espaciais.

Diárias

- 14 diárias nacionais

Milestones

Julho de 2014 - Kick off

Início das atividades

Agosto de 2014 - revisão 1

Reunião com especialista para especificação sistema.

Dezembro de 2014

Conclusão primeiro protótipo

Junho de 2015

Entrega do código VHDL.

Contrapartida IMT

12 horas de professor titular para coordenação.

Instituto Mauá de Tecnologia
Núcleo de Sistemas Eletrônicos Embarcados - NSEE

Especificação do Produto - Acionamento motor BLDC

Rodrigo Alvite Romano & Rafael Corsi
rromano@maua.com & rafael.corsi@maua.com

July 2, 2014



Contents

1	Introdução	5
1.1	Finalidade	5
1.2	Referências	5
2	Requisitos necessários para a aplicação em rodas de reação	6
2.1	Descrição Geral do Projeto	6
2.2	Requisitos	6
2.2.1	O controlador deve prover seis (6) saídas digitais, cujo valor médio possa ser controlado ao modular a largura de pulso desses sinais.	6
2.2.2	O sistema deve dispor de entradas digitais para a leitura de três sensores de posição (baseados no princípio de efeito Hall), dispostos em cada uma das fases do motor	6
2.2.3	O sistema deve prover entradas para a leitura de um encoder incremental de alta resolução	7
2.2.4	O torque fornecido pelo motor deve apresentar baixa oscilação	7
2.2.5	O controlador deve prever três modos operação: controle em malha aberta, controle de velocidade ou por torque.	7
2.2.6	O sistema deve operar em uma ampla faixa de velocidades.	7
2.2.7	Independente do modo de operação, o sistema de controle deve assegurar que o torque fornecido e a velocidade angular e temperatura não excedam valores limites, passíveis de configuração.	8
2.2.8	O sistema deve possibilitar o envio periódico e sob demanda de dados de housekeeping, como variáveis internas e modos de operação.	8
2.2.9	Além do modo de operação, outras configurações possíveis são: ganhos dos controladores, limites de torque/corrente e velocidade angular, resolução do encoder incremental, número de pares de polos do motor, entre outras.	8
2.2.10	Através da mesma interface de comunicação usada para o envio de dados de housekeeping, deve ser possível receber comandos de configuração do sistema	8
2.2.11	O sistema deve prover uma interface de comunicação serial assíncrona, dedicada para fins depuração, integração e testes. Esta interface deve ser capaz de enviar periodicamente ou sob demanda informações sobre variáveis internas do sistemas.	8
2.2.12	O sistema deve estar embarcado em uma FPGA.	8

2.2.13	Uma placa de circuito impresso deve abrigar a FPGA, conversores analógico-digital e outros componentes eletrônicos responsáveis pela implementação da interface de comunicação serial. O circuito da fonte de alimentação, assim como a eletrônica de potência devem estar em outra placa	8
2.2.14	Pontos em aberto:	8
2.3	Representação Gráfica do Produto	9

Revision History

Revision	Date	Author(s)	Description
0.0.1	24.6.14	Corsi	Criação do documento

1 Introdução

1.1 Finalidade

Especificar as principais características do sistema embarcado de acionamento de motores sem escova com principal aplicação em rodas de reação

1.2 Referências

listar documentos de input - IMT e CITAR

2 Requisitos necessários para a aplicação em rodas de reação

2.1 Descrição Geral do Projeto

O objetivo do projeto é desenvolver um código HDL para uso em FPGA capaz de realizar tanto o acionamento como o controle de motores de corrente contínua sem escovas (BLDC). Mesmo não sendo o foco do projeto, devemos para teste e validação desenvolver o circuito de potência necessário para acionar os diversos motores que pretendemos utilizar na etapa de desenvolvimento e validação.

2.2 Requisitos

2.2.1 O controlador deve prover seis (6) saídas digitais, cujo valor médio possa ser controlado ao modular a largura de pulso desses sinais.

As saídas digitais serão usadas para realizar o acionamento de motores sem escovas DC trifásicos. Assim, as saídas digitais realizaram o acionamento lógico de um inversor de frequência.

2.2.2 O sistema deve dispor de entradas digitais para a leitura de três sensores de posição (baseados no princípio de efeito Hall), dispostos em cada uma das fases do motor

A função primordial desses sensores de efeito Hall é determinar a posição absoluta do rotor e assim permitir o correto acionamento das fases do motor, tanto no caso de aceleração quanto no de frenagem. Não é vetada a hipótese de se utilizar os sinais desses sensores para estimar a velocidade de rotação do motor.

2.2.3 O sistema deve prover entradas para a leitura de um encoder incremental de alta resolução

A estimação da velocidade exclusivamente através de pulsos gerados pelos sensores de posição do motor é deficiente em baixas rotações. De acordo com a resolução de medida de velocidade necessária, o sistema realizará a medida da velocidade usando os sinais gerados por um encoder incremental. Também há a possibilidade de se utilizar este sensor, de forma complementar, na comutação de fases do motor.

2.2.4 O torque fornecido pelo motor deve apresentar baixa oscilação

No caso da aplicação do sistema em rodas de reação, a oscilação de torque fornecido pelo motor é indesejável por deteriorar o desempenho do controle de atitude. Assim, o modo de comutação entre as fases deve levar tal requisito em consideração, especialmente durante a operação em baixas rotações.

2.2.5 O controlador deve prever três modos operação: controle em malha aberta, controle de velocidade ou por torque.

No modo de controle em malha aberta, o valor médio das tensões de acionamento são determinadas em função de um sinal de comando entre 0 e 100%. Nos modos de controle de velocidade e torque, os sinais de acionamento são modulados a fim de seguir valores de referência de velocidade de rotação e torque (no eixo do rotor), respectivamente.

2.2.6 O sistema deve operar em uma ampla faixa de velocidades.

No modo de controle de velocidade o sistema deve ser capaz de controlar a velocidade de rotação em uma ampla faixa de valores, desde algumas dezenas até alguns milhares de rotações por minuto.

- 2.2.7 Independente do modo de operação, o sistema de controle deve assegurar que o torque fornecido e a velocidade angular e temperatura não excedam valores limites, passíveis de configuração.**
- 2.2.8 O sistema deve possibilitar o envio periódico e sob demanda de dados de housekeeping, como variáveis internas e modos de operação.**
- 2.2.9 Além do modo de operação, outras configurações possíveis são: ganhos dos controladores, limites de torque/corrente e velocidade angular, resolução do encoder incremental, número de pares de polos do motor, entre outras.**
- 2.2.10 Através da mesma interface de comunicação usada para o envio de dados de housekeeping, deve ser possível receber comandos de configuração do sistema**
- 2.2.11 O sistema deve prover uma interface de comunicação serial assíncrona, dedicada para fins depuração, integração e testes. Esta interface deve ser capaz de enviar periodicamente ou sob demanda informações sobre variáveis internas do sistemas.**
- 2.2.12 O sistema deve estar embarcado em uma FPGA.**
- 2.2.13 Uma placa de circuito impresso deve abrigar a FPGA, conversores analógico-digital e outros componentes eletrônicos responsáveis pela implementação da interface de comunicação serial. O circuito da fonte de alimentação, assim como a eletrônica de potência devem estar em outra placa**
- 2.2.14 Pontos em aberto:**

Vale a pena considerar a opção de realizar o sincronismo de acionamento, assim como a estimação da velocidade através do encoder incremental associado à medida de tensão contra-eletromotriz em cada umas das fases

2.3 Representação Gráfica do Produto

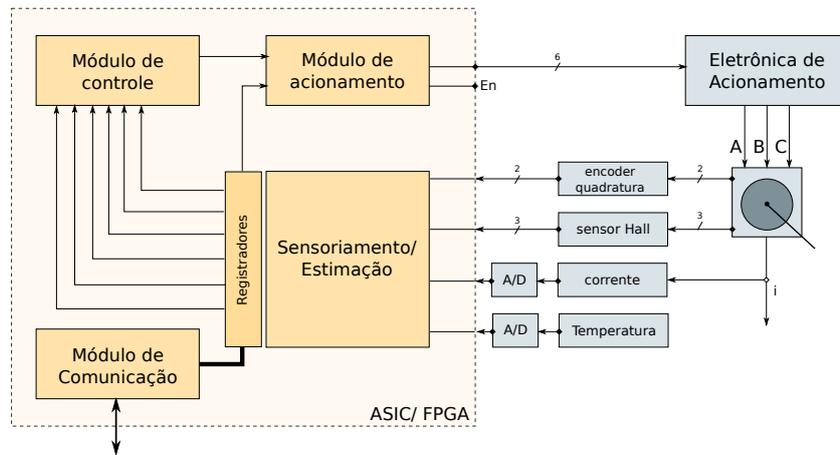


Figure 2.1: Proposta de implementação

Instituto Mauá de Tecnologia
Núcleo de Sistemas Eletrônicos Embarcados - NSEE

Especificação do Produto - Acionamento motor BLDC

Rodrigo Alvite Romano & Rafael Corsi
rromano@maua.com & rafael.corsi@maua.com

3 de julho de 2014



Sumário

1	Introdução	4
1.1	Escopo do documento	4
1.2	Documentos relevantes	4
1.3	Equipe	4
1.4	Cenário	5
2	Objetivos do projeto	6
2.1	Requisitos desejáveis para a aplicação em rodas de reação	6
2.2	Circuito configurável para ampla faixa de motores	6
3	Implementação	7
3.1	Módulo de sensoriamento	8
3.1.1	Torque	8
3.1.2	Posição e velocidade	9
3.1.3	Corrente	9
3.1.4	Temperatura	9
3.2	Módulo de controle	10
3.2.1	Operação em malha aberta	10
3.2.2	Controle de velocidade	10
3.2.3	Controle de torque	11
3.2.4	Controle de corrente	11
3.3	Módulo de acionamento	11
3.3.1	Motor	11
3.3.2	Corrente	12
3.4	Modos de operação	12
3.5	Interface de comunicação	13
3.6	Transcrição para HDL	13
4	Testes e Validação	15
5	Fluxo de desenvolvimento	16
5.1	Fases do projeto	16
5.2	Ferramentas	16
5.3	Gestão	16
	Referências Bibliográficas	16

Revision History

Revision	Date	Author(s)	Description
0.0.1	24.6.14	Corsi	Criação do documento

Lista de distribuição

inserir emails

- IMT
 - Fernando Martins
 - Rafael Corsi Ferrão
 - Rodrigo A. Romano
 - Sergio Ribeiro Augusto
 - Vanderlei Cunha Parro
 - José Carlos
 - Magda
 - Cauê Garcia Menegaldo
 - Cesar Scarpini Rabak
 - Juliano T. A. Laganá Pinto
 - Raphael Ballet
- INPE
 - Valdemir Carrara
 - Hélio ...
- Externo
 - Gerrald
 - Leonardo ?

1 Introdução

1.1 Escopo do documento

Especificar a meta do projeto, os pontos a serem abordados e a solução proposta, assim como a metodologia abordada.

1.2 Documentos relevantes

listar documentos de input - INPE e IMT

1.3 Equipe

- Profissionais do IMT:
 - MSc. Fernando Martins (hardware e dispositivos de validação)
 - Eng. Rafael Corsi Ferrão (software)
 - Dr. Rodrigo A. Romano (coordenação)
 - Dr. Sergio Ribeiro Augusto (hardware de validação)
- Bolsistas:
 - Eng. Cauê Garcia Menegaldo (software)
 - MSc. Cesar Scarpini Rabak (hardware e software)
 - Eng. Juliano T. A. Laganá Pinto (modelagem e simulação)
 - Raphael Ballet (software)
- Consultores:
 - Eng. Gerard Epstein (Engenharia de sistemas)
 - Dr. Valdemir Carrara (Aplicação de BLDC em rodas de reação)

1.4 Cenário

O projeto está sendo desenvolvido no âmbito do projeto de Circuitos Integrados Tolerantes a Radiação (CITAR), coordenado pelo pesquisador Saulo Finco do Centro de Tecnologia da Informação Renato Archer (CTI).

2 Objetivos do projeto

Desenvolver um dispositivo embarcado em uma FPGA capaz de realizar tanto o acionamento como o controle de motores de corrente contínua sem escovas (BLDC). Mesmo não sendo o foco do projeto, devemos para teste e validação desenvolver o circuito de potência necessário para acionar os diversos motores que pretendemos utilizar na etapa de desenvolvimento e validação.

2.1 Requisitos desejáveis para a aplicação em rodas de reação

Devido a aplicação em controle de atitude, é necessário que opere com:

- Baixa oscilação de torque
- Controle com pouca variabilidade em ampla faixa de velocidades
- Flexibilidade para operar em diferentes modos

Verificar com Carrara

2.2 Circuito configurável para ampla faixa de motores

O projeto deve operar com diferentes tipos de motores ...

3 Implementação

A proposta de solução atual pode ser subdividida em cinco principais blocos:

- Eletrônica de acionamento
- Módulo de Sensoriamento e Estimação
- Módulo de controle
- Módulo de acionamento
- Módulo de comunicação

para padronizar-mos

O diagrama de blocos é ilustrado na Fig. 3.1, onde em azul aparecem os blocos que não são de tema do projeto mas que são necessários para teste e validação.

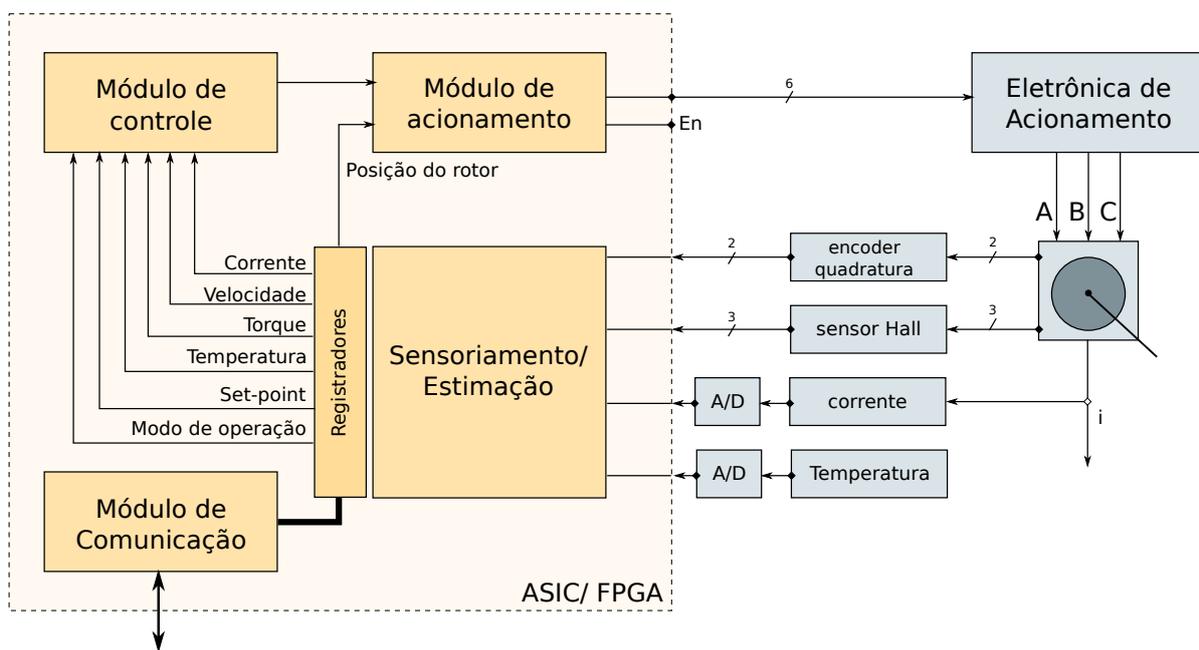


Figura 3.1: Diagrama de blocos da arquitetura do sistema

3.1 Módulo de sensoriamento

Processar sinais de diferentes sensores e disponibilizar informações a outros módulos do sistema.



Figura 3.2: Módulo de sensoriamento e estimação

hall

o Módulo possui diversos sub-módulos, dentre eles: estimação de torque, estimação de posição e velocidade, corrente e temperatura. Deve além de tratar os dados, lidar com os diferentes periféricos conectados ao chip.

3.1.1 Torque

Uma das maiores dificuldades no desenvolvimento de rodas de reação é a estimação do torque útil (T_u) fornecido pela roda, esse torque é resultante do torque aplicado pelo motor menos os atritos viscoso e estático causado tanto pelo mancal (T_m), quanto pelo atrito do disco com o ar (T_a)*, ou seja:

$$T_u = T - T_a - T_m$$

Estimação atrito

Uma das técnicas de estimação de atrito é proposta pelo Valdemir Carrara em [1], onde utiliza um filtro de Kalman estendido para identificar a fricção de Stribeck friction assim como atritos viscoso e de Coulomb.

*algumas rodas estão em uma atmosfera controlada, essa atmosfera serve para evitar que os lubrificantes entrem em evaporação

TBD. 3.1.1: Estimação de atrito

- é necessário estimar atrito online ou podemos trabalhar com a ideia que os atritos serão levantados em uma fase de integração e calibração
- estudar outras técnicas de estimação de atrito
- verificar o impacto da utilização no aumento de complexidade

3.1.2 Posição e velocidade

Outro ponto importante do módulo é o calculo da posição do rotor com relação ao estator, esse dado é importante para que o módulo de acionamento acione corretamente as fases. A posição pode ser derivada dos sensores de efeito HALL e do encoder de quadratura.

Posição

Pode-se a partir dos sensores hall estimar a posição absoluta do encoder e com isso obter uma posição mais precisa do rotor.

A velocidade é um dos modos de operação da roda de reação, e portanto deve ser estimada com precisão, pode ser estimada de duas maneiras distintas: pelo período entre pulsos (tanto do encoder quanto do hall) como também com a utilização de um modelo do motor.

TBD. 3.1.2: Robustez

o módulo deve ser to para funcionar sem algum dos sensores ?

3.1.3 Corrente

A corrente é importante para a malha de controle de torque e velocidade, o módulo deve fornecer o valor filtrado da corrente.

TBD. 3.1.3: Corrente sensor

temos que definir se a corrente vai ser por fase ou total

3.1.4 Temperatura

Por operar basicamente por irradiação de calor, a constante medição da temperatura é importante para evitar operações que possa vir a comprometer o motor, além disso,

pode-se com a temperatura identificar possíveis falhas no sistema.

TBD. 3.1.4: Temperatura sensores

definir quantos sensores de temperatura e qual tipo - PT100, CI, ...

3.2 Módulo de controle

Determinar a intensidade do acionamento das fases do motor de acordo com um determinado modo de operação. Os modos de operação são:

- Operação em malha aberta
- Controle de velocidade
- Controle de torque

TBD

Porém é necessário um controle de corrente para correta operação do sistema.

3.2.1 Operação em malha aberta

Nesse modo de operação, o controle das intensidades nas fases do motor é controlado via a interface de comunicação.

TBD. 3.2.1: Malha de corrente em malha aberta

Definir se a malha de corrente continua operante quando o sistema opera em malha aberta, com isso garantimos que não haverá nenhum sobressinal que possa danificar o motor.

3.2.2 Controle de velocidade

Nesse modo de operação, o sistema é responsável por controlar a velocidade de rotação do motor com base em sinais de referências enviados via a interface de comunicação, Fig. 3.2.

TBD. 3.2.2: Controle de velocidade

Definir malha de controle de velocidade

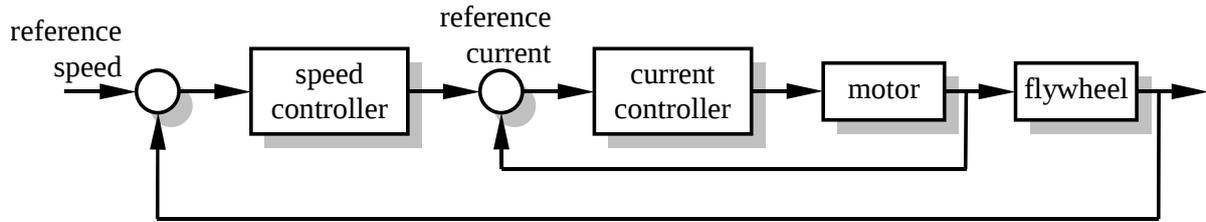


Figura 3.3: Malha de velocidade com controle de corrente

3.2.3 Controle de torque

Nesse modo de operação, o sistema é responsável por controlar o torque útil do motor, utilizando os parâmetros de atrito já calculados (via o estimador ou via testes de bancada).

TBD. 3.2.3: Controle de torque

Definir malha de controle de torque

3.2.4 Controle de corrente

O controle de corrente é importante para evitar que ocorra picos de corrente no motor e que a corrente atinja mais rapidamente o valor desejado.

TBD. 3.2.4: Controle de corrente

Definir malha de controle

3.3 Módulo de acionamento

Como motores sem escovas não possuem a comutação mecânica predefinida na fabricação, deve-se escolher um método apropriado de comutação visando maior estabilidade do sistema e maior eficiência. Para tanto é importante saber a posição do rotor com relação ao estator para que as energizações das bobinas ocorram de forma síncrona. A relação entre a frequência de chaveamento das bobinas e a velocidade do motor é dado por:

3.3.1 Motor

Como motores sem escovas não possuem a comutação mecânica predefinida na fabricação, deve-se escolher um método apropriado de comutação visando maior estabi-

lidade do sistema e maior eficiência. Para tanto é importante saber a posição do rotor com relação ao estator para que as energizações das bobinas ocorram de forma síncrona. A relação entre a frequência de chaveamento das bobinas e a velocidade do motor é dado por:

$$F_{polesw} = \frac{\text{número de fases} * \text{número de pólos}}{\text{velocidade do rotor}}$$

O acionamento das fases do motor é realizado em função dos valores da posição do rotor e a intensidade das fases. Existem diversas formas de acionamento, algumas delas são:

- Quadrada
- Trapezoidal
- Senoidal
- PWM Vetorial - SVPWM

TBD. 3.3.1: Formas de acionamento

Definir quais as formas de acionamento que vão ser implementadas e como fazer o chaveamento entre elas se for necessário

Uma dificuldade no acionamento de BLDC é a partida do motor onde a posição do rotor é desconhecida. Para solucionar esse problema, o motor é colocado em uma situação forçada de movimento onde uma sequencia predefinida é executada. Após o início do movimento é possível detectar a posição do rotor e então o controle é alterado da forma forçada (malha aberta) para a forma síncrona (malha fechada).

3.3.2 Corrente

Acionar o circuito responsável pelo controle de corrente das fases do motor.

TBD. 3.3.2: Acionamento do controle de corrente

O controle de corrente pode ser realizado de duas maneiras distintas, ou via a tensão nas fases do motor ou via um circuito chopper na linha de alimentação dos motores.

3.4 Modos de operação

Os modos de operação do sistema:

- OFF : Todos os periféricos estão desligados

- Init : Nesse modo, a hardware é inicializado, incluindo as configurações dos módulos e periféricos
- Stand-by : O sistema está totalmente operante porém nenhuma ação é tomada quanto ao acionamento do motor
- Config : **TBD**
- MA : Modo de operação em malha aberta
- MV : Modo de operação em malha fechada de velocidade
- MT : Modo de operação em malha fechada de torque
- ERRO : Modo de erro

3.5 Interface de comunicação

Permitir o monitoramento e a programação do sistema por meio de uma interface de comunicação serial, para a padronização do bus de comunicação em satélites pretende-se utilizar o protocolo SpaceWire como forma de comunicação.

A interface deve:

- Prover informações sobre o funcionamento do dispositivo:
 - Velocidade de rotação
 - Torque gerado
 - Temperatura do estator
- Configurações do sistema:
 - Seleção do modo de operação
 - Envio de sinais de referência para o módulo de controle
 - Reprogramação de parâmetros de sintonia do sistema de controle

TBD. 3.5.1: Protocolo de comunicação

Definir protocolo, (CCSDS) PUS ou RMAP.

3.6 Transcrição para HDL

Com o objetivo de obtermos no final um hardware dedicado para acionamento de motores BLDC devemos transcrever a lógica de acionamento para uma linguagem de descrição de hardware (HDL).

TBD. 3.6.1: Metodologia

- Verificar especificação com CITAR
- TestBenchs

4 Testes e Validação

Todos os testes de validação serão realizados considerando a aplicação do sistema no controle de rodas de reação. Serão avaliados nos testes os requisitos do sistema, tais como : oscilação de torque, velocidade, passagem pelo zero, acompanhamento do sinal de referência, estimadores, ... **TBD**.

Prevê testarmos os algoritmos propostos primeiramente utilizando um sistema de HIL (DSPACE) que possibilitara testarmos a lógica proposta em uma interface de alto nível (MATLAB - Simulink) para então receber as otimizações necessárias para a transcrição em HDL.

Outra utilização prevista é a de utilização do HIL para simulação do motor e teste do acionamento proposto, esse cenário é interessante para a execução de certos testes que seriam de difíceis em bancada, como por exemplo, o teste de diferentes tipos de motores, a validação dos estimadores com os dados reais (simulados).

Com o HIL é possível também inserir o acionamento em um contexto aerospacial, simulando a dinâmica de um satélite controlado por um conjunto de rodas de reação que estão sendo acionadas pelo acionamento externo.

Para a validação como aplicação em rodas de reação, um sistema composto por um motor BLDC mais um componente de inercia e uma eletrônica serão desenvolvidos e testados na mesa com mancal hidrostático do LABSIM. Essa avaliação demonstrara o efeito do acionamento no controle de atitude de um satélite.

Cenários de validação:

- Teste em bancada
- Simulação híbrida
- Mesa com mancal hidrostático (INPE)

5 Fluxo de desenvolvimento

5.1 Fases do projeto

5.2 Ferramentas

5.3 Gestão

Referências Bibliográficas

- [1] Valdemir Carrara, Adolfo Graciano Da Silva, and Hélio Koiti Kuga. A dynamic friction model for reaction wheels. *Advances in the Astronautical Sciences*, 145:343–352, 2012.

Anexo C – Estudo de caso Mauá

MEMO

	Name	Company
From:	R. Haarmann	OHB
To:	PLATO Consortium	
CC:	DLR Berlin	
Subject:	Comments on PLATO Document and Configuration Management Procedure (PLATO-ESTEC-MAN-PRO-001, issue 1, 25/09/2014)	
Info:	<p>General: All involved PLATO entities must use the Document Numbering systematic explained in PLATO-ESTEC-MAN-PRO-001 [AD09]. Each entity is responsible for its own document configuration management. All official new documents (especially if part of the PDCR or SRR data package or references to that) must get a new document number following [AD09]. Documents from the M1/M2 phase, which are revised and reissued during this phase B1, must get a new document number, according to PLATO-ESTEC-MAN-PRO-001. The change must be recorded in the change log. Documents, which remain unchanged from the M1/M2 phase and are for reference only can maintain their document number.</p> <p>Numbering System Acc. [AD09] §4.2.2 the following document numbering systematic is used for the PLATO Project:</p> <p style="text-align: center;">PLATO-„Originator“-“Project Element“-„Doc Type“-„Number“</p> <p>Originator [AD09] §4.2.2.1: Abbreviation between 2-4 Digits; the given list in [AD09] can be extended. Each entity is requested to decide for one unique abbreviation, which must be maintained during the whole project (e.g. KT, OHB, DLR, INAF, CAB, LAM, MSSSL, UBE (Uni Bern), KUL (Uni Leuven), LESIA, IAS, etc.)</p> <p>Project Element [AD09] §4.2.4: 2-3 Digits The given table is extended, to distinguish between Payload, PDC and PSPM documents. Following abbreviations shall be used: PL: Payload (already contained in list) PDC: Plato Data Center PSPM: Plato Science Preparation Management</p> <p>Doc Type [AD09] §4.2.5:</p>	

The abbreviations acc. The given list must be used
Number: 3-digits, serially numbered
 e.g -001, -002, -003,....

This results in following types of Document Numbers (exemplary):
 e.g. for KT/OHB PL-documents:
 PLATO-KT-PL-TN-0001
 PLATO-KT-PL-TS-0005
 PLATO-OHB-PL-RP-0010
 e.g. for PL-Documents of partners:
 PLATO-INAF-PL-DDF-0001
 PLATO-LAM-PL-RQ-0005
 PLATO-DLR-PL- PL-0014
 e.g. for Plato Data Center documents:
 PLATO-MPG-PDC-PL-0001
 PLATO-KUL-PDC-RP-0005
 e.g. for PSPM documents:
 PLATO-UBE-PSPM-RP-0019
 PLATO-UGE-PSPM-TN-0158

	Name	Signature
Prepared by:	R. Haarmann (OHB)	
Checked by:	H. Rauer (DLR)	



DOCUMENT

PLATO Document Configuration and Management Procedure

Prepared by Philippe Gondoin, David Agnolon
Reference ESA-PLATO-ESTEC-MAN-PRO-001
Issue 1
Revision 0
Date of Issue 25/09/2014
Status Approved/Applicable
Document Type PR
Distribution SRE-F, SRE-S, SRE-O, HSO-O, HSO-G



APPROVAL

Title	
Issue 1	Revision 0
Author Philippe Gondoin, David Agnolon	Date 25/09/2014
Approved by	Date

CHANGE LOG

Reason for change	Issue	Revision	Date

CHANGE RECORD

Issue 1		Revision 0	
Reason for change	Date	Pages	Paragraph(s)



Table of contents:

1 INTRODUCTION.....4

2 APPLICABILITY4

3 PLATO REQUIREMENT DOCUMENTATION5

4 DEFINITION OF DOCUMENT REFERENCE CONVENTION8

4.1 Definitions8

4.2 Reference System and File Naming Convention for Documents and Data Files9

4.2.1 ESA users9

4.2.1.1 Referencing – Document identification code9

4.2.1.2 Originator field9

4.2.2 Non-ESA users9

4.2.2.1 Referencing – Document identification code9

4.2.2.2 Originator field10

4.2.3 File names10

4.2.4 Project Element10

4.2.5 Document Type Code11

4.2.6 Numbering12

4.2.7 Non-compliance to the Document identification code – exceptions13

4.3 Document Configuration Control.....13

4.3.1 Registering Documents and Reference Numbering13

4.3.2 Revisions14

5 ECLIPSE PROCEDURES 15

5.1 General 15

5.2 Document Configuration and Change Management System 15

5.2.1 Document download..... 15

5.2.2 Document generation 17

5.2.3 Document additional data..... 20

5.2.4 Document upload and release..... 21

5.2.5 Document up issue 21

5.3 eRID – Review Item Discrepancy System..... 21

Table of figures:

Figure 3-1: PLATO top-level requirement specification tree..... 5

Figure 3-2: Spacecraft development specification tree..... 6

Figure 3-3: Payload development specification tree 7

Figure 5-1: Access SRE-F PLATO project.....15

Figure 5-2: Access DCCM function.....15

Figure 5-3: Browse documents16

Figure 5-4: Open and download documents17

Figure 5-5: Edit documents..... 18

Figure 5-6: Document generation fields..... 18

Figure 5-7: Document ID fields19

Figure 5-8: Editing Document fields 20

Figure 5-9: document metadata.....21

Figure 5-10: Up-issue document.....21



1 INTRODUCTION

The purpose of the document is to provide all participants to the Phase B1 of the PLATO mission with requirements and guideline to implement and adhere to common document references. The aim of this uniform approach is to enable efficient control of documentation and communication, and to support a structured approach to a document management system.

The basis for the creation of document references is established through a breakdown of the originators, project elements, and document types. Its purpose is to facilitate the integration of data between the various project elements, attributing unique item identification, references and definitions. The references are defined on top-level, providing guidelines for a structured and coherent document configuration control.

The following procedures are applicable to all ESA-generated PLATO document configuration:

- [RD 1] ESA IMS Generic procedure for document management, ESA/IMS/N/DPR-20-005
Issue 1 Revision 0, 17/10/2012

2 APPLICABILITY

DCMP#010: The conventions and definitions provided in this document shall be applicable to all configuration controlled documentation related to the phase B1 of the PLATO mission, including documents originating from industry, instrument teams and ESA.

It is anticipated that a revised version of this document will be produced for Phase B2/C/D/E1.

In order to leave sufficient flexibility for individual non-ESA participants, it is allowed the use of additional codes or identifiers, provided that an overall compatibility with these guidelines is maintained.

3 PLATO REQUIREMENT DOCUMENTATION

An overview of the PLATO requirement documentation is given in the following document tree.

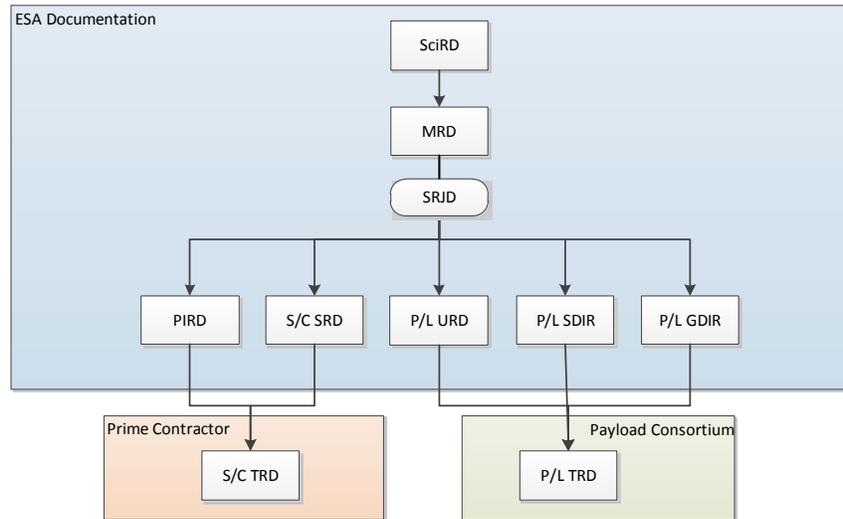


Figure 3-1: PLATO top-level requirement specification tree

ESA top-level requirement documents are as follows.

- The SciRD – Science Requirements Document specifies all the requirements needed to achieve PLATO science objectives (not necessarily space related or under ESA responsibility),
- The MRD – Mission Requirements Document specifies the mission requirements that are PLATO-related and under ESA responsibilities,
- The S/C SRD – Spacecraft System Requirements Document specifies the PLATO spacecraft Technical Requirements Specifications derived from the MRD that are under the spacecraft Prime Contractor responsibility,
- The P/L URD – Payload User Requirements Document specifies the PLATO payload Technical Requirements Specifications derived from the MRD that are under the PLATO Consortium responsibility,
- The SRJD – System Requirement Justification Document justifies the key S/C SRD and P/L URD functional and performance requirements,
- The PIRD – Payload Interface Requirements Document defines the payload interface requirements that the spacecraft shall comply with,
- The P/L SDIR – Payload Specific Design and Interface Requirements Document defines the interface requirements that the payload shall comply with (part of the “EID-A – Experiment Interface Document – Part A”),
- The P/L GDIR – Generic Design and Interface Requirements Document (part of the “EID-A – Experiment Interface Document – Part A”).

Note that historically, the Experiment Interface Requirements Document – Part A, was a single document grouping all sort of payload interface requirements. The EID-A for PLATO is a data package consisting of the following documents: SDIR, GDIR, URD (as well as PMRD (Programmatic and Management Requirements Document) not show above).

The PLATO spacecraft Prime Contractor responds to the ESA S/C SRD and PIRD with a PLATO spacecraft technical specification and derive the spacecraft subsystem specifications and interface requirements.

The PLATO payload Consortium respond to the ESA EID-A data package (i.e. URD, SDIR, GDIR, PMRD, PARD) with a PLATO payload technical specification and derive the payload subsystem specifications and interface requirements.

The following technical specifications tree describe the relevant specifications applicable to the Prime Contractor for spacecraft activities and to the Payload Consortium for payload activities. Management requirements are not described here.

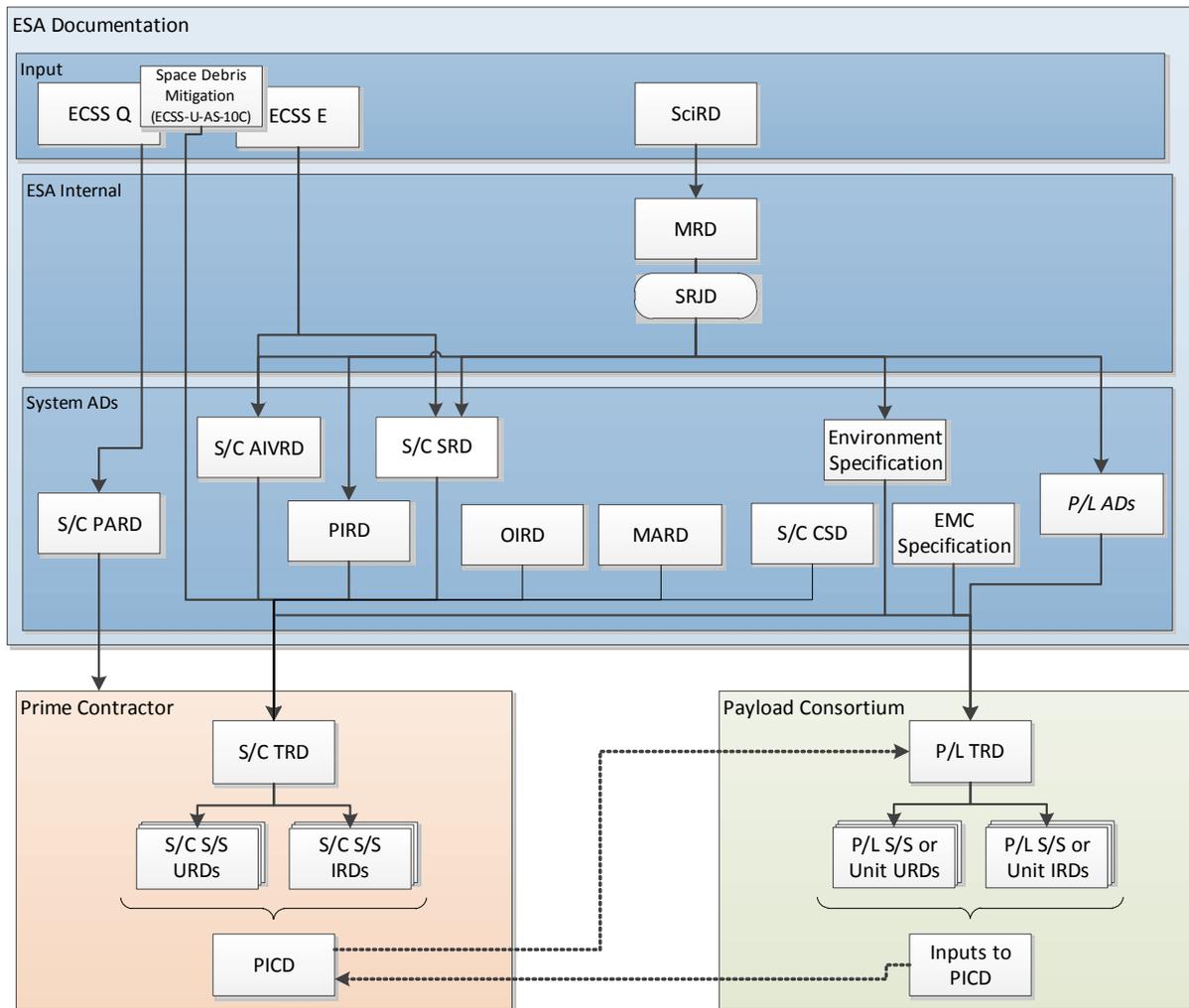


Figure 3-2: Spacecraft development specification tree

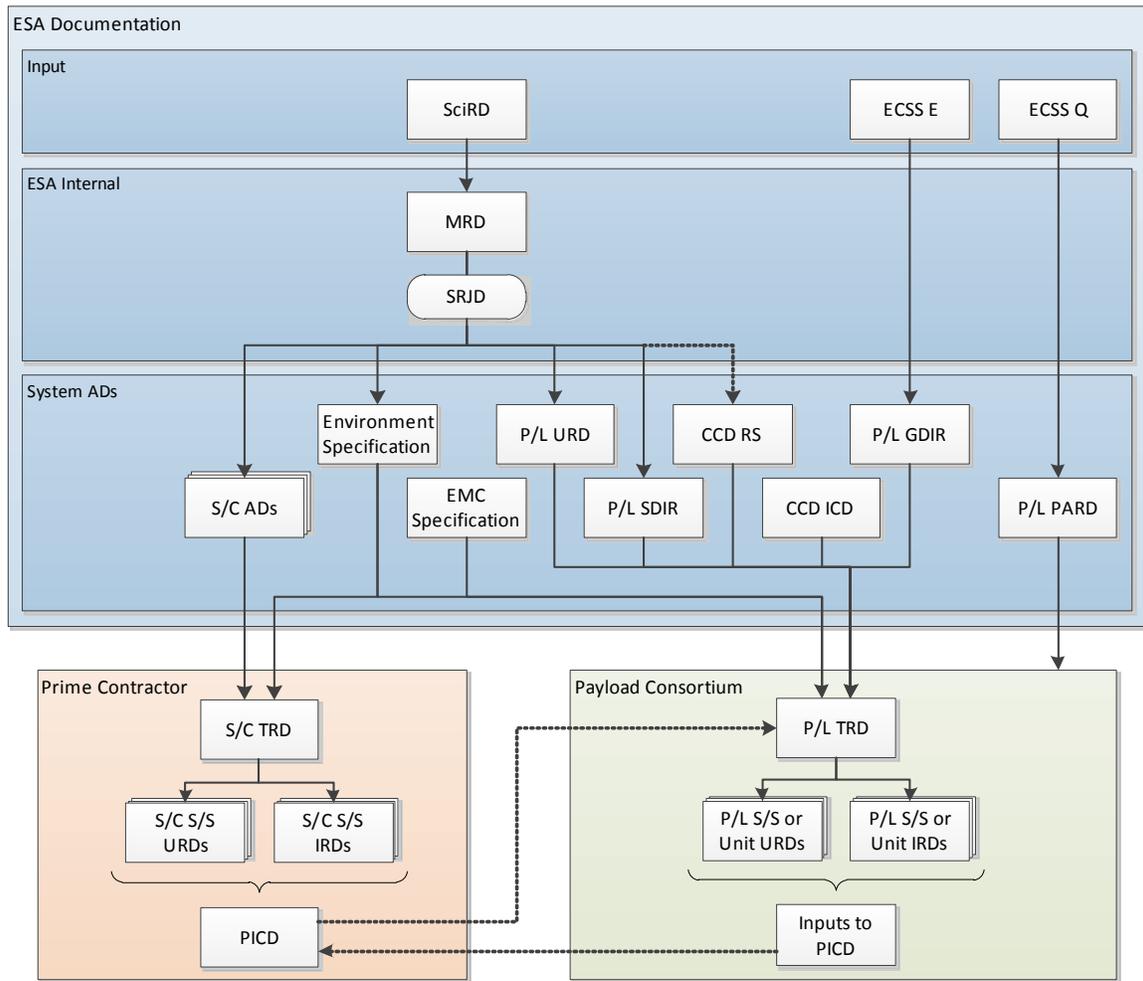


Figure 3-3: Payload development specification tree

The following specifications tree describe the relevant specifications applicable to the ground segment development under respectively ESAC and ESOC responsibilities.

To be written.



4 DEFINITION OF DOCUMENT REFERENCE CONVENTION

4.1 Definitions

DCMP#020 - All documents and data files shall be configuration controlled and shall be stored in the PLATO document management system as follows:

Document type	Req. for configuration control
1. Documents are narrative and/or descriptive (including inter-alia Technical Notes, Reports, Drawings, Minutes of Meetings, etc.) matter describing the formal status of a project element within the project or toward external contractors, scientific consortium and international participants.	All documents shall be submitted in electronic .PDF format such that the: - Document content is searchable - Table Of Content (TOC), where present, is available as bookmarks. See DCMP#150 for ESA users.
2. Correspondence covers communication and exchange of information via e-mail, fax, or surface mail. Examples of such communication items are Letters and Memos or the exchange of opinions or questions/answers.	Only written correspondence that carries a reference number shall be configuration controlled. Electronic signature should be used.
3. Data files and models are digital data that are not suitable e.g. as PDF format. This includes all kind of data files (CAD exchange files, trajectory files and other files useable by established computer programmes), computer programmes, etc.	All documents/information shall be registered in the document management system with a dedicated reference and version number.

DCMP#025 - The selected ESA documentation configuration tool for PLATO is the PRISMA ECLIPSE system.



4.2 Reference System and File Naming Convention for Documents and Data Files

DCMP#030 - The specifications in this subsection shall be applied for documents, data files and models (items 1 and 3 of section 4.1).

In order to comply with both the IMS procedure ([RD 1]) for ESA internal documents and the needs of the PLATO project for document management beyond ESA internal documents, two referencing systems are defined: 1. one for ESA users (documents created and maintained under ESA responsibility) and another 2. one for non-ESA users (e.g. Prime contractor, payload consortium, etc.).

It is possible to create a document without complying with the above referencing systems under special conditions. See chapter 4.2.7.

4.2.1 ESA users

4.2.1.1 Referencing – Document identification code

DCMP#040 - In order to identify unambiguously each document and data file the following Document Identification Code shall be adhered to by all ESA participants.

The documents will be identified by a Document Identification Code according to a system, which is made up of six elements separated by a dash symbol “-”:

ESA-PLATO-<originator>-<project element>-<document type>-<number>

Example: ESA-PLATO-ESTEC-SYS-PRO-001 for a procedure written at system level in ESTEC (i.e. spacecraft).

4.2.1.2 Originator field

A two to four letter-code is assigned for the originator of the Document. The following list shows the codes that have been currently allocated for ESA users.

HQ	HQ
ESAC	ESAC
ESOC	ESOC
ESTEC	ESTEC

Example: ESA-PLATO-ESTEC-XX-XX-XXX

4.2.2 Non-ESA users

4.2.2.1 Referencing – Document identification code

DCMP#040 - In order to identify unambiguously each document and data file the following Document Identification Code shall be adhered to by all non-ESA participants who are generating a file related to the PLATO project.

The documents will be identified by a Document Identification Code according to a system, which is



made up of five elements separated by a dash symbol “-“:

PLATO-<originator>-<project element>-<document type>-<number>

Example: PLATO-DLR-PL-PRO-001 for a procedure written at payload level in DLR.

4.2.2.2 Originator field

A two to four letter-code is assigned for the originator of the Document. The following list shows the codes that have been currently allocated for non-ESA users. This list can be amended at any time to account for new Originators joining the study:

To be defined	Contractor working on S/C
DLR	Deutsches zentrum fur Luft- und Raumfahrt
E2V	E2V
KT	Kayser-Threde
INAF	Istituto Nazionale di Astrofisica

Example: PLATO-INAF-XX-XX-XXX

The following chapters are applicable to both ESA and non-ESA users as long as the relevant document ID code as defined above is used.

4.2.3 File names

DCMP#050 - The file names shall comply to the following naming convention, using the “_” symbol as a separator:

<Document Identification Code>_i<issue>.<revision>_<Title>.<file extension>

The value stored in the <Title> may be an abbreviation of the actual document title.

Examples:

- ESA users: ESA-PLATO-ESTEC-MAN-PRO-001_i1.0_Doc_Config_Procedure.pdf
- Non-ESA users: PLATO-DLR-MAN-PRO-002_i1.0_Doc_Config_Procedure.pdf

DCMP#060 - The keywords making up the Document Identification Code are described hereafter. Only valid values shall be used, without the “<” and “>” symbols

4.2.4 Project Element

The first digit of the four-digit document number refers to the respective project element as shown below.

SCI	Science
SOC	Science Operations Center
MOC	Mission Operations Centre
LV	Launch Vehicle

CON	Contract
MAN	Management
AIV	AIV
PQA	Product Quality Assurance
PL	Payload
SYS	System
MIS	Mission

Example: ESA-PLATO-ESTEC-SYS-XX-XXX

4.2.5 Document Type Code

The document identification characters (two or three-letter) clarify the purpose and the objective of the document. The following codes have been allocated:

AD	Assumption Document
AG	Agenda
AN	Analysis
AO	Announcement of Opportunity
CCN	Contract Change Notice
CON	Contract/Rider
COS	Cost Documents (Estimate/CaC/CtC, etc)
DD	Design Description
DP	Data Package
DRD	Document Requirements Description/Definition
DRL	Document Requirements List
DW	Drawing/Diagram
EID	Experiment Interface Document
EX	Executive Summary
FI	File (Software/Configuration/Network)
HO	Handout/Presentation
ICD	Interface Control Document
IRD	Interface Requirement Document
ITT	Invitation to Tender
LE	Letter
LEG	Legal Text
LI	List
MAN	Manual/User Guide/Handbook
MEM	Memo
ML	Model
MOM	Minutes of Meeting
MOU	Agreement/Memorandum of Understanding
MX	Matrix/Compliance
NCR	Non-Conformance Report
NOT	Note

OPS	Operations Document
PLN	Plan
PO	Proposal
PRE	Progress Report/Status Report
PRO	Procedure
PT	Procedure
REG	Regulation
REC	Record
RP	Report (Technical, Budget, Cost, Manpower, Travel, Audit, etc.)
RFQ	Request for Quotation
RS	Requirement Document/Specification (System, Subsystem, Unit, Equipment level)
RW	Request for Waiver
RES	Resolution
SCH	Schedule/Network/Barchart/Chart
SP	Specifications
ST	Standards
SOW	Statement of Work
TOR	Terms of Reference
TN	Technical Note
TP	Test Procedure/Test Plan
TR	Test Report/Test Result
TS	Test Specification
VC	Verification Control Document
WBS	Work Breakdown Structure
WP	Working Paper
WPD	Work Package Description

4.2.6 Numbering

DCMP#070 - The first free sequential number per Document Identification Code (the full string as per section 3.2) shall be used.

Note: Therefore, for each type of Document Code, 999 document numbers can be assigned.

Example: ESA-PLATO-ESTEC-SYS-RS-001

- The reference number (last three digits “001” is sequential for each type of Document Identification Code (“ESA-PLATO-ESTEC-SYS-RS”)
- The sequential reference numbers are independent from the other elements of the Document Identification Code (e.g. ESA-PLATO-ESTEC-PF-RS-001 may follow ESA-PLATO-ESTEC-CON-CCN-010).



4.2.7 Non-compliance to the Document identification code – exceptions

DCMP#075 – In some exceptional cases, to be agreed with the ESA PLATO team, the document generator can create a document without the need to comply to the document ID code. In this case, the generator will use the following specific document generation fields in the ECLIPSE system (see chapter 5.2.2):

- **“PLATO Document with no reference”**: For all users who need to create a new document, have agreed with the ESA PLATO team that a document ID code is not required and who need a document reference.
- **“Free text”**: For all users who need to upload an existing document, have agreed with the ESA PLATO team that a document ID code is not required and who do not need a document reference (e.g. have your own document reference).

4.3 Document Configuration Control

DCMP#080 – Each originator (ESTEC, ESOC, non-ESA users) shall create and control his own sequential references and sequential numbers. Note: the PLATO phase B1 definition study team is responsible for managing all documents created within ESTEC, e.g. D/TEC-issued notes, etc.

In the case of the ESA study team, each team member is responsible for obtaining the relevant document number (see section 4.2.6) and for keeping the document under configuration control.

Outside the ESA study team, the originator’s project office will be responsible for the assignment of correct and sequential document references.

DCMP#085 – Only formal documents (e.g. deliverables of a contract, requirements specifications, procedures, etc.) or documents which need traceability shall be uploaded on ECLIPSE, i.e. technical notes, management reports, formal analysis, etc. Informal notes shall be exchanged by email or other means (e.g. ftp, industry servers, etc.).

4.3.1 Registering Documents and Reference Numbering

DCMP#090 - The selected ESA documentation system (Eclipse) can automatically assign unique reference numbers that are in-line with the constraints described in this document. All ESA study team members shall obtain a reference for their document from Eclipse. Once the reference number is obtained the document can be finalized and uploaded.

DCMP#095 - External participants shall comply with the above document reference system and file naming conventions. They are responsible for the assignment of the proper document references for documents produced under their responsibility and for their distribution into ECLIPSE.

DCMP#096 - The document generation fields described in chapter 5.2.2 shall be used for creating documents on ECLIPSE. The field to be used depends on the user’s affiliation. The **“PLATO Document with no reference”** and **“free text”** fields shall be used only in specific cases only to be agreed with the Agency, see chapter 4.2.7.



4.3.2 Revisions

DCMP#100 - When a Document is updated this shall either be identified by a new revision or by a new issue.

New issues or re-issues will be made when major changes in scope or in the text occur (e.g. class 1 changes affecting approved technical specifications, including interfaces, and terms of the business agreement between a customer and his supplier).

Revisions will be made for minor additions, deletions or class 2 changes (i.e. changes not identified as class 1) of information.

For the revisions of parts of a document, each page affected will acquire the next revision number compared to the previous revision of the page concerned for the same issue. Revisions are set to 0 in case of a re-issue of the document. Following the issue-number, and separated by a point, a number indicates the revision (*Example*: Issue: 1.1).

For the re-issue of a document the Issue number will be changed. Issue numbers are indicated by numerical characters. In the case of a draft document the first version is “Issue 0”, a second draft version is “Issue 0.1”, and so on.

DCMP#110 - Changes (deletions, modifications and additions) in the text of the document between issues and revisions shall be described.

DCMP#120 - The document change record sheet, which immediately follows the cover page, shall be kept up to date. It shall be updated with every revision / issue of the document. Changes related to previous issues of this document shall be deleted. Each approved new issue should have a new issue number and a signed copy should be made available.

DCMP#130 - After the change record sheet, a distribution sheet shall provide traceability to whom and when the document has been issued.

DCMP#140 - Revisions of existing documents shall be stored as new versions of the respective document in the document management system and not in a new entry.

DCMP#150 - The ESA study team shall also store the source file (WORD, EXCEL, MPP) of their documents with the final (signed) document (PDF) in the ESA documentation system.

5 ECLIPSE PROCEDURES

5.1 General

The direct address to access the PRISMA ECLIPSE system is <https://sre-eclipse.esa.int>. Enter your Login and password.

Login and passwords can be requested at prisma@esa.int or by request to the ESA PLATO team (david.agnolon@esa.int or philippe.gondoin@esa.int).

For access to the PLATO project, click on the SRE-F PLATO project in the top-left scroll-down menu as follows:



Figure 5-1: Access SRE-F PLATO project

The Eclipse system integrates various functions: AIM (Action Item Management), DCCM (Document Configuration and Change Management), eRID (RID system), MyEcl, PAM (Project Administration Module), etc. The eRID system may be used for PLATO reviews TBC. See chapter 5.3. A user guide can be found for each function in the help menu of each function.

5.2 Document Configuration and Change Management System

Most (ESA and non-ESA) users will only need access to the DCCM function (in particular to download and upload documents) reachable via the tab which can be found on the ECLIPSE interface at the top-right of the SRE-F PLATO Project as follows:

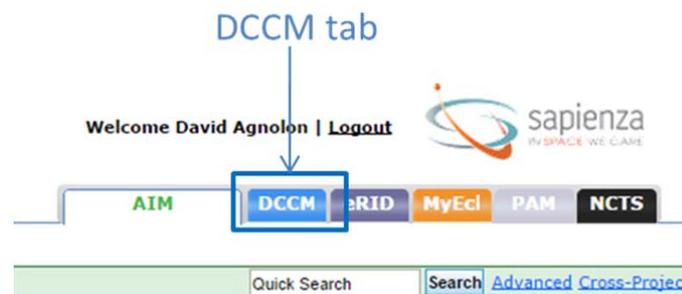


Figure 5-2: Access DCCM function

5.2.1 Document download

The various steps to create a document are described here below.

1. Access the SRE-F PLATO project as described above.

2. Access the DCCM function by clicking on the above shown **DCCM** tab.
3. Click on the “**Documents**” scroll-down menu and click on “**Browse**” as shown below

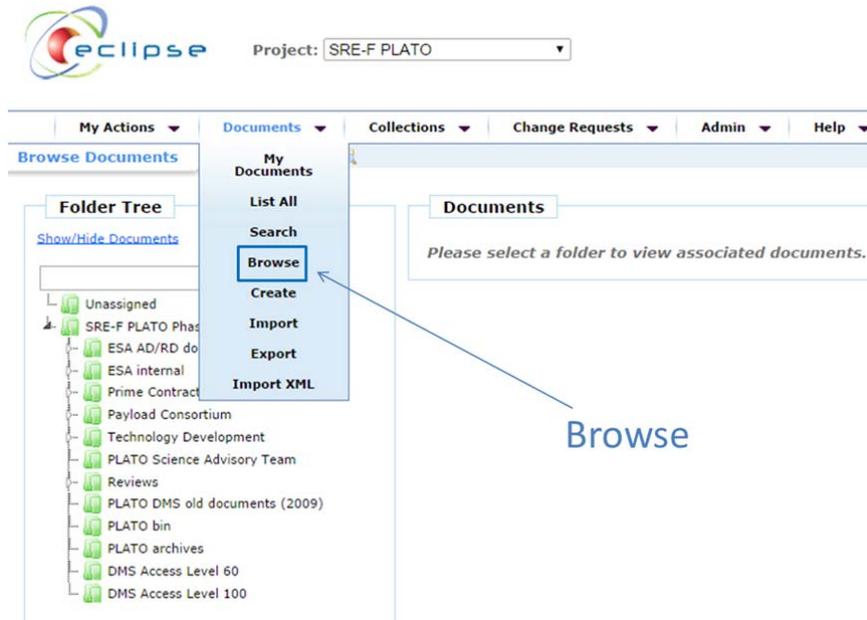


Figure 5-3: Browse documents

The left-handed menu allows you to browse through the folders to which you have access.

Note!!!: You can click on the arrows in front of the folders’ name to go to the lower-level folders. If there is no arrow, you have reached the lowest folder level.

- Green folders are folders to which you have both write and read access (i.e. you can upload documents).
- Blue folders are folders to which you have read access (i.e. you can download documents).
- Grey folders are folders to which you have no access.

When you click on the folder where the document you are looking for is located, you will see all documents belonging to this folder. Click on the document’s name of relevance to download it. See example below (ESA Mission Requirements Document in the ESTEC folder):

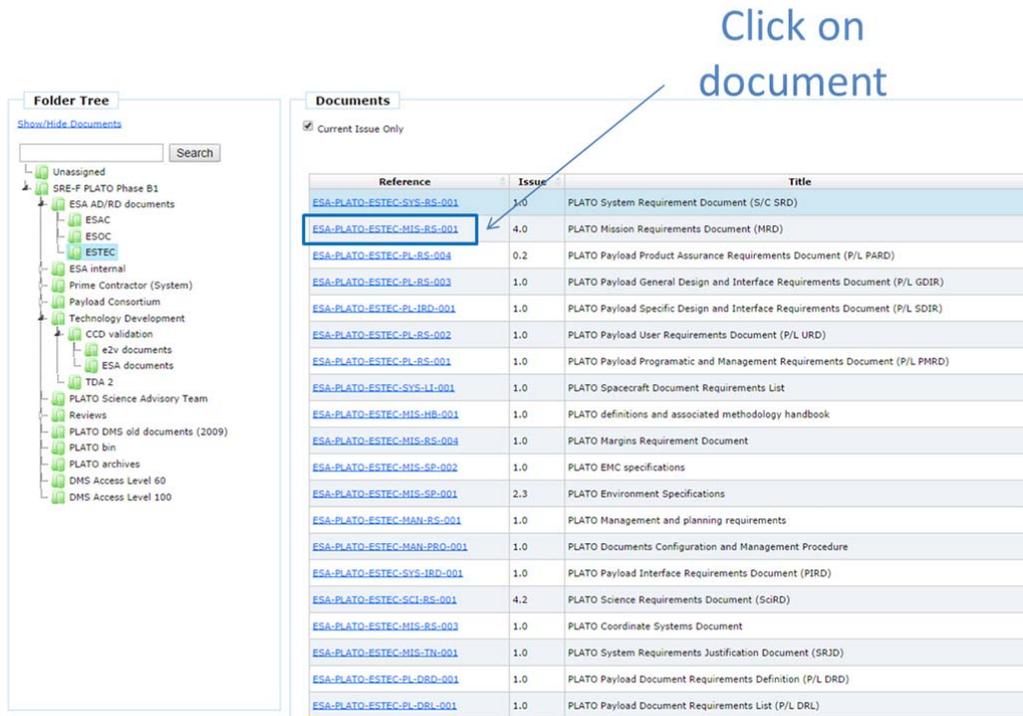


Figure 5-4: Open and download documents

5.2.2 Document generation

The various steps to create a document are described here below.

4. Access the SRE-F PLATO project as shown on Figure 5-1.
5. Access the DCCM function by clicking on the **DCCM** tab as shown on Figure 5-2.
6. Click on the **“Documents”** scroll-down menu and click on **“Create”** as shown below

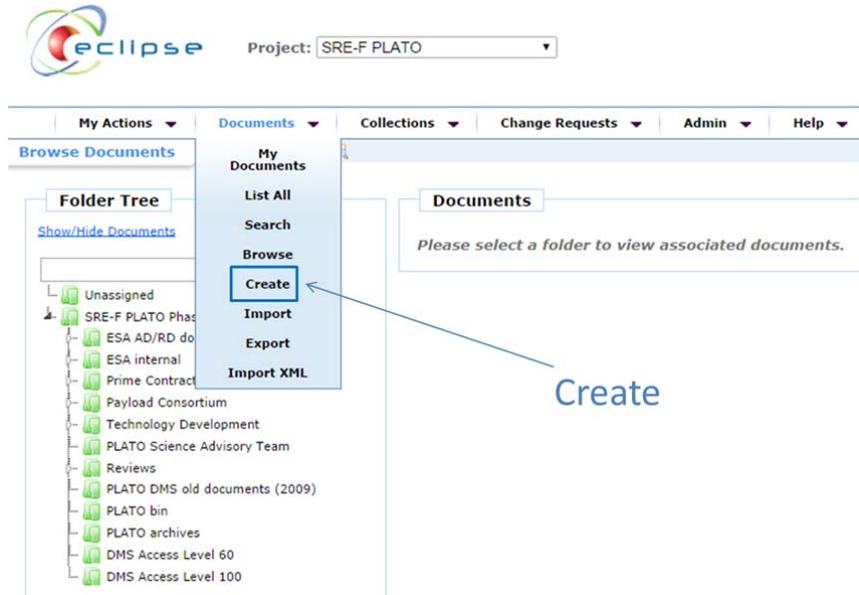


Figure 5-5: Edit documents

The following menu appears:

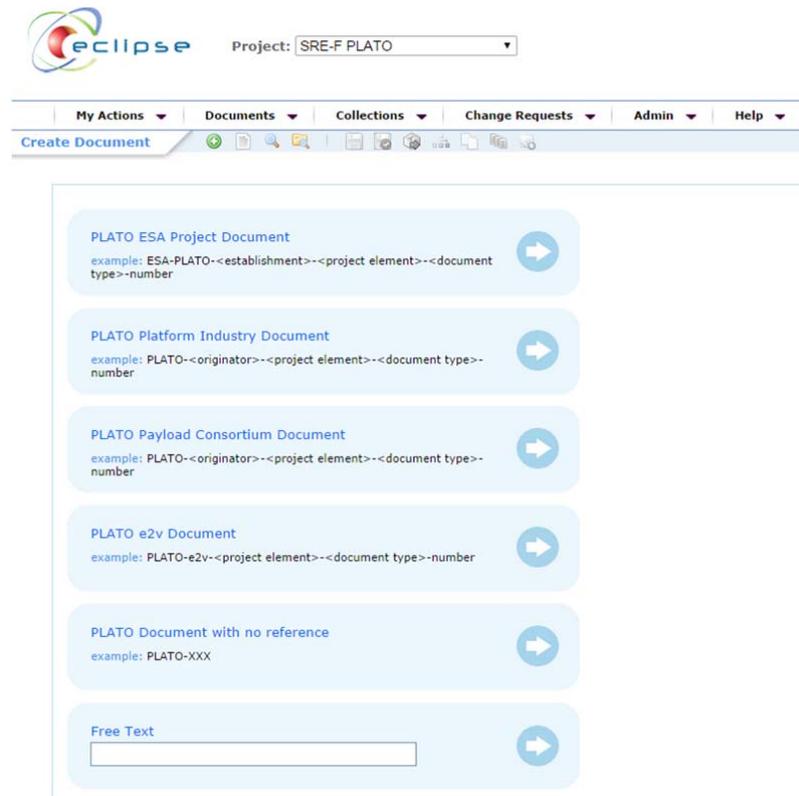
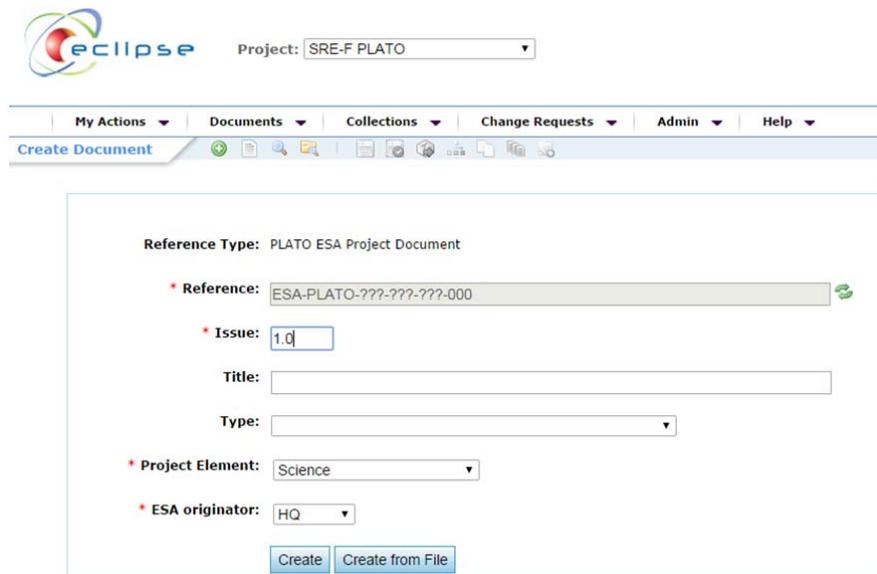


Figure 5-6: Document generation fields

7. Depending on the user category you belong to (see chapter 4.2) or the referencing system you would like to use, click on the relevant field:
 - a. If you are an ESA user and need to create a document with a new document identification code as described in chapter 4.2.1, click on the “**PLATO ESA-Project Document**” field,
 - b. If you are a Prime Contractor user and need to create a document with a new document identification code as described in chapter 4.2.2, click on the “**PLATO Prime Contractor Document**” field,
 - c. If you are a Payload Consortium user and need to create a document with a new document identification code as described in chapter 4.2.2, click on the “**PLATO Payload Consortium Document**” field,
 - d. If you are an e2v user and need to create a document with a new document identification code as described in chapter 4.2.2, click on the “**PLATO e2v Document**” field,
 - e. If you need to create a new document with no need for a document identification code, click on the “**PLATO Document with no reference**” field.
 - f. If you need to upload an existing (i.e. including existing reference number) document with no need for a document identification code, click on the “**free text**” field,

The following screen will appear:



The screenshot shows the Eclipse PLATO interface. At the top, there is a logo and a dropdown menu for 'Project: SRE-F PLATO'. Below this is a navigation bar with tabs for 'My Actions', 'Documents', 'Collections', 'Change Requests', 'Admin', and 'Help'. The main content area is titled 'Create Document' and contains the following fields:

- Reference Type: PLATO ESA Project Document
- Reference: ESA-PLATO-???-???-???-000
- Issue: 1.0
- Title: (empty text box)
- Type: (empty dropdown menu)
- Project Element: Science
- ESA originator: HQ

At the bottom of the form, there are two buttons: 'Create' and 'Create from File'.

Figure 5-7: Document ID fields

8. Fill in the following relevant fields as described in chapter 0 to create the relevant document identification code:
 - a. Issue
 - b. Document type
 - c. Project element
 - d. Originator (depending on the user, different originator options will appear)

The reference itself cannot be changed. Whichever field you have selected above, ECLIPSE will automatically generate the reference (document identification code), including the numbering, except for the “**free text**” field for which you are free to use whatever reference you are using or would like to use.

9. Click on “**Create**”, the following screen appears:

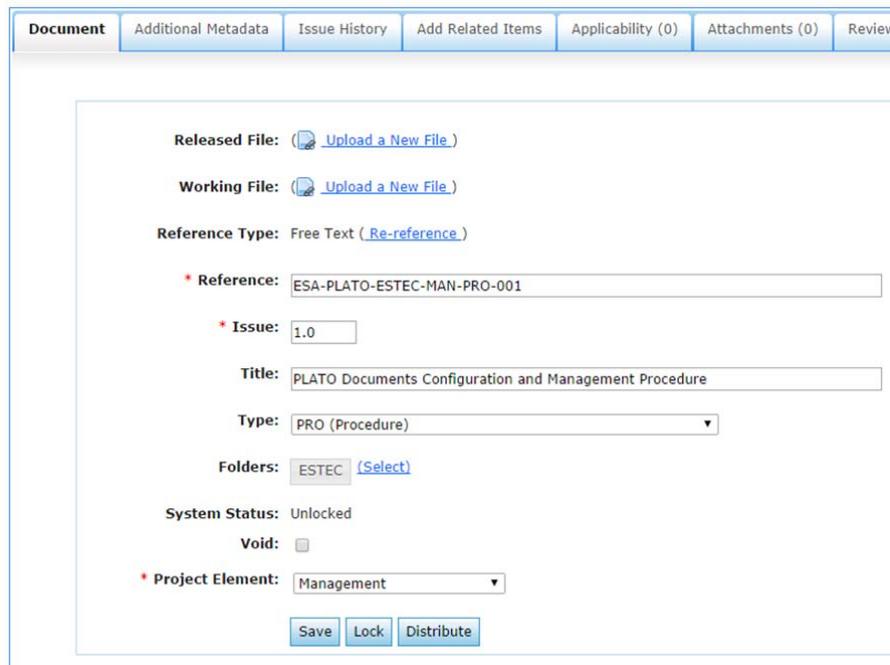


Figure 5-8: Editing Document fields

10. Upload the relevant file by clicking on “**Upload a new file**”, see chapter 5.2.4 in order to know if it is a “working” or “released” file.
11. Select the folder(s) in which you need to upload the document by clicking on “**select**”. A menu will appear allowing you to do that.
12. Click on “**Save**”.

5.2.3 Document additional data

Once you have created a document, you can enter metadata associated to it (e.g. author, issue date, comments). Click on the tab “**Additional metadata**” at the top of the screen above and the following screen appears on which you can enter the relevant data in each field.

Document Additional Metadata Issue History Add Related Items Applicability (0) Attachments (0) Review Ac

Originator / Author:

Issue Date:

Comments:

ESA originator: ESTEC

Industry generator: void

Consortium generator: void

Figure 5-9: document metadata

5.2.4 Document upload and release

On step 10 above, you are requested to upload the document. All non-released versions, or editable versions (.DOC, etc.), unsigned versions or documents which are still in a draft form (issue o.X) shall be uploaded in the “**Working file**” location. This is the location which shall be used for all files which are still in the iteration process, i.e. the issue has not been released.

When a file is due for release in its final form, it shall be uploaded in the “**Released file**” location.

Once a file is released, it shall be locked by clicking on the “**Lock**” tab at the bottom of the screen in Figure 5-8.

Important note: Once released/locked, this issue cannot be modified and an up-issue must be generated, see chapter 5.2.5.

5.2.5 Document up issue

Once a document has been released, only new issues can be generated. This can be done by going to the browse menu and to the folder where the document is located.

On the right-hand side of the document, click on the *grey wheel*. A menu will appear as shown below and the “**up-issue**” option will show on which you can click to up-issue the document.

Up-issue document

Reference	Issue	Title	System Status	Action Items	Last Updated	Actions
ESA-PLATO-ESTEC-MIS-SP-001	2.3	PLATO Environment Specifications	Locked		29-09-2014 11:53:02	
ESA-PLATO-ESTEC-SYS-RS-001	1.0	PLATO System Requirement Document (S/C SRD)	Unlocked		25-09-2014 16:16:16	
ESA-PLATO-ESTEC-MIS-RS-001	4.0	PLATO Mission Requirements Document (MRD)	Unlocked		25-09-2014 16:16:16	
ESA-PLATO-ESTEC-PL-RS-004	0.2	PLATO Payload Product Assurance Requirements Document (P/L PARD)	Unlocked		25-09-2014 16:16:16	
ESA-PLATO-ESTEC-PL-RS-003	1.0	PLATO Payload General Design and Interface Requirements Document (P/L GDIR)	Unlocked		25-09-2014 16:16:16	
ESA-PLATO-ESTEC-PL-IRD-001	1.0	PLATO Payload Specific Design and Interface Requirements Document (P/L SDIR)	Unlocked		25-09-2014 16:16:16	

Figure 5-10: Up-issue document

5.3 eRID – Review Item Discrepancy System

To be written.