MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2017/11.01.14.43-TDI

METRICS FOR OVERSIGHT OF SOFTWARE SUPPLIER OF SAFETY-CRITICAL AEROSPACE SYSTEMS

Benedito Massayuki Sakugawa

Doctorate Thesis of the Graduate Course in Space Engineering and Technology, guided by Drs. Ana Maria Ambrosio, and Carlos Henrique Netto Lahoz, approved in November 30, 2017.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34P/3PTG678>

INPE
São José dos Campos
2017

PUBLISHED BY:

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
E-mail: pubtc@inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2017/11.01.14.43-TDI

METRICS FOR OVERSIGHT OF SOFTWARE SUPPLIER OF SAFETY-CRITICAL AEROSPACE SYSTEMS

Benedito Massayuki Sakugawa

Doctorate Thesis of the Graduate Course in Space Engineering and Technology, guided by Drs. Ana Maria Ambrosio, and Carlos Henrique Netto Lahoz, approved in November 30, 2017.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34P/3PTG678>

INPE
São José dos Campos
2017

Cataloging in Publication Data

Sakugawa, Benedito Massayuki.

Sa29m Metrics for oversight of software supplier of safety-critical aerospace systems / Benedito Massayuki Sakugawa. – São José dos Campos : INPE, 2017.

xxii + 206 p. ; (sid.inpe.br/mtc-m21b/2017/11.01.14.43-TDI)

Thesis (Doctorate in Space Engineering and Technology) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2017.

Guiding : Drs. Ana Maria Ambrosio, and Carlos Henrique Netto Lahoz.

1. Software safety. 2. Aerospace system. 3. Software metric. 4. Civil aviation certification. 5. Software supplier oversight. I.Title.

CDU 629.78:004.056

Aluno (a): ***Benedito Massayuki Sakugawa***

Título: "METRICS FOR OVERSIGHT OF SOFTWARE SUPPLIER OF SAFETY-CRITICAL AEROSPACE SYSTEMS".

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

Engenharia e Tecnologia Espaciais/Eng. Gerenc. de Sistemas Espaciais

Dra. **Maria de Fátima Mattiello-Francisco**

Presidente / INPE / São José dos Campos - SP

Dra. **Ana Maria Ambrosio**

Ana Maria Ambrosio

Orientador(a) / INPE / São José dos Campos - SP

Dr. **Carlos Henrique Netto Lahoz**

Orientador(a) / IAE/DCTA / São José dos Campos - SP

Dr. **Ronaldo Arias**

Membro da Banca / INPE / São José dos Campos - SP

Dra. **Emilia Villani**

Convidado(a) / ITA / São José dos Campos - SP

Dr. **Edgar Toshiro Yano**

Edgar Toshiro Yano

Convidado(a) / ITA / São José dos Campos - SP

Este trabalho foi aprovado por:

( ) maioria simples

(X) unanimidade

São José dos Campos, 30 de novembro de 2017

DEDICATION

To Nae Oyadomari *(In Memoriam)*

ACKNOWLEDGEMENTS

First, I would like to thank my advisors, for their guidance, support and patience.

I would like to thank my close colleagues, for letting me abuse of their precious time.

I would also like to thank my colleagues from the aviation industry for their transparency, which has allowed improving my knowledge.

I am thankful to my close friends, for always reminding me that there is life even during the doctoral program.

And finally, special thanks to my family, for their love.

ABSTRACT

The current scenario shows a tendency to outsourcing space systems, and for managing presumed risk the oversight (supervision) of supplier increases in importance by identifying problems and nonconformities at earlier stages. Moreover, particularly for software the civil aviation performs some oversight-like activities (informally called audits) for certification purpose, but the criteria used for classifying the issues are not adequate for evaluating the audit result, and may mislead the management decision. The above scenario added by the proximity between space and aviation, gives opportunity to creating a mechanism able to capture the audit result, which can be used with confidence for management decision. This work presents metrics for oversight of software supplier of safety-critical aerospace system, called "Aerospace Metrics". The purpose of the Aerospace Metrics is to evaluate the oversight results for supporting management decision. The metrics are generated analytically by using the Goal-Question Metric (GQM) technique combined with the Reason's human error model. A survey is performed with software safety specialists from the civil aviation to providing additional information for metrics adjustment and evaluation. For evaluation in aeronautics, the generated metrics are applied to selected cases of aviation software audits, and evaluated against the related software certification history. For evaluation in astronautics, software safety systematic comparison between space and aviation is performed to identifying adjustments in the metrics and the oversight activities, due to space specific necessities. As case study, the adjusted oversight activities are applied to a space project called QSEE (Quality of Software Embedded in Space Applications), and the results are submitted to the metrics for evaluation. The work produced acceptable results, showing that the Aerospace Metrics are feasible, and can be applied to either aeronautics or astronautics with few adjustments due to specific necessities.

Keywords: software safety; aerospace system; software metric; civil aviation certification; software supplier oversight; software supervision;

MÉTRICAS PARA SUPERVISÃO DE FORNECEDORES DE SOFTWARE DE SISTEMAS AEROESPACIAIS DE SEGURANÇA CRÍTICA

RESUMO

O cenário atual mostra uma tendência à terceirização de sistemas espaciais, e para gerenciar o risco presumido, cresce a importância da supervisão do fornecedor ao identificar problemas e não conformidades em estágios iniciais. Além disso, particularmente para o software, a aviação civil realiza algumas atividades de supervisão (informalmente chamadas de auditoria) para fins de certificação, mas os critérios usados para classificar os problemas encontrados não são adequados para avaliar o resultado da auditoria e podem prejudicar a decisão gerencial. O cenário acima, acrescido com a proximidade entre espaço e aviação, cria oportunidade para um mecanismo capaz de capturar o resultado da auditoria que possa ser usado com confiança na decisão gerencial. Este trabalho apresenta métricas para supervisão de fornecedor de software de sistema aeroespacial crítico em segurança (safety), denominados "Métricas Aeroespaciais". O objetivo das Métricas Aeroespaciais é avaliar os resultados da supervisão para prover suporte à decisão gerencial. As métricas são geradas analiticamente usando a técnica GQM (Goal-Question Metric) em combinação com o modelo de erro humano de Reason. Uma pesquisa de opinião é realizada com especialistas em segurança (safety) de software da aviação civil para obter informações adicionais que auxiliem no ajuste e avaliação das métricas. Para avaliação em aeronáutica, as métricas geradas são aplicadas a casos selecionados de auditorias de software da aviação e avaliadas contra o histórico de certificação do software em questão. Para avaliação em astronáutica, é feita uma comparação sistemática de segurança (safety) de software entre espaço e aviação, para identificar ajustes nas métricas e nas atividades de supervisão devido a necessidades específicas do espaço. Como estudo de caso, as atividades ajustadas de supervisão são aplicadas a um projeto espacial chamado QSEE (Qualidade do Software Embarcado em Aplicações Espaciais), e os resultados são submetidos às métricas para avaliação. O trabalho produziu resultados aceitáveis, mostrando que as Métricas Aeroespaciais são viáveis e podem ser aplicadas tanto na aeronáutica como na astronáutica, com poucos ajustes devido a necessidades específicas.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AEB –	Agência Espacial Brasileira
AEH -	Airborne Electronic Hardware
ANAC -	Agência Nacional de Aviação Civil
AR –	Acceptance Review
ARP –	Aerospace Recommended Practice
ASIC -	Application Specific Integrated Circuit
CDR –	Critical Design Review
CMMI –	Capability Maturity Model - Integration
DAL –	Development Assurance Level
DDR –	Detailed Design Review
EASA -	European Aviation Safety Agency
ECSS -	European Cooperation for Space Standardization
ESA -	European Space Agency
FAA -	The United States Federal Aviation Administration
FPGA -	Field Programmable Gate Array
GQM –	Goal-Question Metric
HLR –	High-level Requirement
HSIA -	Hardware-Software Interaction Analysis
IAE –	Instituto de Aeronáutica e Espaço
ICAO –	International Civil Aviation Organization
IEC -	International Electrotechnical Commission
IEEE -	Institute of Electrical and Electronics Engineers
IMA –	Integrated Modular Avionics
INPE –	Instituto Nacional de Pesquisas Espaciais
IT -	Information Technology
IVV -	Independent Verification and Validation
LLR –	Low-level Requirement
MBD -	Model-Based Development
MECB -	Missão Espacial Completa Brasileira
NASA -	National Aeronautics and Space Administration
N/A –	Not Applicable
OOT –	Object Oriented Technology
PDR –	Preliminary Design Review
PLD -	Programmable Logic Device
PNAE –	Programa Nacional de Atividades Espaciais
QR –	Qualification Review
QSEE –	Qualidade do Software Embarcado em Aplicações Espaciais

RB -	Requirements Baseline
RID -	Registro de Item de Desvio
RTCA -	Radio Technical Commission for Aeronautics
S4S –	SPiCE for Space
SAE –	Society of Automotive Engineers
SIL -	Safety Integrity Level
SOI -	Stage Of Involvement
SoS –	System of Systems
SPiCE -	Software Process Capability dEtermination
SQA -	Software Quality Assurance
SRR –	System Requirement Review
STAMP -	Systems-Theoretic Accident Model and Processes
STPA -	Systems Theoretic Process Analysis
SWRR –	Software Requirement Review
TBD –	To Be Defined
TS -	Technical Specification
V&V –	Verification and Validation

CONTENTS

1. INTRODUCTION

This chapter presents the motivation for the thesis in terms of problems and opportunities, the thesis proposition, scope, approach, activities, structure and evaluation criteria, as well as the research paradigm. Problems and or limitations are identified and tagged for convenience (refer to section 1.8).

1.1 – The motivation (problems and opportunities)

In line with the world tendency, the Brazilian Space Agency (AEB) issued the National Program of Space Activities - PNAE (2012) for the period 2012-2021, which included among the priorities:

- Engage industry at all stages of the space project development - from equipment conception and construction to complete space systems;
- Standardization and certification to ensure the quality and safety of space activities in the country.

In such scenario, for managing presumed inherent risk of outsourcing space systems, the oversight (supervision) of supplier increases in importance by identifying project problems and product nonconformities at earlier stages of development, or eventually for compliance verification with certification regulations.

The PNAE also highlighted among its priorities, "*master critical technologies and restricted access technologies, with the industry's participation, and with the expertise and talent in universities and national research institutes*". The embedded software can be considered one of the critical technologies. According to Leveson (2003), software is quickly becoming a major part of and a major concern in space applications. It is also playing an increasing role in space accidents (LEVESON, 2004).

Problem/Limitation-1: *It is presumed an inherent risk on outsourcing software-critical space system, which demands an oversight of software supplier to identifying project problems and product nonconformities at earlier stages of development.*

As result of the problem/limitation-1, three more problems and or limitations were identified and are described in Chapter 3 due to the suitability of the context.

Baufreton et al. (2010) presented an analysis of safety standards and their implementation in certification strategies from different domains (e.g., aviation, industry automation, automotive, nuclear, railway and space), and concluded that aviation and space are very close domains, sharing many concerns, needs and solutions in terms of processes, methods and techniques. Historically, aviation and space are very close to each other. For example, the term "*aerospace*" is widespread, NASA stands for National Aeronautics and Space Administration, and in Brazil the IAE is the Aeronautics and Space Institute.

Particularly for software, the civil aviation performs oversight-like activities (informally called audits) throughout the development for verifying compliance with the certification regulation. The audits are performed in stages with some relation to the software lifecycle phases, and the result is recorded mainly in a list of issues, where each issue is classified according to pre-established criteria. These audit´s result influence the certifier decision for the next steps, which can be from the re-execution of the audit (the worst scenario) to the non-execution of the next audit stage (for the best scenario). Consequently, both the certifier and audited company give importance to the result. However, the criteria used for issue classification are not adequate for reflecting the audit result (refer to section 3.3.5), and may lead to inappropriate interpretations that can adversely affect managerial decisions. Examples of inappropriate use of the result classification for audit evaluation include: company overreacting against substantial number of audit issues even before evaluating the technical severity, or trying to use a small number of audit issues to argue about possible reduction of the certifier level of involvement.

Problem/Limitation-2: *In the civil aviation software audit, the criteria used for issue classification are not adequate for evaluating the audit result and may lead to inappropriate interpretations that can adversely affect managerial decisions.*

The above scenario in civil aviation gives opportunity to creating a mechanism able to capture the audit result, which can be used with confidence for managerial decision. Additionally, considering the current space scenario and the proximity between both domains, the opportunity can be extended to the space domain.

1.2 – The thesis proposition

Considering the scenario presented in section 1.1, this thesis investigates the following proposition:

> ***Considering the presumed inherent risk of systems outsourcing, it is feasible to construct metrics for evaluating oversight's result of software supplier of safety-critical aerospace system, which can be used for managerial decision.***

For the aeronautics, it was chosen the civil aviation approach, and for the astronautics the European Cooperation for Space Standardization (ECSS) standards adopted by the European Space Agency (ESA), due to the following reasons:

- The civil aviation contains harmonized regulations among the various member nations of the International Civil Aviation Organization (ICAO). The National Civil Aviation Agency (ANAC) is the Brazilian organization responsible for the certification of aeronautical products, and has vast material to support the metrics generation;

- The National Institute for Space Research (INPE), where this research has been carried out, is responsible for the development of main Brazilian satellites and has followed the European trend of standardization since its first Space mission.

Basic differences exist between the aviation and the space for software oversights. For instance, one is in the scope of regulator-regulated relationship, whereas the other is for customer-supplier. One is to verify compliance with certification regulation supported by international law/agreement, whereas the other may be required by contract. The thesis proposition is supported by:

a. The use of the consolidated Goal-Question Metric (GQM) technique (BASILI et al., 1994) and the Reason's human error model (REASON, 1990), for constructing systematically and analytically the initial version of the metrics;

b. An examination of vast material gathering 12 years of ANAC practical experience in performing software audits, comprising relevant world aviation system suppliers, for metrics adjustment and evaluation;

c. A software safety systematic comparison between aviation and space to identifying adjustments in space oversight activities and impact in the metrics due to space specific necessities. A bibliographic review on recent works was performed for identification of comparison criteria, limitations and assumptions;

d. Workshops and surveys with software senior specialists from important aviation industries and ANAC, for identifying metrics relevancies, quantitative values, adjustment and evaluation;

e. The use of space project as case study, by applying the oversight activities and recording the non-compliances similarly to software audits in civil aviation certification, in order to obtain representative oversight results for exercising and evaluating the metrics.

Remark: From now on the metrics of this thesis are called "*Aerospace Metrics*" or simply *metrics*, but the latter must be clear in the context to avoid ambiguity.

1.3 – The thesis scope

The following describes the scope of the thesis in terms of "*INs*" (in the scope) and "*OUTs*" (out of the scope):

a. Type of metrics:

IN: metrics for evaluating software supplier oversight result (refer to section 2.3);

OUT: Metrics for evaluating software properties, e.g., lines of code, function point analysis, cyclomatic complexity; Metrics for evaluating the quality of the software development and verification process, e.g., number of errors detected by code inspection, by testing, by requirements review;

b. Type of software:

IN: safety-critical software for aerospace application. It could be applied, with some adjustments, to other domains, e.g., automotive, nuclear, medical, chemical industry;

OUT: software without safety-critical concern, e.g., commercial, financial, entertainment;

c. Applicability:

IN: focuses on aerospace onboard application, e.g., airplanes, helicopters, satellites, launchers;

OUT: a priori, excludes non-embedded aerospace application, e.g., air traffic management, ground segment. However, for the space domain, due to the strong coupling between satellite and earth station, it may be necessary to include some of the ground segment scope;

d. Software assurance (refer to section 1.8 for the rationale):

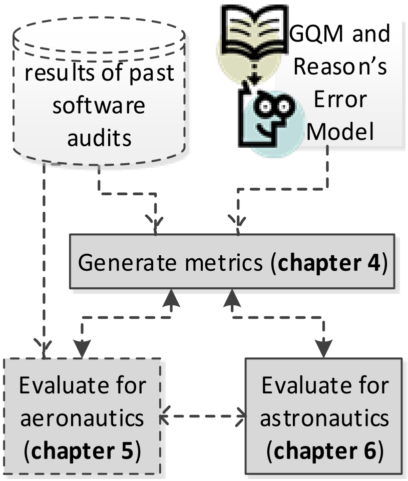
   IN: process-based approach, goal-based (or objective-based) approach;

   OUT: safety-evidence (or safety-directed) approach, wider scope approaches (e.g., STAMP/STPA).

1.4 – The thesis activities

The figure-1.1 illustrates the overall activities performed in this thesis:

Figure-1.1: Overall activities performed in this thesis


results of past software audits
GQM and Reason's Error Model
Generate metrics (chapter 4)
Evaluate for aeronautics (chapter 5)
Evaluate for astronautics (chapter 6)

Generate metrics: The initial version of the metrics is generated by using the GQM technique, and combining with the Reason's human error model. These metrics are further refined by examining the results of past civil aviation software audits, and by performing surveys and workshops with civil aviation software safety senior specialists.

Evaluate for aeronautics: The generated metrics are applied to representative set of past software audits and the resultant measurement is evaluated against the software certification history. Surveys and workshops with senior specialists are also used. Any adjustments in the metrics are forwarded to astronautics for evaluation of impact and or applicability.

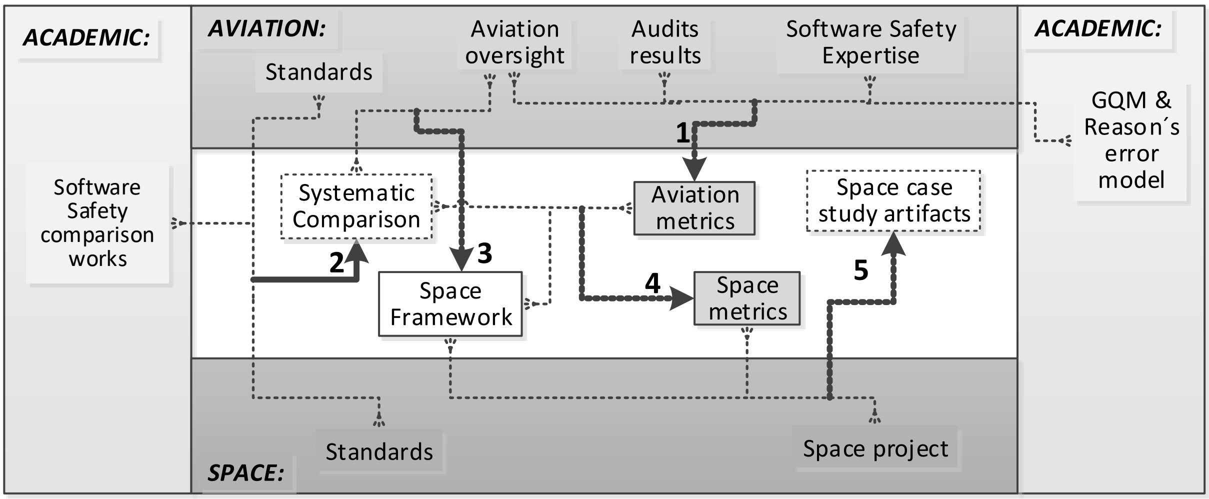
Evaluate for astronautics: First, a systematic comparison between aviation and space is performed for identifying adjustments in oversight activities and impact in the metrics due to space specific necessities. Then, software audits similar with those from the civil aviation are performed in a real space project (i.e., case study), the results are submitted to the metrics and the resultant measurement is evaluated. Any adjustments in the metrics are forwarded to aeronautics for evaluation of impact and or applicability.

1.5 – The thesis approach

For metrics construction and refinement, this thesis relies on the software safety experience of the civil aviation. Hence, metrics evaluation for the space domain is not comprehensive, as it is assumed that enough pedigree has been ensured by the civil aviation experience.

There is strong coupling between the metrics and the related oversight activities, i. e., the metrics may not be applicable to evaluate the results of an oversight whose activities have been performed differently from expected. The figure-1.2 illustrates the thesis approach in terms of artifact´s sequential generation:

Figure-1.2: The thesis approach in terms of artifacts generation



The main sources of knowledge are the aviation, space and academic, represented by the bigger grayish rectangles. Inside the big white rectangle are the main artifacts generated by this work, where *Aviation metrics* and *Space metrics* compose the *Aerospace Metrics*. The generations are represented by numbered arrows in bold (1 to 5), while the inputs for the generations are represented by dotted arrows. Although not shown in the figure, the standards are inputs for all generations. The list below describes the artifact´s generations in step sequence:

- **Step-1**: The aviation metrics are generated by using the GQM technique and Reason´s human error model, and further refined by using past 12 years of ANAC audits results, together with the expertise of software safety senior specialists captured through surveys and workshops. Then, selected software cases are used for the metrics evaluation, again with the expertise of software safety senior specialists (surveys and workshops). The aviation oversight is always used as reference.

- **Step -2**: Once the metrics are generated, refined and evaluated for aviation, a systematic comparison between aviation and space is performed in the software safety scope focusing on a representative set of standards from both domains. The purpose is to identify reuses of aviation oversight and adjustments in space oversight, rather than differences and similarities among standards. In order to have confidence that the systematic comparison provides a representative result, academic works on software safety comparison are evaluated, where assumptions, limitations and comparison criteria are identified.

- **Step -3**: The systematic comparison result generated in step-2 is used for identification of reuses and adjustments in aviation oversight for building the space oversight activities, captured by the Space Framework.

- **Step -4**: The systematic comparison result is also used for identification of adjustments in the aviation metrics for applying in space. As a consequence, the space metrics are built. The Space Framework built in step-3 is used as reference.

- **Step -5**: Audits are performed on INPE´s space project by using the Space Framework. The audits results are submitted to the space metrics for metrics evaluation and adjustment. This is recorded as case study for the metrics evaluation in space domain.

Although metrics and oversights are strongly connected, the main focuses are the metrics, i.e., the main purpose of the thesis activities and artifacts generated is to produce and evaluate the Aerospace Metrics. Nevertheless, the aviation oversight and space oversight play essential roles. The first captures a significant portion of the aviation experience and is used as reference for the metrics generation and evaluation. The latter is used for metrics evaluation in space domain, and is considered an important thesis contribution. An overview of the space oversight framework is provided in appendix-C.

1.6 – Thesis structure

This thesis is organized into the followings:

- Chapter 1 presents the introduction to the thesis (this chapter);
- Chapter 2 presents an overview of the Aerospace Metrics and a bibliographic review on related works;
- Chapter 3 provides a summary of software safety in aerospace domain (i.e., aeronautics and astronautics); more specifically, the related ECSS standards for representing the astronautics domain, and the main software safety standards adopted by the civil aviation for representing the aeronautics domain;
- Chapter 4 describes the metrics generation process: the use of GQM and Reason´s human error model for the initial metrics version, the use of past audits results for refining the metrics, a survey with aviation software safety specialists to obtain quantitative values for the metrics, and the metrics equations;
- Chapter 5 describes the process for the metrics evaluation in aeronautics: the measurements comparison against the results of a survey with aviation software safety specialists, the metrics applied to a representative set of aviation software audits, and the evaluation against the software certification history;
- Chapter 6 describes the process for the metrics evaluation in astronautics: the systematic comparison between aviation and space, the adjustment in space oversight activities and metrics, and the space project used as case study;
- Chapter 7 presents the Aerospace Metrics results in terms of tables and equations, and examples of use in aviation and space;
- Chapter 8 presents the conclusion, including the thesis contribution and future works;
- Appendix A presents a summary of the objectives of DO-178C (2011), the main standard for software safety in civil aviation;
- Appendix B presents the Systematic Comparison Process between aviation and space domains in the software safety scope;
- Appendix C presents an overview of a framework for oversight of software suppliers of safety critical space systems;

- Appendix D presents a survey with software safety specialists from civil aviation;
- Appendix E provides a glossary of terms definition used in this thesis.

1.7 – The research paradigm

The research of this thesis uses the Design Science approach as reference. Simon (1996) in his work "*The Sciences of the Artificial*" has described the differences between the more traditional or natural science, which concerns on explaining the present world as it is (or the nature), and the design-based science, which is driven by finding solutions for "practical world" problems. Van Aken and Romme (2009) define Design Science as research that develops valid general knowledge to solve field problems, and has the following characteristics:

a. Research questions are driven by field problems;
b. Emphasis on solution-oriented knowledge;
c. Justification largely based on pragmatic validity.

Hevner (2007) have analyzed the design science research as an embodiment of three closely related cycles of activities as follow:

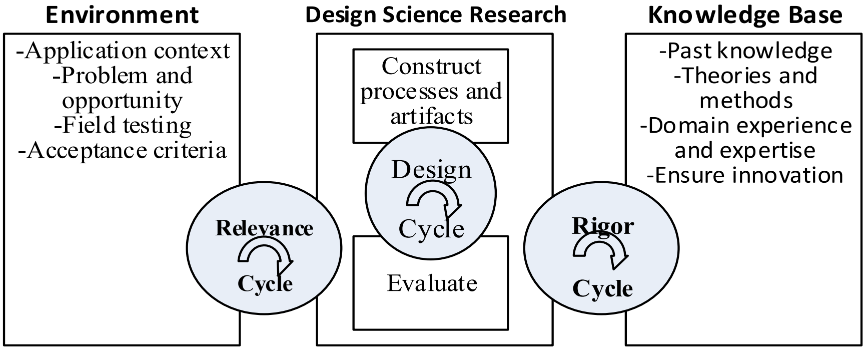
a. *The Relevance Cycle*: it inputs requirements from the contextual environment into the research and introduces the research artifacts into environmental field testing. It initiates Design Science research with an application context by providing the opportunity/problem to be addressed, and defines the acceptance criteria for the ultimate evaluation of the research results.

b. *The Rigor Cycle*: it provides grounding theories and methods along with domain experience and expertise from the foundations knowledge base into the research, and adds the new knowledge generated by research to the growing knowledge base. It provides past knowledge to the research project to ensure its innovation.

c. *The Design Cycle*: it supports a tighter loop of research activity for the construction and evaluation of design artifacts and processes. It is the heart of any Design Science research project, and iterates more rapidly between the construction of an artifact, its evaluation, and subsequent feedback to refine the design further.

The figure-1.3 illustrates the three cycles:

Figure-1.3: Design Science Research Cycles



Source: adapted from Hevner (2007)

The contents of this thesis can be mapped to the three cycles of Design Science as follow:

a. To the Relevance Cycle:

i. The preliminary research summarized in section 1.1 to describe the current scenario and identify problems and opportunities, i.e., the thesis motivation;

ii. The acceptance criteria for the ultimate evaluation of the research result (section 1.5);

iii. The summarized description of the aeronautics and astronautics domains in software safety scope, with identification of potential improvements (chapter 3 and appendix-A);

iv. The application of the metrics to ANAC software audits, for evaluation of the research results in aeronautics (last sections of chapter 5);

v. The application of the metrics to INPE space projects, for evaluation of the research results in astronautics (last sections of chapter 6).

b. To the Rigor Cycle:

i. The introduction of basic concepts and definition of terms to set the theoretical fundamental (chapter 2 and appendix-E);

ii. The bibliographic review on related works to support ensuring the innovation and relevance of the research (chapter 2);

iii. The investigation of recent academic works on software safety comparison (chapter 2), supporting a systematic comparison between aviation and space;

iv. The use of GQM technique and Reason´s human error model (first sections of chapter 4);

v. The thesis contribution (chapter 8);

vi. The academic papers produced (chapter 8).

c. To the Design Cycle:

i. The Aerospace Metrics generation, described in chapter 4;

ii. The metrics evaluation in aeronautics, described in chapter-5;

iii. The metrics evaluation in astronautics, described in chapter-6;

iv. The Aerospace Metrics result provided in chapter-7;

v. A systematic comparison between aviation and space (appendix-B);

vi. A survey with aviation software safety senior specialists (appendix-D);

1.8 – Thesis evaluation criteria

The evaluation focuses on the three cycles of Design Science, where the key points of each cycle are identified in italics and in quotation marks, as follow:

a. *The Relevance Cycle*:

i. Concerning the "*inputs from the contextual environment into the research*", applicable problems and limitations from aerospace are captured and evaluated against coverage by the thesis.

ii. Concerning the "*research artifacts into environmental field testing*", the acceptance criteria for the ultimate evaluation of the research result (i.e., the Aerospace Metrics) should be met along with the practicality of the application.

b. *The Rigor Cycle*:

i. Concerning the "*theories, methods and domain expertise from the foundations knowledge*", they are evaluated for concept solidity.

ii. Concerning the "*past knowledge to ensure research innovation*", an evaluation of the bibliographic review is performed.

iii. Concerning the "*new research knowledge to the knowledge base"*, an evaluation of the thesis contribution is performed.

c. *The Design Cycle*: concerning the *"tighter loop of research activity for the construction and evaluation of design artifacts and processes*", the consistency of the research steps used for the design artifacts construction is evaluated.

Relating to item (a.ii) above, the acceptance criteria for the ultimate evaluation of the Aerospace Metrics by environmental field testing do not aim at finding the best solution, but as stated by Van Aken and Romme (2010), "*it is about changing the actual into the preferred, in which research-informed designing is the core activity*". Concerning the practicality of the application, this thesis is influenced by Weaver (2003), which states that "*it is not possible to completely demonstrate the practical application of the concepts within the timescale of a Doctoral program. However, it is possible to demonstrate the practicality of the concepts to a certain level."*

1.9 – Additional considerations

According to Leveson (2003), the civil aviation accident model is based on component failure as the main cause. Therefore, the safety approach focuses mainly on component reliability, and for software such "reliability" comes down to complying with the Design Assurance Level (DAL) assigned by the safety analysis of the aircraft and its systems, addressing the safety aspects indirectly. Such approach justifies the software assurance scope described in section 1.3. Complying with DAL implies a level of rigor in development and verification process, which for aviation domain is guided by objective-based standards, and for the space domain by process-based standards. Although there are works questioning the current aviation safety approach by asking for safety evidences (McDERMID, 2001; NAIR, 2013), or asking for a wider scope approach based more on system engineering rather than component engineering (LEVESON, 2005), this thesis is built on the current aviation safety approach and does not intend to address any issue beyond that, unless deemed necessary by demand from the space.

2. THE AEROSPACE METRICS OVERVIEW AND RELATED WORKS

2.1 – Overview

This chapter is related to the rigor cycle of the Design Science. It introduces the basic concepts and definition of terms to set the theoretical fundamental of the thesis, as well as an overview of the Aerospace Metrics. It also presents a bibliographic review on related works to support ensuring the relevance and innovation of the research. Additionally, it presents an investigation of recent academic works on software safety comparison to supporting a systematic comparison between aviation and space (refer to appendix-B).

2.2 – Essential concepts for critical software

In order to understand the critical software, it is important to understand the possible causes and consequences of its malfunction. Hence, the following concepts are explained here: error, fault, failure, reliability, accident, hazard, risk, and safety. For fundamentals on software safety, refer to Leveson (1995).

2.2.1 - Error, fault, failure, reliability

In relation to software, for the space domain the ECSS-Q-HB-80-03A (2012) states that a human *mistake* made in specifying requirements, design, or coding may result in a *fault* that would be present latently in a software. This hidden defect under circumstances can manifest as an *error*, a discrepancy between the expected and the actual value, which in turn can generate a *failure*, that is, an unforeseen or unplanned system behavior. For further information on terms used in space domain, refer to ECSS-S-ST-00-1C (2012).

For the civil aviation, an *error* is defined as a mistake in requirements, architecture or code. Such mistake may be a result of deficiencies in development processes or poor adherence to these processes for variety of reasons. These errors, if manifested through executable code, result in *faults*. A fault, therefore, is a manifestation of the error in the software through the executable code. If the fault causes the software to not comply with the requirements, there is a software *failure*. The DO-178C defines failure as the inability of a system or system component to perform a required function within specified limits. A failure is produced by a fault. Software without faults is totally reliable as it will always behave as specified (assuming its specification is correct and complete). As one can see, there are some differences between space and aviation regarding to those basic definitions. For convenience, this thesis adopted the civil aviation definition.

Software *reliability* is the likelihood that software will behave as specified by the requirements over a given period. Unlike hardware, the metrics used for software numerical reliability are still immature and controversial. In aerospace domain, the numerical reliability of the software is not considered, i.e., the expression software reliability is more associated with the quality of its development and verification processes, as well as the level of adherence to them.

2.2.2 - Accident, hazard, risk, safety

A combination of failures can generate an unintended event with harmful consequences. *Accident* is defined as an unintentional event or sequence of events that causes death, injury, property damage or environmental damage. However, it does not make sense to define critical software using the term accident, as the computer is inherently safe and cannot, by itself, cause deaths, injuries, or property damage. It is then defined the term *hazard*, which is a situation that can lead to an accident. The state of the system that is part of the hazard is called a *hazardous state*. Critical software would then be the one whose failure can contribute to bringing the system into a hazardous state.

For Storey (1996), the product of the probability of existence of the hazard by the magnitude of its consequences is called *risk*. *Safety* means a property of the system that will not endanger human lives or the environment. A safety system implies a sufficiently low and acceptable risk. It does not necessarily mean absence of hazard, much less absence of failure. Therefore, although high reliability has a positive contribution to safety, the first does not necessarily imply the second.

2.2.3 - The safety-critical software

According to NASA (1997), safety-critical software is the one that:

> (1) Exercises direct command and control over the condition or state of hardware components; and, if not performed, performed out-of-sequence, or performed incorrectly could result in improper control functions (or lack of control functions required for proper system operation), which could cause a hazard or allow a hazardous state to exist.
>
> (2) Monitors the state of hardware components; and, if not performed, performed out-of-

sequence, or performed incorrectly could provide data that results in erroneous decisions by human operators or companion systems that could cause a hazard or allow a hazardous state to exist.

(3) Exercises direct command and control over the condition or state of hardware components; and, if performed inadvertently, out-of-sequence, or if not performed, could, in conjunction with other human, hardware, or environmental failure, cause a hazard or allow a hazardous state to exist.

Software alone cannot be unsafe, but the way it interacts with hardware and other systems can cause hazardous states. The software will never "fail" like the hardware because it does not suffer from aging or wear-out or something similar that is typical of the hardware. The software will fail if it generates an unintended output. Software errors can be induced via logic or requirements errors. Therefore, theoretically, software failures can be eliminated through the degree of control to avoid errors of logic and requirements, since in practice it is impossible to guarantee that a software is totally free from failures, since the combination of conditions and variables can be too large to an extent that exhaustive tests is impossible. Software engineering efforts can only increase confidence that the software will behave as specified.

2.3 – The Aerospace Metrics overview

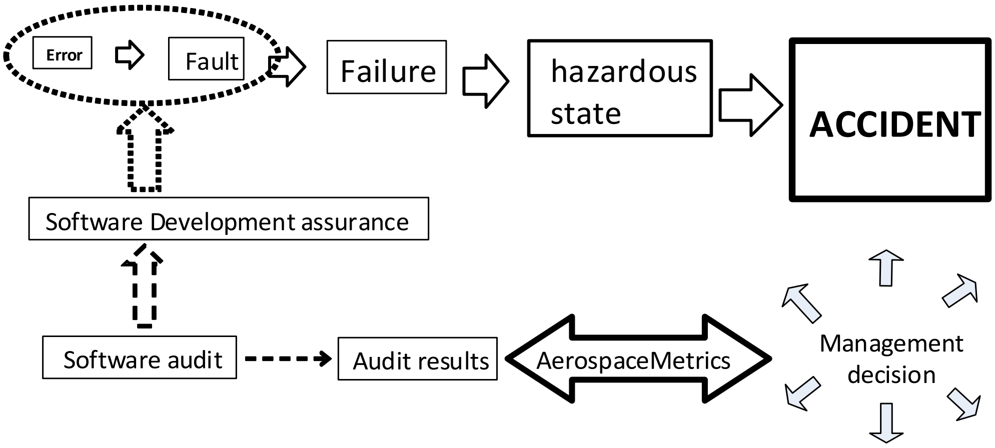
According to Pressman (2015), a key element of any engineering process is measurement. But unlike other engineering disciplines, software engineering is not grounded in the basic quantitative laws of physics. Software metrics refers to a broad range of measurements, and can be related to direct measures (e.g., execution speed, number of lines of code - LOC), or indirect measures (e.g., quality, complexity). Indirect measures demand some analysis prior to obtaining the values, sometimes with the construction of additional artifact. One example is the cyclomatic complexity (McCABE, 1976), the most popular metric for measuring software complexity, and uses the flow graph as input. Another example is a function-oriented metric called function point (ALBRECHT, 1979), which uses direct software measures combined with the qualitative complexity assessment to calculate the final value.

For Pressman, metrics can also be related to a product, process or project. *Product metrics* measure the product attributes for indirect indication of the efficacy of development and verification processes, and the overall software quality. *Process metrics* directly measure the efficacy of those processes. They are collected over lengthy periods of time to support process improvement. *Project metrics* are used by a project manager and a software team to adapt project work flow and technical activities, aiming at avoiding delays and mitigating potential problems and risks. In fact, there are overlaps among those three types. For example, a set of metrics used for project domain can also be applied for process improvement, or a set of product metrics can be used for management decision as a project metrics.

The Aerospace Metrics of this thesis are more related to indirect measures. They are basically built upon a set of tables describing cases of audit issues and related severities in numeric values, and the measurement itself is calculated by using the values extracted from those tables and the numeric relevance of each table. Considering the three types of metrics described by Pressman, the Aerospace Metrics of this thesis cannot be classified as product metrics because the audit issues are not result of any direct assessment of the software product attributes. And cannot be classified as process metrics either, as the audit issues cannot be used to directly measure the efficacy of software processes. Moreover, the audit issues lack representativeness for process metrics purpose due to the relative small number of samplings collected during the audit if compared to the large scope of processes covered. The Aerospace Metrics can be classified as project metrics because their main purpose is to provide support for management decision as stated in section 1.1.

The figure 2.1 illustrates the Aerospace Metrics in the context of the essential concepts of critical software described in section 2.2:

Figure-2.1: The Aerospace Metrics in the context of software safety



As described in section 2.2, the sequence "*error*" to "*fault*" to "*failure*" to "*hazardous state*" may lead to an "*accident*". The software safety approach in aerospace domain basically concentrates on "*software development assurance*", which focuses on eliminating the occurrence of errors and consequently of faults, by application of best practices of software engineering captured in the aerospace standards. The processes used can be subjected to "*software audits*" for assessment of level of compliance with the applicable aerospace standards. The list of issues identified and recorded during the audit comprises the "*audit results*", which are submitted to the "*Aerospace Metrics*" for the calculation of the values to be used as support for management decision.

2.4 - Bibliographic review

This section presents the bibliographic review of works related to the thesis and is divided in three parts: subsection 2.4.1 presents the works that were used as main references for this thesis, and influenced the approach adopted; subsection 2.4.2 presents works related to the main subjects of this thesis for supporting the relevance and innovation; and subsection 2.4.3 presents works on software safety comparison, in order to support specifying the *Systematic Comparison Process* (refer to appendix-B).

Note: the standards related to software safety in aerospace domain are presented in chapter 3.

2.4.1 - Works used as main references

There are four works used as main references for the thesis. The first two influenced in using the GQM for initial metrics generation; the third influenced in using the Reason´s human error model for a metric related to root cause; and the fourth influenced in applying the framework concept for specifying the space oversight activities.

Cruickshank et al. (2009) described a validation metric framework applied to safety-critical software-intensive systems. The framework was built using two well-known software engineering tools: the GQM and Goal Structuring Notation (GSN). It was applied to a fictitious surface-to-air missile system, and historical metric data from successfully finalized systems was used as reference for comparison. The case study demonstrated that the metrics cannot determine validity of the safety requirements, but the framework can provide early warnings of the invalidity of software safety requirements. To determine if the requirements are valid, further investigation is necessary. Michael et al. (2010) extended the framework to safety requirements validation of system of systems (SoS). Emergent hazards are an SoS concern, and the paper classified the emergent hazards into three categories, and presented a new process for analyzing one type of emergent hazard known as interface hazard. The way this thesis applies the GQM is influenced by Cruickshank and Michael works, but basic differences exist. Cruickshank and Michael apply GQM to generate metrics that can help evaluating software safety requirements for validation purpose. This thesis applies GQM to generate an initial version of the metrics, which are further refined by other means, and are used to evaluating oversight results of software suppliers of safety-critical systems. The metrics can be seen as indirectly evaluating the development and verification of the software through evaluation of the oversight results. Different from Cruickshank and Michael, the requirements validation is out of the thesis scope, and the indirect evaluation applies to the full set of requirements allocated to the software, and not only the software safety requirements.

Howden (2011) proposed and error-based approach to software certification. A software interpretation of the Reason´s human error model was developed, and the root causes of failures were viewed as errors made during software development phases. The error-based approach was applied to a collection of 38 known defects ranging from student projects to industrial products, and for each of the defects the effectiveness of twelve well-known methods were evaluated and compared to the error-based approach. The result showed the error-based approach with the highest performance, being effective for 35 out of the 38 known defects, followed by a combination of the other methods at 29, Bounded Exhaustive Testing (BET) at 19, black-box at 18, and the rest at lower than 12. The author concluded that the proposed approach can provide a stronger level of certification than one based on a single method. This thesis uses the Reason´s human error model influenced by Howden´s work, but in a different way. In Howden, the Reason´s human error model is applied to software development to identify and classify error-prone construction types during detailed design and coding, which are further captured in checklists for supporting design/coding reviews. Although the paper mentioned the use in software certification, the purpose focused more on finding errors inserted during design and coding. In this thesis, as part of metrics generation, the Reason´s model supports identifying and classifying root-cause of software audit issues. One can say that it is indirectly applied to the activities required for compliance with the applicable aerospace software safety standards.

Cleland et al. (2002) partially described a study funded by ESA aiming at defining a framework for the software aspects of the safety certification of a space system. Three previous ESA projects were studied and their approaches for certification were evaluated. Research in other domains was also performed to identifying best practices, techniques and methods which are relevant to space software certification. As a result, an overview of the proposed framework was presented: a goal-based approach (i.e., non-prescriptive) with tailoring of ECSS standards under certification requirements demand, and safety cases plus development process activities for providing certification evidences. A certification model, actors and roles were also presented. The paper stated that by the end of April 2003 it would have concluded a certification and accreditation framework for software, but no further related publication could be found. Similar to Cleland, this thesis uses the concept of framework with tailoring of ECSS standards. However, in this thesis the framework is applicable to software supplier oversight (instead of software certification), the ECSS standards tailoring is driven mainly by the comparison between aviation and space (instead of certification requirements), and the purpose is to obtain evidences for compliance to ECSS requirements defined by customer-supplier contract (instead of certification evidences). Besides, the framework described by Cleland has broader scope including roles as regulators, independent safety assessors, management, accreditation bodies, suppliers and operators, whereas the framework scope of this thesis is restricted to customer-supplier relationship.

2.4.2 - Related works for supporting the thesis relevance and innovation

This subsection is divided in two parts: first, it provides a summary of works related to software metrics, which is the thesis main subject, for supporting the thesis innovation. Then, it provides an overview of works closely related to software metrics for showing the thesis relevance, which are: (1) software oversight (or supervision) along with software outsourcing due to strong connection between the metrics and the oversight activities; and (2) compliance demonstration approaches including those for certification purpose, as the metrics and oversight activities are used for compliance verification with applicable requirements.

As discussed in section 2.3, the Aerospace Metrics are considered project metrics; hence, four works are presented: the first one for brief evaluation of 25 years of software metrics, and the other three are software project metrics in space, aviation and telecom domains, respectively. Works on product and process metrics are mentioned briefly.

Pfleeger (2008) presented a brief evaluation of the past 25 years of software metrics, and commented that "it started to be less about the right metrics and more about the right kind of evaluation", i.e., a tendency to value the metrics focusing on their end use and not in the metrics themselves. For Pfleeger, considerable progress has been achieved but there are obstacles to overcome, and mention among others, the use of heuristics to help in understanding when some metrics are good enough, though not perfect. There is no need to always measure everything with high precision. The approach adopted for the Aerospace Metrics of this thesis is in line with Pfleeger. The purpose of the metrics and the rationale for their necessity is clear. The numeric values assigned are strongly based on qualitative judgment, with approximation commensurate to the end use of the metrics. The thesis does not claim perfect metrics, but good enough to address the problem/necessity identified in the current scenario. Considering that Pfleeger evaluates software metrics up to 2008, it was decided to perform this bibliographic review focuses on works since then for covering the gap between Pfleeger and the present date.

Layman et al. (2011) applied the Technical and Process Risk Measurement methodology to perform software safety risk in NASA's Constellation spaceflight program. They collected metrics from 154 hazard reports and found that: 49-70% of hazardous conditions could be caused by software or software was involved in the prevention; 12-17% of the hazard causes involved software; and 23-29% of all causes had a software control. The work concluded that simply defining a development process is not sufficient to identify safety risk. Management, measurement, and feedback of the process being used are important to ensuring process adherence, resulting in lower risk of safety problems. Adherence cannot guarantee a quality product, but non-adherence increases the risk of failure. The result presented by Layman reinforces the relevance of the Aerospace Metrics of this thesis. Like Layman, the Aerospace Metrics aim to be a tool for management decision and are applied to audit results which are mainly assessments of process adherence led by software contribution to hazardous conditions, and the ultimate purpose is to lowering risk of safety problems. But unlike Layman which uses as input the artifacts produced at system development level (i.e., hazard reports), this thesis concentrates on artifacts produced at software level. Moreover, Layman methodology can be used as tool for supporting the planning of the supplier oversight activities, whereas the metrics of this thesis are tools for supporting the oversight activities themselves.

Dodd and Habli (2012) proposed a statistical method for assessing the readiness of airborne software projects for audits in civil aviation certification. The method used 15 metrics refined by the GQM, and was further evaluated by case study comprising 58 software from 9 different projects. The authors concluded that the method can help the certifier and the audited company to gain confidence in the software certification readiness and predict the likely outcome of the audits. The work has the following similarities with the thesis: the scope is airborne software safety in the civil aviation focusing on metrics for software audits support; and to building the method, the work used GQM together with lifecycle data from past projects, and expertise of experienced auditors. However, the following differences exist: the work focused on the complete software lifecycle data as input (instead of audit results), the obtained measurement reflected the software readiness for audit (instead of reflecting the audit result), and although mentions aerospace, the scope was restricted to civil aviation. Moreover, the method used problem reports (PR) as a mean for measurements, and weighted more those PRs with adverse safety or functional impact, relegating to minor relevance the PRs related to process issues. Such approach seems incoherent with the purpose of the work, as the main objective of software audits is to assess the quality of (and adherence to) the process. Differently, the Aerospace Metrics of the thesis aim to measure the relevance of process issues recorded by the software audit. Another difference is that by needing the complete lifecycle data as input, the method presented by the work may not be practicable for use by the certifier due to independence or confidentiality issue, and sometimes not even by the customer due to restrictions imposed per customer-supplier contract. Differently, the input for the Aerospace Metrics of the thesis are the audit issues, which are recorded by the auditor whose role is usually performed by the certifier or the customer; therefore, no independence or confidentiality issue exist.

Asthana (2009) described a quantitative software readiness criteria for product delivery, by considering parameters from all aspects of software development life cycle, e.g., requirements, project management, development testing, audit assessment, stability and reliability, and technical documentation. The method organized existing data into a simple metric that is applicable across products and releases. As case study, the method was used with real data for several software from the telecom industry. According to Asthana, the method can be a supporting tool for objective and effective decision-making at management level to ensuring timely product delivery. Although Asthana work has a broad scope, it can be seeing as an organizer of several metrics to capturing into a single metric for easy visualization and evaluation, and assumes that every software aspect is properly measured by adequate metrics. Analyzing the work in the context of this thesis, the Aerospace Metrics could also be one of those metrics to be considered by Asthana method. However, Asthana is not clear about the stakeholders involved, but it seems more applicable to the supplier scope, rather than the customer. As such, the Aerospace Metrics should be used internally by the supplier as part of Software Quality Assurance (SQA) audits and assessments, instead of supplier oversight by customer.

There are plenty of works addressing software metrics, but most of them fall into the product or process types, which are not the focus of this thesis. Just to mention some: the use of GQM technique, extended with Data Warehousing model design concepts to extract a set of metrics for measuring the gains of software reuse is proposed by Vieira et al. (2011); selection of appropriate software metric for verification of system testing models of safety-critical systems is presented by Spendla et al. (2013); Sharma and Kushwaha (2013) proposed a test metric for the estimation of software testing effort at very early stage of development (i.e., using the software requirement specification). As far as the bibliographic review of this thesis has reached, no works with same characteristics of the proposed Aerospace Metrics were found. The works used as main references are believed to be the most similar to the Aerospace Metrics of this thesis. Nevertheless, significant differences exist and were already highlighted in subsection 2.4.1. As for the three works described above, which fall into project metric type, it is possible to identify some similarities with the Aerospace Metric, but they are essentially different. Therefore, it can be stated that the bibliographic review provided enough confidence in the innovation of the proposed Aerospace Metrics of this thesis.

To support the relevance of the thesis, works on subjects closely related to software metrics (i.e., software oversight, outsourcing, and compliance demonstration approaches) were reviewed and some of them are herein briefly described.

The term "*oversight*" as used in this thesis, although often used in aviation industry, is not popular in academic works. Terms like "supervision" or "audit", though not totally equivalent, are found more often, and some works are mentioned here: Axelrod (2011) presented a supply-chain integrity model comprising audit reviews, monitoring of critical processes, and testing of individual components along the lifecycle; Boer (2007) applied a technique called Latent Semantic Analysis to guide the auditors through the documentation to the software architectural knowledge needed; and Kumar (2010) described the regulatory review and audit process required by the Atomic Energy Regulatory Board (AERB) for assessing the qualitative reliability of software based nuclear instrumentation, as well as a case study of AERB audit on

V&V for software based safety related systems used in an Indian plant.

Note: Description of oversight-related activities in space and aviation domains is provided in chapter 3.

As already described in section 1.1, the PNAE included among the priorities to increase industry outsourcing from equipment to complete space systems. Some works on outsourcing are herein mentioned: Sharma (2013) analyzed the Indian IT outsourcing industry, and concludes that the future depends on availability of quality manpower, capability to move up the value (i.e., research, innovation and product development), and the growth in domestic IT consumption; Peterson (2011) reported on the problems of the Boeing 787 Dreamliner massive outsourcing experience, which faced years of delay and paid out hundreds of millions of dollars for late delivery penalties; Tokgoz and Erdogan (2016) collected data by semi-structured interviews with aviation company´s IT managers to investigate how IT management differ in the aviation industry, and the reasons for aviation organizations to choose outsourcing; Yajing and Deying (2011) proposed an IT outsourcing risk analysis based on critical distance factors, i.e., information distance, spatial distance and the knowledge distance.

There are several works on software compliance demonstration and certification. Some works addressed the safety evidence issue (WALAWEGE et al., 2010); others proposed a product-based certification in lieu of or in addition to the more widespread process-based certification (RODRIGUEZ, 2012); or a hybrid approach (i.e., both goal-based and prescriptive) for software safety certification (STENSRUD et al., 2011). Some works addressed specific issues like software component certification (CARVALHO et al., 2009); or use of fault injection for certification credits (COTRONEO, 2013); Falessi et al. (2012) presented a model-based tool-supported approach for assisting in compliance demonstration with safety standards for certification purpose; Domis et al. (2009) developed a method that introduces the Safety Concept Trees as a backbone to achieve vertical and horizontal traceability between all safety information, facilitating compliance with safety standards as needed for certification purposes.

Several works were reviewed on software metrics, software outsourcing and oversight, and software compliance demonstration approach and certification. Due to constraints, only some of them were shortly described here. Nevertheless, considering that those are the subjects closely related to the thesis theme, it can be stated that the bibliographic review provided enough evidence of the thesis relevance.

2.4.3 - Evaluating works on software safety comparison

As part of metrics evaluation for space, a *Systematic Comparison Process* between aviation and space domains in the software safety scope was specified and is described in appendix-B. At first, works on software safety comparison were evaluated to identifying assumptions, limitations and comparison criteria, aiming at using them in the systematic comparison process for producing a representative result. For the evaluation, it was prioritized works from well-known publishers (e.g., IEEE, ACM, Elsevier, Springer), but the following were excluded: books, works about standards not widely known, advertising-like material, preliminary works, and works dedicated to very specific subjects like fault injection and unmanned aircrafts. Initially, 54 papers were identified, but only 13 were selected due to relevant content to use in this investigation. This subsection provides a brief description of those works followed by a summary of assumptions, limitations and comparison criteria identified.

The CG2E (*Club des Grandes Entreprises de l'Embarqué*) comprises more than twenty innovative companies involved in the development of critical embedded systems covering six important domains: civil aviation, automotive, space, industrial automation, nuclear plants and railway. The CG2E objectives are to improve its members' capabilities to meet the major challenges of the development of embedded systems, particularly the software-intensive safety-critical embedded systems, and the main results are summarized in the four papers that follow. Baufreton et al. (2010) described a general comparison of safety standards considering the orientation of standards towards integrated or external safety, towards the prescription of objectives vs. means, their notions of severity, criticality and assurance levels, their focus on fault tolerance or fault prevention, on probabilistic vs. deterministic safety assessment methods, and the notion of safety case. Concluding, it mentioned that the aviation and space are very close domains, sharing many concerns, needs and solutions in terms of processes, methods and techniques. Blanquart et al. (2012) described a more in-depth comparison focusing in criticality categories (e.g., DAL, SIL) across safety standards, and concluded that the definition and allocation of safety categories in those application domains are not fundamentally different, and could be seen as various instances of a single consistent scheme. All domains share the same fundamental basis where the categories represent the risks associated to the end effects of the potential failures of the considered system. Machrouh et al. (2012) presented an analysis of the impact of the criticality categories on the system activities in the concerned application domains. The most decisive influence is on the processes which are recommended to establish the system safety requirements. Although differences exist among safety standards (e.g., some are domain-specific while others are more generic), standards generally agree on a common framework which combines hazard assessment and risk analysis techniques. Ledinot et al. (2012) also provided similar analysis, but focusing on software development assurance impact. The paper stated that the criticality categories have influenced six aspects of the software development assurance, with significant difference among some industrial domains. A table compared the level of influence among the domains, where one can notice that space is closer to both aviation and nuclear domains.

Some works used a set of criteria to evaluate groups of standards. Wong et al. (2014) developed a set of 15 criteria to evaluate software safety standards in terms of usage, strengths and limitations, and applied it in five popular safety standards including aviation and space. Additionally, some software-related accidents were reviewed and potential enhancements were discussed based on comments from users of these standards. Results showed that there is no standard which is superior to others on all criteria. Ceccarelli and Silva (2013) investigated the commonalities and differences between relevant aerospace standards through a qualitative comparison of 11 criteria called key arguments. The results showed major commonalities between the standards, but the existence of several specificities complicates the definition of a common development process. Esposito et al. (2011) analyzed 12 well-known safety standards from six different domains (including aviation and space) by applying a fixed set of nine criteria called metrics. The intent was to point out communal areas of interest and features in which the standards diverge. The paper concluded that a super-standard could be artificially created to collect all the similarities and divergences, but unfortunately a very costly and time-consuming complex document.

Some works compared two specific standards or domains. Gerlach et al. (2011) presented ongoing work on safety standards comparability between automotive and avionics, more specifically, an attempt to high-level mapping of processes and artifacts between ISO 26262 (2011) and DO-178B. A case study was also presented. The work argued that a mapping between both standards exists, and ISO 26262 development can make use of the artifacts and processes defined in DO-178B, but considering the addition of some processes. Jacklin (2012) presented an overview and comparison of the standards used for the development of safety-critical airborne and ground-based software (i.e., RTCA/DO-178C and DO-278A (2011)), and related documents, i.e., tool qualification, technology-specific supplements, clarification document. The objective was to help those not familiar with the new documents to obtain the scope of the information contained within. Youn and Yi (2014) presented a comparison between software and hardware certification of safety-critical avionic systems, by reviewing and summarizing DO-178B and DO-254 in terms of objectives, independence, design assurance, life cycle processes, tool qualification, etc. The paper mentioned that the ambiguity and flexibility of the guidelines result in various interpretations and implementations, and excessive costs in software development and hardware design are often attributed to insufficient understanding of these standards.

Some works performed comparison by focusing on a single criterion or concern: Regan et al. (2012) presented a literature review on traceability together with eight case studies in real organizations, focusing on identifying motivations and benefits to implement traceability for both generic and safety-critical domain. The paper concluded that implementing and using traceability support gain in productivity, maintenance and quality for both domains. But particularly for the safety-critical domain, 'regulation' and 'safety case' are two extremely important motivators. Daniels (2011) presented process differences in creating standards, by comparing the process that has created the DO-178B, a standard widely adopted in aviation, with the process that created the Defense Standard 00-55 (1997), a standard not accepted by industry and declared obsolescent. The comparison is supported by the author´s experience of participating in the committee that created the DO-178C, and aims to encourage the readers to think about how safety-related standards are best developed. Wong et al. (2011) presented an evaluation of five software safety standards in terms of cost effectiveness, and several projects were examined covering both high-cost and cost-effective cases. The paper concluded that no single factor can be identified as 'the' contributing cause to high-cost. Various company factors as well as insufficient guidance in some standards, all can contribute to project difficulties. Conversely, some examined projects showed that it is possible to build a cost-effective safe software through effective planning and engineering practices.

Considering the potential deficiencies that should be covered by the *Systematic Comparison Process,* the following limitations were identified:

- Different scopes: equivalence in standards scope is not considered in the comparison results analysis (e.g., RTCA/DO-178B (1992) is for airborne software, whereas DoD-MIL-STD-882D (2000) is for system safety);

- Lack of integral analysis: some standards should have been analyzed as a group instead of individually (e.g., RTCA/DO-178C together with SAE-ARP4754A (2010));
- Unclear comparison: differences and similarities among standards are not explicitly identified, but rather the level of adherence to the criteria used;
- Limited point of view: when comparing two standards, there is a tendency to use as reference the characteristics of one standard only, probably due to the author`s main expertise, and the result tend to be one-side standpoint;
- Lack of completeness: use of a reduced set of criteria and in some cases a single criterion, not covering enough aspects of the standards.

Regarding the criteria to be used by the *Systematic Comparison Process*, a total of 184 comparison criteria were identified and grouped by similarities and refined by removing repetitions, overlaps and subsets. Then, they were classified according to the subject resulting a final list with 32 criteria. The list description is provided in appendix-B, table-B.1.

Some works provided a description of the basic approach of aviation and space domains or related standards, which can be used as assumption for specifying the *Systematic Comparison Process* (e.g., both are process-based with activities commensurate with the assurance levels). Further description on assumptions is provided in appendix-B.

2.5– Summary of chapter 2

This chapter introduced the basic concepts and definition of terms, provided and overview of the Aerospace Metrics, a bibliographic review on related works, and an investigation of recent academic works on software safety comparison. The Aerospace Metrics of this thesis can be classified as project metrics with indirect measures. The bibliographic review provided enough confidence in the innovation of the proposed Aerospace Metrics, as well as the relevance of the thesis subject. A summary is provided in table-2.1:

Table-2.1: Summary of bibliographic review

Review type	Main purpose	#Works recorded	Comment	Conclusion
On main works	Use as reference for thesis approach	4	Influenced the use of GQM, Reason´s model and framework for oversight	Have influenced, but differ from thesis use
On metrics	Ensure thesis innovation	7	No works with same characteristics of Aerospace Metrics found	Enough confidence of thesis innovation
On thesis related subjects	Show thesis relevance	14	Works reviewed on Sw metrics, outsourcing and oversight, and compliance demonstration approach and certification.	Enough evidence of thesis relevance
On software safety comparison	Support comparison between aviation and space	13	Works evaluated to identify assumptions, limitations and comparison criteria, for Systematic Comparison process	Helped Systematic Comparison to produce representative results

3. SOFTWARE SAFETY IN AEROSPACE DOMAIN

3.1 – Overview

This chapter is related to the relevance cycle of the Design Science. The summary provided here focuses on information related to the thesis and does not intend to cover all aspects of software safety. The aerospace is composed of aeronautics and astronautics, and emphasis is given to standards and best practices from the civil aviation for aeronautics and the ECSS standards for the astronautics.

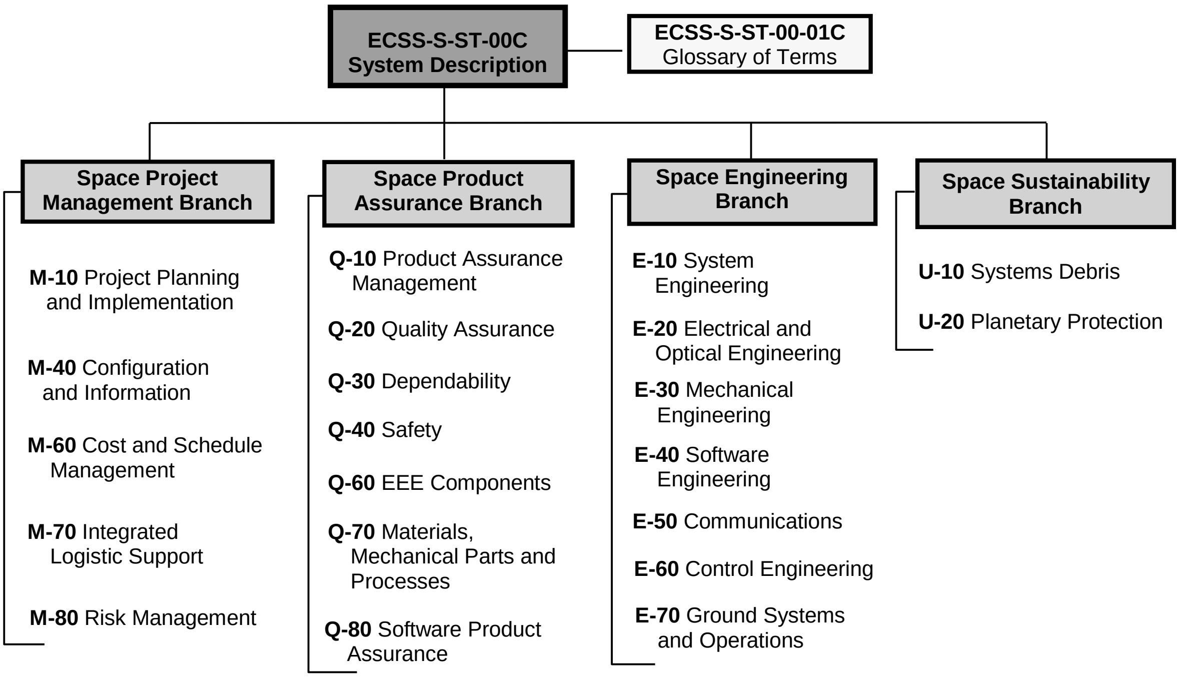
3.2 – Software safety in space domain

The main standards and practices are concentrated in two major centers: USA and Europe. This thesis focuses more on the standards and practices adopted by European agencies. The INPE on its first space mission (MECB) has consulted ESA, and has since followed the European trend. Additionally, the need for harmonization of standards and practices between the space agencies of different European countries makes the material well-organized and widely available. Nevertheless, whenever necessary, complementary material was consulted for support: NASA-GB-8719 (2004), NASA-STD-8719 (2004), Owens et al. (2007), Hill and Tilley (2010), Stetson et al. (2012), Hill and Victor (2008), Lutz and Hine (2008), Schumann (2007), Havelund (2011), Mattiello et al. (2006).

3.2.1 - ECSS standards related to software safety

According to the preface of the ECSS standards, "*ECSS is a cooperative effort of the European Space Agency, national space agencies and European industry associations for the purpose of developing and maintaining common standards*". ECSS standards are organized into four groups: management, engineering, product assurance and sustainability, and each group is organized into subjects (total of 21), covering various aspects of the space domain. The standards undergo continuous revisions, and in April 2014 there were about 118 active standards, plus additional material (handbooks) for standards clarification or detailing specific issues. The figure-3.1 shows the organization of the ECSS standards:

Figure-3.1: Organization of the ECSS standards in groups and disciplines


ECSS-S-ST-00C
System Description
ECSS-S-ST-00-01C
Glossary of Terms
Space Project Management Branch
Space Product Assurance Branch
Space Engineering Branch
Space Sustainability Branch
M-10 Project Planning and Implementation
M-40 Configuration and Information
M-60 Cost and Schedule Management
M-70 Integrated Logistic Support
M-80 Risk Management
Q-10 Product Assurance Management
Q-20 Quality Assurance
Q-30 Dependability
Q-40 Safety
Q-60 EEE Components
Q-70 Materials, Mechanical Parts and Processes
Q-80 Software Product Assurance
E-10 System Engineering
E-20 Electrical and Optical Engineering
E-30 Mechanical Engineering
E-40 Software Engineering
E-50 Communications
E-60 Control Engineering
E-70 Ground Systems and Operations
U-10 Systems Debris
U-20 Planetary Protection

Source*:* adapted and updated from ECSS-P-00A (2000)

ECSS standards are organized by requirements and focus on the customer-supplier relationship, which lets one specifies by contract which requirements of the standards should be mandatory. The safety standard ECSS-Q-ST-40C (2009) contains a matrix that maps applicability of requirements to the different space systems: satellite, unmanned systems, manned systems, and launch vehicle. Likewise, the software standards ECSS-E-ST-40C (2009) and ECSS-Q-ST-80C (2009) contain matrixes that maps applicability of requirements to different software criticality. For any interface with ground station, ECSS-E-ST-70C (2008) provides guidance on ground systems and operations, and ECSS-Q-HB-80-04A (2011) presents a software metrication program definition and implementation.

For further information, Feldt et al. (2010) presented results from two industrial case studies of companies in the European space industry that are following ECSS standards in various V&V activities; Mattiello et al. (2005) presented a comparative study between PMBoK/DoD and ECSS Management Process for software acquisition; and Martin et al. (2013) presented a methodology that relies on MBD and formal verification, with integrated tool support in compliance to the phases of ECSS-E-ST-40C**.**

3.2.2 - Dependability and safety of software: characteristics

For the ECSS, although dependability includes attributes of Reliability, Availability, Maintainability and Safety (*RAMS*), the term “safety” is used separately and “dependability” refers to reliability, availability and maintainability. The ECSS differentiates safety and dependability. The safety classification is restricted to two higher severities, while dependability applies to all levels of severity, according to table-3.1. For information on space dependability, Lahoz et al. (2012) presented a quality factors approach to dependability attributes for space computer systems.

Table-3.1: Severity of failure modes consequences

Severity	Level	Dependability	Safety
Catastrophic	1	Failure propagation	. Loss of life, life-threatening or permanently disabling injury or occupational illness; . Loss of system; . Loss of an interfacing manned flight system; . Loss of launch site facilities; . Severe detrimental environmental effects.
Critical	2	Loss of mission	. Temporarily disabling but not life-threatening injury, or temporary occupational illness; . Major damage to interfacing flight system; . Major damage to ground facilities; . Major damage to public or private property; . Major detrimental environmental effects.
Major	3	Major mission degradation	N/A
Minor or Negligible	4	Minor mission degradation or any other effect	N/A
NOTE: When several categories can be applied to the system or system component, the highest severity takes priority			

The software can be classified into four categories depending on the functions that implements, whose failure can lead to one of those events and related severity classified above. Table-3.2 shows the four software classifications, and will be assigned the category based on the function associated with the highest severity:

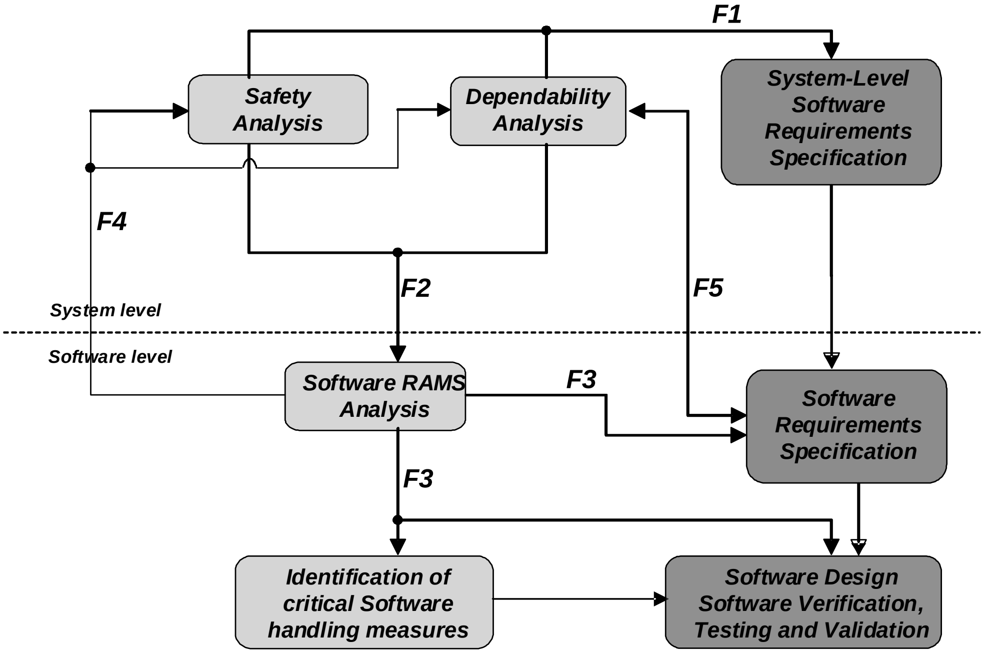
Table-3.2: Software criticality categories

Category	Definition
A	Software that if not executed, or if not correctly executed, or whose anomalous behavior can cause or contribute to a system failure resulting in *Catastrophic consequences*
B	Software that if not executed, or if not correctly executed, or whose anomalous behavior can cause or contribute to a system failure resulting in *Critical consequences*
C	Software that if not executed, or if not correctly executed, or whose anomalous behavior can cause or contribute to a system failure resulting in *Major consequences*
D	Software that if not executed, or if not correctly executed, or whose anomalous behavior can cause or contribute to a system failure resulting in *Minor or Negligible consequences*

The sets of requirements of standards ECSS-Q-ST-80C and ECSS-E-ST-40C vary according to the category of software, and for category "A" they are all applicable. The categories of software and the requirements of the standards are mapped, indicating whether the requirement is applicable, not applicable, or applicable under certain conditions.

Figure-3.2 provides a complete overview of dependability and safety workflow, based on the requirements defined by ECSS-Q-ST-40C, ECSS-Q-ST-30C, ECSS-E-ST-40C and ECSS-Q-ST-80C:

Figure-3.2: Software dependability and safety


F1
Safety Analysis
Dependability Analysis
System-Level Software Requirements Specification
F4
F2
F5
System level
Software level
Software RAMS Analysis
F3
Software Requirements Specification
F3
Identification of critical Software handling measures
Software Design Software Verification, Testing and Validation

Source: adapted from ECSS-Q-HB-80-03A (2012)

The handbook defines 5 flows, which are:

Flow F1: Safety requirements from safety analysis “translated" into requirements for software safety, as well as requirements that affect software dependability. Safety requirements are considered mandatory, whereas dependability requirements are negotiable, depending on other system characteristics and the level of risk acceptable to the customer.

Flow F2: Software criticality classification after analysis of safety. It should consider all decisions made at the system level to prevent or reduce the consequences of system failure caused by software, and should target the entire software without breaking down into components.

Flow F3: Software components criticality after analysis of the architecture. It allows for focusing engineering efforts and product assurance in the most critical components.

Flow F4: Software criticality analysis results relevant to systems level (e.g., software failures with potential critical impact on the system that were not considered at the system level analysis).

Flow F5: Information related to the HSIA, to ensure that the software reacts to hardware failures in an acceptable manner. The software requirements and potential hardware failures are inputs to the HSIA, which in turn may require defining new software requirements, if it detects that the reactions of the software for specific hardware failures are not appropriate.

3.2.3 - Project life cycle and oversight activities

This section describes those ECSS activities that are closely related to the thesis, more specifically to the oversight activities responsible for raising issues that are submitted to the metrics. Problems and or limitations are identified and tagged for convenience (refer to section 1.8).

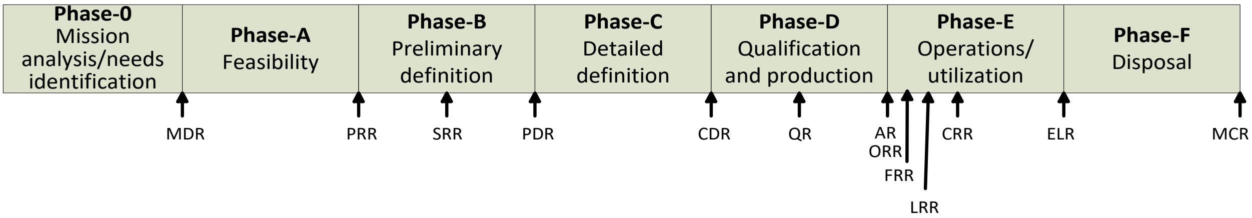
Joint reviews can be of two types: project review aiming at defining a customer approved technical baseline, and technical review aiming at defining a technical baseline, described in ECSS-E-ST-40C. The joint reviews take place at distinct phases of the project. According to ECSS-M-ST-10C (2009), the life cycle of space projects is typically divided into 7 phases, as follows:

- Phase 0 - Mission analysis/needs identification

- Phase A - Feasibility
- Phase B - Preliminary definition
- Phase C - Detailed definition
- Phase D - Qualification and production
- Phase E –Utilization
- Phase F - Disposal

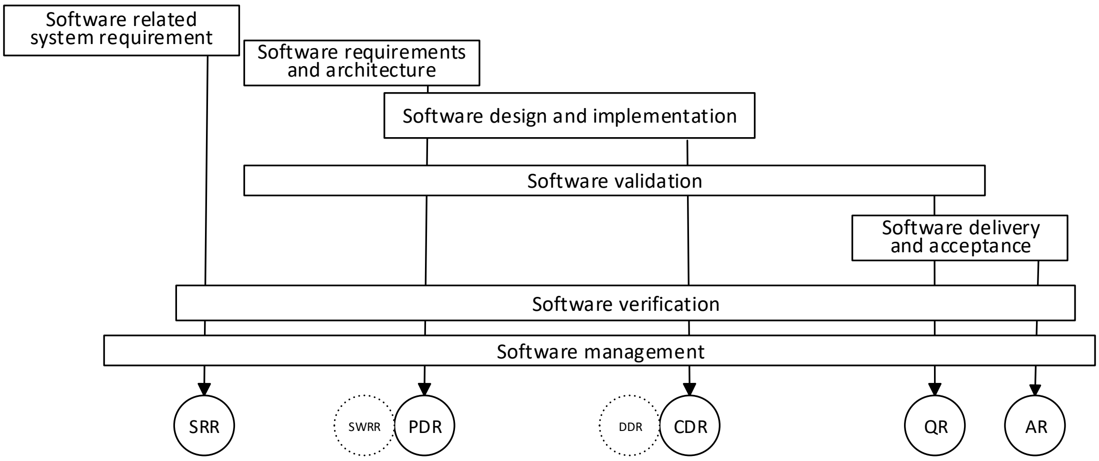
Figure 3.3 shows the project life cycle and respective reviews:

Figure-3.3: Space project life cycle



The planned reviews are: Mission Definition Review (MDR), Preliminary Requirements Review (PRR), System Requirements Review (SRR), Preliminary Design Review (PDR), Critical Design Review (CDR), Qualification Review (QR), Acceptance Review (AR), Operational Readiness Review (ORR), Flight Readiness Review (FRR), Launch Readiness Review (LRR), Commissioning Result Review (CRR), End-of-Life Review (ELR) and Mission Close-out Review (MCR). Software safety scope is more concentrated on phases B, C and D. The figure-3.4 shows an overview of software life cycle processes and related reviews:

Figure-3.4: Software life cycle processes


Software related system requirement
Software requirements and architecture
Software design and implementation
Software validation
Software delivery and acceptance
Software verification
Software management
SRR
SWRR
PDR
DDR
CDR
QR
AR

Source: adapted from ECSS-E-ST-40C (2009)

The SRR, PDR, CDR, QR and AR are project reviews. At software scope two additional reviews are planned: Software Requirements Review (SWRR) prior to PDR and Detailed Design Review (DDR) prior to CDR. The joint reviews focus on documents evaluation. The space oversight framework described in appendix-C includes documents evaluation, but focuses mainly in process evaluation and process adherence assessment, i.e., an evaluation of the actual process implementation.

Problem/Limitation-3: *The joint reviews described in ECSS-E-ST-40C focus on documents evaluation, rather than process evaluation and process adherence assessment.*

Audits are described in ECSS-M-ST-10C and can be performed by the customer, a third party, or even by the supplier of his own projects or of lower tier suppliers. It is the customer responsibility to notify the supplier in due time about the audit, objectives, scope and schedule. The oversight activities described in this thesis can fit in the context of these audits.

Problem/Limitation-4: *The audits described in ECSS-M-ST-10C are in line with the software supplier oversight of this thesis. However, the requirements provided are general and specific guidelines for software audits are lacking.*

Risk management - The Risk Management (ECSS-M-ST-80C, 2008) is a 4 steps process:

1. Define risk management implementation requirements;
2. Identify and assess the risks;
3. Decide and act; and
4. Monitor, communicate and accept risks.

Step 1 is executed at the beginning of the project, while steps 2, 3 and 4 compose a cycle that repeats throughout the project life cycle phases. For the case of outsourcing safety-critical software, those steps may include oversight activities. However, for the management the oversight would focus on the tripod 1-scope, 2-time and 3-resources, which is different from the thesis concern.

Process assessment - The ECSS-Q-ST-80C requires assessment and improvement process to be conformant to ISO/IEC-15504 (2004). In order to meet such requirement, the handbook ECSS-Q-HB-80-02A (2010) provides a framework called SPiCE for Space (S4S). The Software Process Capability dEtermination (SPiCE) is a major international initiative to support the development of ISO/IEC 15504. According to the handbook, customers can benefit from the S4S because:

- Reduces uncertainties in selecting suppliers of software by enabling the risks associated with the supplier capability to be identified before contract award;
- Enables appropriate controls to be put in place for risk containment;
- Provides a quantified basis for choice in balancing business needs, requirements and estimated project cost against the capability of competing suppliers.

Considering the current maturity level of Brazilian space industry, i.e., small companies, lack of experience, low demand from space domain, and also considering the effort needed to implement the S4S, the application of S4S may not be adequate.

Problem/Limitation-5: *Concerning process assessment and improvement, the effort needed to implement the S4S described in ECSS-Q-HB-80-02A may not be adequate for the current maturity level of Brazilian space industry or small companies in general.*

3.3 - Software safety in civil aviation domain

3.3.1 – Airborne software in civil aviation certification

An important characteristic of the civil aviation is that certification is mandatory under an agreement between the members of ICAO. Every country that manufactures or makes modifications to aircraft or other aeronautical products used in air transportation is required by ICAO to maintain a civil aviation certification organization to ensure compliance with minimum airworthiness requirements. In Brazil, that role is played by ANAC, in the United States by the Federal Aviation Administration (FAA), and in Europe by the European Aviation Safety Agency (EASA). Certification regulations require that the consequences of all failures must be analyzed, and classify according to the severity of their effects as follows:

a. Catastrophic: would result in multiple fatalities, usually with the loss of the airplane.
b. Hazardous: would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions to the extent that there would be:
    - A large reduction in safety margins or functional capabilities;

- Physical distress or excessive workload such that the flight crew cannot be relied upon to perform their tasks accurately or completely; or
- Serious or fatal injury to a relatively small number of the occupants other than the flight crew.

c. Major: would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions to the extent that there would be, for example, a significant reduction in safety margins or functional capabilities, a significant increase in crew workload or in conditions impairing crew efficiency, or discomfort to the flight crew, or physical distress to passengers or cabin crew, possibly including injuries.

d. Minor: would not significantly reduce airplane safety, and involve crew actions that are well within their capabilities.

e. No safety effect: for example, would not affect the operational capability of the airplane or increase crew workload.

A failure with catastrophic consequences in theory should never occur during the fleet lifetime of an aircraft type, while those with less severe consequences are more tolerated. Table-3.3 describes the quantitative and qualitative probabilities associated with each failure condition.

Table-3.3: Failure condition classification

Failure Condition	Quantitative Probability	Qualitative probability
Catastrophic	$< 10^{-9}$	**Extremely improbable:** so unlikely that they are not anticipated to occur during the entire operational life of all airplanes of one type
Hazardous	$< 10^{-7}$	**Extremely remote:** not anticipated to occur to each airplane during its total life but which may occur a few times when considering the total operational life of all airplanes of the type
Major	$<10^{-5}$	**Remote**: unlikely to occur to each airplane during its total life, but which may occur several times when considering the total operational life of a number of airplanes of the type
Minor	$<10^{-3}$	**Probable**: anticipated to occur one or more times during the entire operational life of each airplane

Source: Adapted from FAA (2002)

Certification regulations specify levels of safety that are required. Ensuring an acceptable level of safety should always take into consideration:

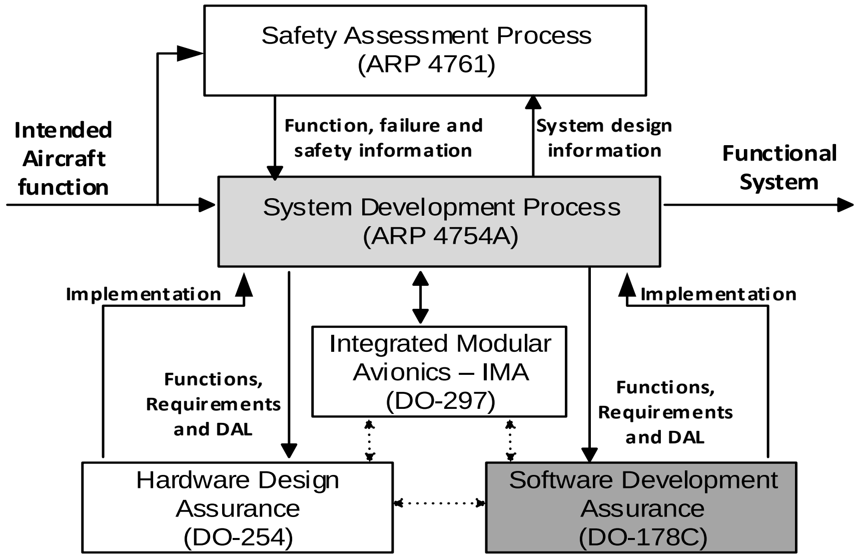
a. Number of failures: No single failure can lead to catastrophic consequence, regardless of how remote is the occurrence of the failure;

b. Probability: For each category of failure it should have a quantitative or qualitative measure that satisfy the Table-3.3, and

c. Design evaluation: Assessment to confirm the absence of design errors.

Aircraft systems and their hardware usually require all three considerations. For software, only design evaluation is required. In order to verify the considerations (a) and (b), a systematic safety analysis of the aircraft and its systems should be performed and is described in standard SAE/ARP4761 (1996). Concerning the consideration (c), for design evaluation applied to the aircraft and its systems, recommendation has been developed with the basic idea of minimizing the development errors that may adversely affect safety by the systematic application of a set of development processes and V&V, and can be found in SAE/ARP4754A (2010). Similarly, the design evaluation for airborne electronic hardware (AEH) follows the standard RTCA/DO-254 (2000), and airborne software items follow the standard RTCA/DO-178C. The RTCA/DO-297 (2005) is for Integrated Modular Avionics (IMA) development, which is a specific architecture for aviation domain. Figure-3.5 shows the relationship between these standards.

Figure-3.5: Aviation standards covering system, safety, software and hardware


Safety Assessment Process (ARP 4761)
Intended Aircraft function
Function, failure and safety information
System design information
Functional System
System Development Process (ARP 4754A)
Implementation
Implementation
Integrated Modular Avionics – IMA (DO-297)
Functions, Requirements and DAL
Functions, Requirements and DAL
Hardware Design Assurance (DO-254)
Software Development Assurance (DO-178C)

Source: Adapted from SAE-ARP4754A

An important result of the system development process is the Development Assurance Level (DAL) for system, software and hardware. The degree of effort and detail required to perform development activities depends on the DAL assigned to the system and its hardware and software items. The DAL is based on the most severe failure condition classification associated with a function which has been implemented in system, software or hardware. It is important to mention that in the civil aviation the safety assessment process is restricted to the aircraft and its systems, and the software addresses safety through the satisfaction of the DAL assigned to it. Table-3.4 shows the failure conditions classification and the corresponding software DAL required. For further details on classification of failure conditions, refer to FAA (2002).

Table-3.4: Failure conditions and respective levels of software

Failure Condition	Software DAL
Catastrophic	A
Hazardous	B
Major	C
Minor	D
No Effect	E

Note: It is not acceptable to assign probabilistic numbers to software levels

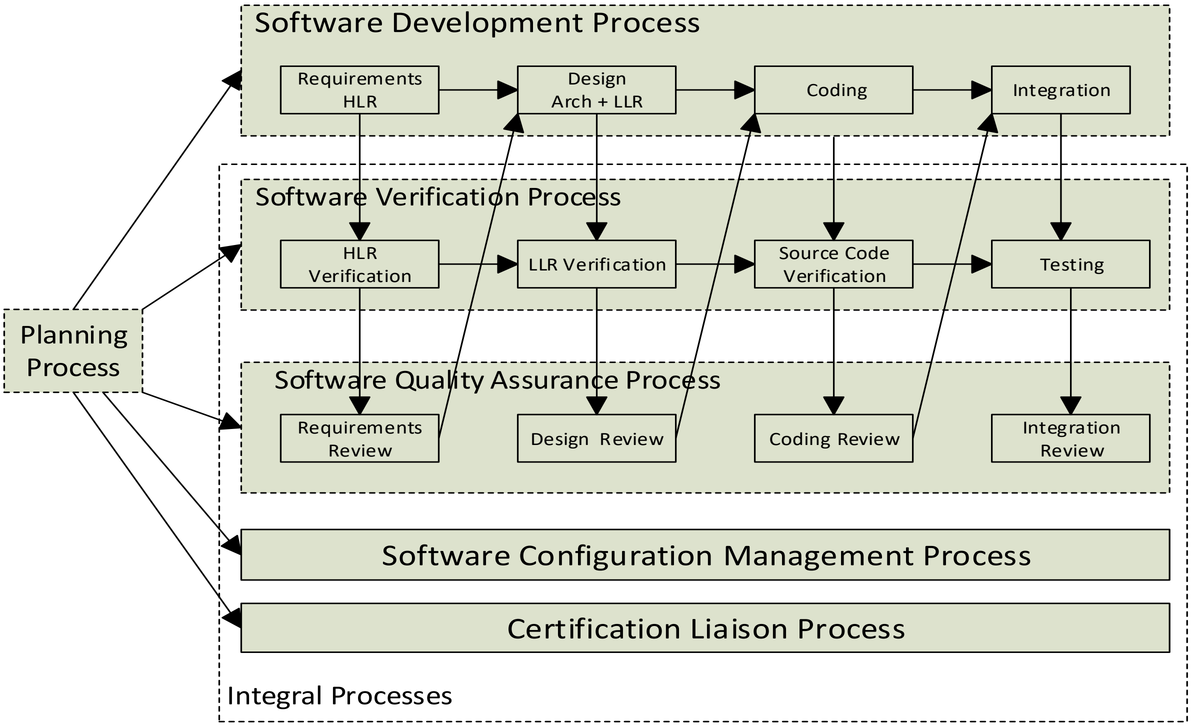
The software DAL in Table-3.4 can also be applied to partitions. According to Rushby (1999), partitioning is a technique to provide isolation between functionally independent software components to contain and / or isolate failures and potentially reduce the effort of the software verification process.

Civil aviation does not use the term "dependability". The principal component of dependability is safety, since the goal of civil aviation certification is to ensure an acceptable level of safety in airworthiness. Thus, the existing standards and recommendations consider the other components of dependability (reliability, availability, maintainability, and even security) regarding the impact in safety.

3.3.2 – The *RTCA/DO-178C*

The DO-178C, entitled "Software Considerations in Airborne Systems and Equipment Certification", provides recommendations for the development and certification of software on board civil aircraft. It contains guidelines for determining consistently and with an acceptable level of confidence that the software aspects comply with certification regulations. Figure-3.6 shows the DO-178C processes.

Figure-3.6: The DO-178C processes



Source: Adapted from the RTCA training material

The planning process defines and coordinates the activities of the software development process and integral processes for a project. The software development process produces the software product, and comprises the following phases:

a. Requirements specification, where high-level software requirements (HLR) are created;

b. Design, where the software architecture is generated and the HLR are refined into low-level requirements (LLR) whose level of detail allows for its implementation in programming language;

c. Coding, where the LLRs together with the software architecture are transcribed into source code using programming language; and

d. Integration and testing, where the software components are integrated into the target hardware and the executable code is exercised in a test environment representative of the actual system.

The integral processes ensure correctness, control, and confidence in the processes of the software life cycle and its outputs, and consist of verification, quality assurance, configuration management, and certification. The transition criteria between phases, generated life cycle data, and additional considerations (software reuse, tool qualification, alternative methods) are also described. A list of 71 objectives is provided and if the developer can show compliance with the objectives applicable to the software according to the DAL, the software will be approved for use in the aircraft under certification. Table-3.5 shows the distribution of the number of objectives by processes of DO-178C.

Table-3.5: Number of objectives for each process

Process →	**Planning**	**Develop.**	**Verific.**	**Config. Manag.**	**Quality Assur.**	**Certif.**	**Total**
Number of Objectives	7	7	43	6	5	3	71

The significant effort is spent in the verification process which consists of a technical assessment of the software development process, and includes activities such as reviews, analysis and testing. Table-3.6 shows the number of objectives for each software DAL, listing whether objectives should be met with or without independence. In this context, independence according to DO-178C means that the verification activity must be performed by a person different from the one who developed the item to be verified.

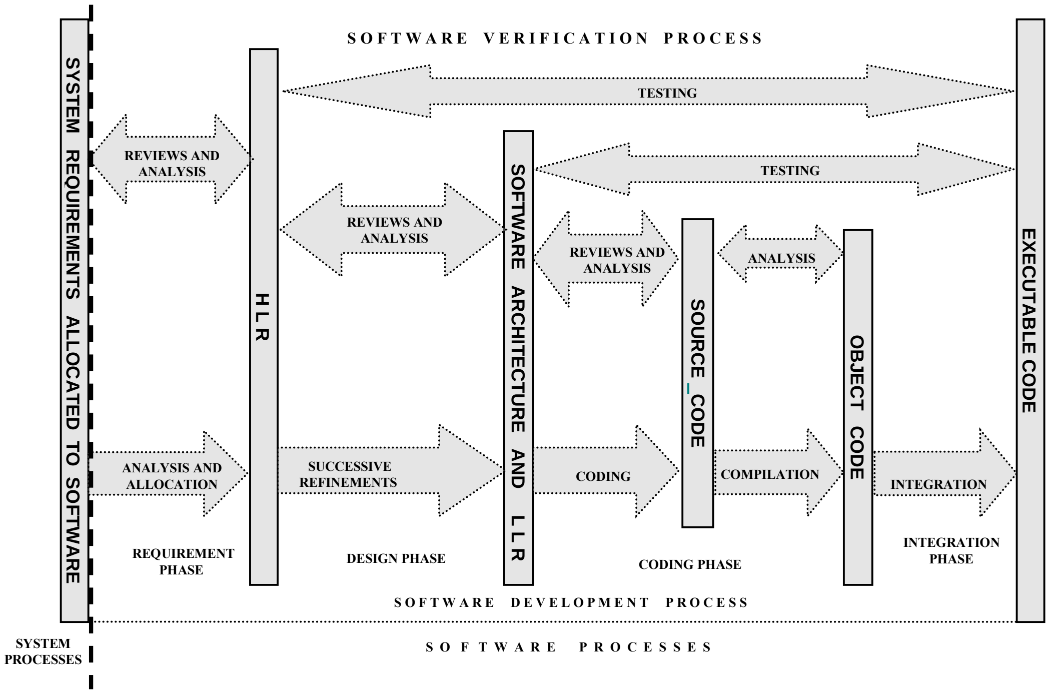
Table-3.6: Number of objectives for each software level

Software DAL	Number of Objectives		
	With	without	Total
A	30	41	71
B	18	51	69
C	5	57	62
D	2	24	26

Note: With = with independence
Without = without independence

The higher the software DAL, the more rigorous the guide will be, i.e., more objectives should be met. The DO-178C does not provide any guidance for software classified as DAL E, as there is no impact on safety. Figure-3.7 illustrates the software development process and the verification process in parallel. The DO-178C puts emphasis on requirements-based testing (LLR and HLR). In addition, traceability from system requirements to software requirements, going through HLR, LLR, source code, object code (for DAL A), and including test cases, procedures and results is mandatory.

Figure-3.7: Flows of development and verification processes



Source: Adapted from the draft material of the working group (RTCA 2007)

Note: Some verification activities are not represented to not compromise clarity (e.g., coverage analysis)

Regarding the analyses, DO-178C requires to analyze the structural coverage of the code for DAL A, B and C (with different degrees of accuracy) for the detection of dead or deactivated code, which may indicate the presence of non- intended function. Further information can be found in Dupuy and Leveson (2000) and FAA (2001).

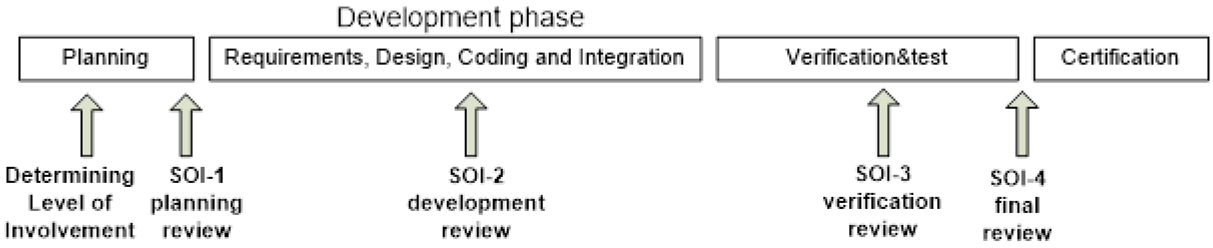
The DO-178C has statistics in its favor: no catastrophic accident had software as its main cause. However, this information does not indicate that software developed according to DAL-A is infallible, since it is customary practice to mitigate the effects of possible failures of software DAL-A through other means (e.g., system architecture mitigation, electro-electronic, mechanical, operational, etc.). Information on accidents can be found in Leveson (2004), MIT (2017), NASDAC (2017) and NTSB (2017).

DO-178C also requires qualification of software tools, when used to eliminate, reduce or automate part of the planned activities, without their outputs being verified. Additional information on tool qualification can be found in RTCA/DO-330 (2011), and guidance for use of Model-Based Development and Verification – MBDV in RTCA/DO-331 (2011), Object-Oriented Technology – OOT in RTCA/DO-332 (2011) and Formal Methods in RTCA/DO-333 (2011). Clarifications on the DO-178C in the form of frequently asked questions or articles on specific topics can be found in RTCA/DO-248C (2011).

3.3.3 - Certification authority level of involvement

The DO-178C states that the certification authority may review the software life cycle processes and data to assess compliance to DO-178C. The FAA Order 8110.49 (2011) provides guidelines related to those reviews, and the figure-3.8 illustrates when they occur during the software life cycle:

Figure-3.8: Certification authority review



The Order 8110.49, chapter 3 provides guidelines on determining the authority level of involvement in supplier, which consider but is not limited to: DAL, product attributes (e.g., size, complexity), use of new technologies, design features, methods, supplier previous experience. The resulting level of involvement may be from none to all reviews called Stage of Involvement (SOI), which are summarized below:

- ***SOI#1 – planning review:*** assure plans, standards, and processes meet DO-178C objectives and other applicable software certification guidance;

- ***SOI#2 – development review:*** assess implementation of plans and standards for the software development to ensure that the resulting life cycle data satisfies DO-178C objectives and other applicable certification guidance;

- ***SOI#3 – verification review:*** assess implementation of verification as planned to ensure that its activities satisfy DO-178C objectives and other applicable certification guidance;

- ***SOI#4 – final review*:** assure final software product meets DO-178 objectives and is ready for certification.

Remark: In this work, the term SOI will be referred as stage (i.e., Stage#1, Stage#2, Stage#3 and Stage#4).

To assist in performing software reviews, the FAA has written a Job Aid (2004), which addresses tasks to be performed before, during and after the review, activities and questions to be considered during the review, key players primary roles and responsibilities, review readiness criteria and issues classification.

Although determining the level of involvement and performing related reviews are under certification authority scope, aviation companies usually do similar activities in order to mitigate certification risk. In this case, it is in the scope of supplier oversight and from now on will be called *Aviation Oversight.* It was used as reference for the classification of the results of the systematic comparison process, and comprises a set of procedures, checklists and applicable standards. It is important to note that, although Order 8110.49 refers to DO-178B, the content is still applicable for defining the *Aviation Oversight* in the scope of this work, as the basic characteristic has been preserved from DO-178B to DO-178C, and the main differences are in the supplements that provide specific technology-dependent guidance (e.g., model-based development, object-oriented technology and formal methods).

3.3.4 – The limitation of the review result classification

This section describes the problems and or limitations that are closely related to this thesis, and tags them for convenience (refer to section 1.8). The Job aid presents a classification of stages results. These results influence the decision of the next steps of the certification authority, which can be from the re-execution of a stage (the worst scenario) to the non-execution of the next stages (for the best scenario). Consequently, the applicant and the software supplier give importance to these results. However, such classification may not reflect the quality of the development and verification processes of the audited software, and may lead to inappropriate interpretations that affect managerial decisions. The FAA Job Aid classifies the results of the reviews (stages) as follows:

- **Compliance:** the satisfaction of a DO-178 objective.
- **Finding:** the identification of a failure to show compliance to one or more of the RTCA/DO-178 objectives.
- **Observation:** the identification of a potential software life cycle process improvement. An Observation is not an RTCA/DO-178 compliance issue and does not need to be

addressed before software approval.

- **Action:** an assignment to an organization or person with a date for completion to correct a Finding, error, or deficiency identified when conducting a software review.

ANAC uses the Job Aid as reference, but has adjusted the classification as follows:

- **Finding***: a non-compliance to a DO-178 objective. A Finding may have instances of non-compliance linked to it. It should be addressed covering not only the specific instances of non-compliance, but also any systemic process deviations. An assessment for impacts in the activities already performed is also expected for any Finding.
  Note *: in the case of Stage#1, it is assumed that it is sampled and therefore, it cannot be stated that the information does not exist, is incomplete, or is not clear, requiring in these cases an Action instead of Finding.

- **Action**: a request for clarification purposes. It may become a Finding if not provided or if the Action response drives to an evidence of non-compliance. A clarification provided may need to be incorporated in the life cycle data (as applicable) to correct a deficiency or an error, even if it does not become a Finding.

- **Observation**: identification of potential process improvement. An Observation is not a compliance issue; hence, it is not requested to be addressed prior to software approval.

The above classifications have limitations if used to measure the audit result. Only the number of Findings and Actions can give the wrong view. Many Findings may not necessarily be indications that the software development process is bad and vice versa, few Findings (or even absence of) may not mean that the outcome is good, since a single Finding is enough to generate a heavy rework and even compromise the project. In addition, an Action can become a very impacting Finding, and this is not reflected in the initial audit result. It is worth to mention that depending on the way the items are organized (grouped), the number of Findings and Actions can vary greatly.

Figure-3.9: Findings, actions and observations

There are **FINDINGS** and findings,

ACTIONS and actions.

Observations are just observations.

The following are cases of inappropriate use of the classification for audit evaluation:

- Company overreacted against a large number of Findings even before evaluating the technical severity of the items, generating unnecessary stress at managerial level;

- Company trying to use the absence of Findings in an audit to argue about possible reduction of level of involvement;

- Company performing supplier oversight avoided classifying the issue as Finding, and created terms like “Major-Action” for mitigating the managerial impact;

- An Action has recorded lack of readiness, but did not receive proper managerial attention because it was not a Finding. However, the impact was very heavy;

- Post-Stages activities have detected some Actions as non-compliance cases demanding heavy workload, but stage's initial outcome was not revised to better reflect the supplier's situation.

Problem/Limitation-2 is again applicable: *In the civil aviation software audit, the criteria used for issue classification are not adequate for evaluating the audit result and may lead to inappropriate interpretations that can adversely affect managerial decisions.*

For further information on software safety in civil aviation, Sozen (2012) proposed the use of adapted software product line engineering for complex certifiable avionics software, Rierson (2013) provided a complete material on developing safety-critical software, Romanski (2012) wrote some considerations on combining safety and security certification, Kornecki (2008) showed the role of software certification in development of dependable systems, Sakugawa et al. (2005) presented airborne software concerns in civil aviation certification, Cury and Sakugawa (2004) described the Brazilian experience in civil aviation certification concerning software, and Marques et al. (2012) described the use of MBD as software low-level requirement (LLR) to achieve airborne software certification.

3.4 – Summary of chapter 3

This chapter provided a summary of software safety in aerospace domain with emphasis on standards of civil aviation for aeronautics and ECSS for astronautics. Limitation on evaluation of software audit result was identified in civil aviation, together with opportunity for improvements by adequate metrics. Software audits, supervision or oversight-like activities were also identified in the ECSS standards, suggesting the possibility of extending such opportunity to the space domain (i.e., Aerospace Metrics). For further information, the appendix-B, section-B.2, provides a summarized comparison between the main software safety standards from both domains.

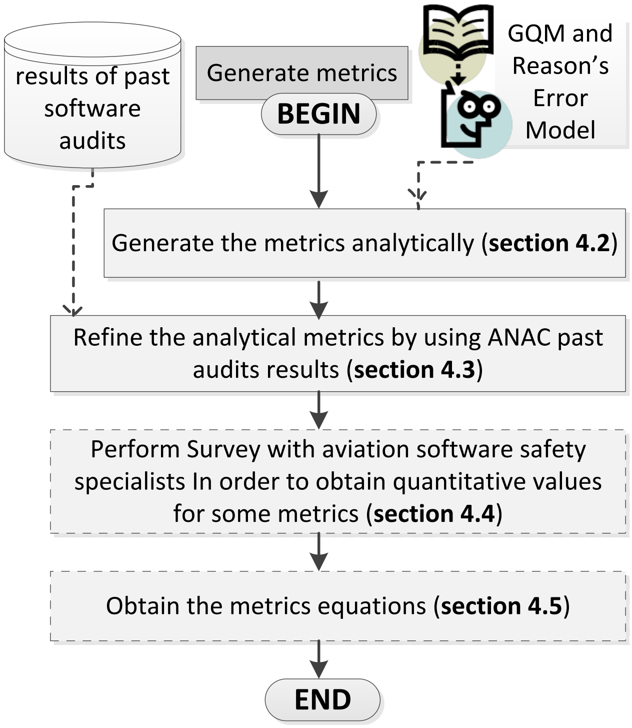
Remark: Near completion of this thesis, the FAA revised Order 8110.49 (2017) to allow flexibility in conducting software reviews and for alignment with their risk-based directives. The revised Order no longer prescribes the SOIs for compliance assessments, but leaves the choice for the stakeholders, which may be a more continuous oversight, a set of SOIs not necessarily in four stages, or even a single full coverage review at the end of development. Although this thesis assumes a stage-based involvement (i.e., SOI) for building the metrics and related oversight activities, the revised Order does not invalidate the thesis contribution to aviation domain. However, some adjustments are deemed necessary and are suggested as future work in chapter 8.

4. THE METRICS GENERATION PROCESS

4.1 - Overview

This chapter presents the generation of the metrics, which is related to the relevance cycle of the Design Science. The generation is supported by consolidated techniques (GQM and Reason´s human error model), as well as contribution of software safety expertise and vast material accumulated by many years of experience of the civil aviation certification. The use of GQM and Reason´s model is related to the rigor cycle of the Design Science. The figure 4.1 illustrates the overall process, whose general context is illustrated in the figure-1.1:

Figure-4.1: The process for metrics generation


results of past software audits
Generate metrics
BEGIN
GQM and Reason's Error Model
Generate the metrics analytically (section 4.2)
Refine the analytical metrics by using ANAC past audits results (section 4.3)
Perform Survey with aviation software safety specialists In order to obtain quantitative values for some metrics (section 4.4)
Obtain the metrics equations (section 4.5)
END

Generate the metrics analytically: Combine GQM and Reason's human error model to generate the metrics analytically**.** The GQM technique assists in systematically identifying metrics for all types of issues that can possibly be raised in software audits, whereas the Reason's human error model assists in classifying the root cause of those types of issues.

Refine the analytical metrics by using ANAC past audits results: The results of past audits performed by ANAC, some jointly with the main international certification authorities (i.e., FAA and EASA), are analyzed and mapped into the metrics identified analytically. Thus, it is possible to adjust some metrics, identify additional ones, and remove those deemed not effective in practice.

Perform Survey with aviation software safety specialists to associating quantitative values for some metrics: The adjusted metrics are submitted to a survey with software safety senior specialists from the civil aviation to obtain quantitative values for their severity and relevance, as well as suggestions of additional metrics. The results of the survey are compiled and summarized in tables and graphics, and further discussed and analyzed by the survey participants in a dedicated workshop. Potential dependency among some metrics is also discussed, as well as the additional metrics suggested in the survey. As an output, a consolidated list of metrics and related quantitative values are produced, together with some considerations on dependency among those metrics.

Obtain the metrics equations: The consolidated list of metrics, their quantitative values and dependency considerations are analyzed for generating equations to calculate the related measurements. The equations express the Aerospace Metrics and are further validated in aeronautics and astronautics domains.

4.2 - The analytical metrics generation

The GQM is a systemic approach to identifying and organizing metrics of interest according to the organization goals. It is a goal-driven top-down approach and composed of:

a. Conceptual level: a process for identifying goals;

b. Operational level: generation of questions that help in characterizing the way of assessment/achievement of a specific goal;

c. Quantitative level: Specification of metrics to answer the questions.

Goals have five attributes:

a. The object of interest;
b. The purpose of studying the object of interest;
c. The focus on the characteristics of the object of interest;
d. The stakeholder of the goal; and
e. The context of the study of the object of interest.

Particularly for the case of this thesis, only one goal was applicable, and the related attributes are illustrated in the table-4.1:

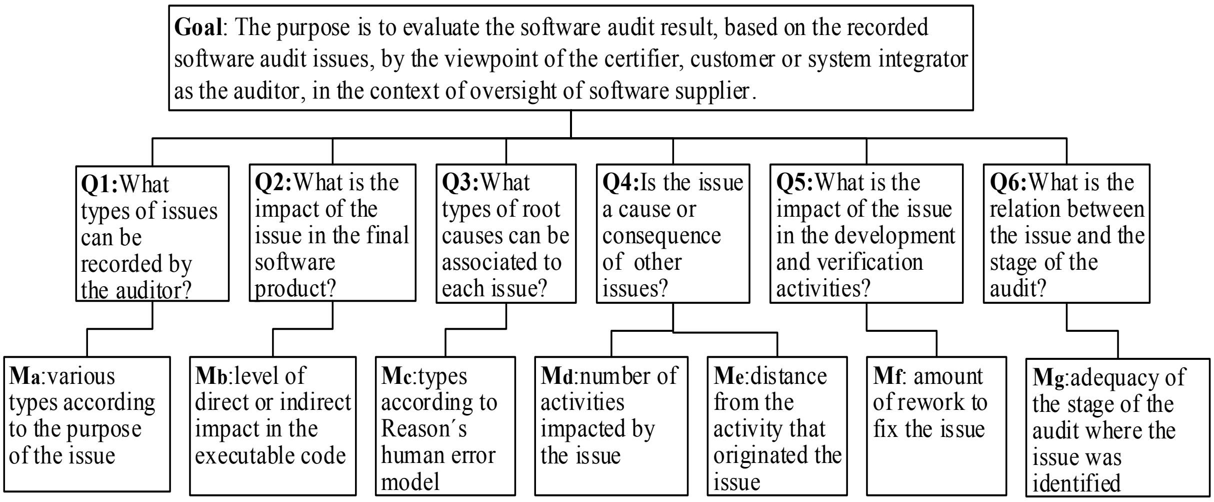
Table-4.1: The goal´s attributes

Attribute	Content
Purpose	Evaluation
Object of interest	Software audit result
Focus	Recorded software audit issues
Viewpoint	Certifier, customer or system integrator as the auditor
Context	Oversight of software supplier

Goal statement: *The purpose is to* ***evaluate*** *the* ***software audit result,*** *based on the* ***recorded software audit issues,*** *by the viewpoint of the* ***certifier, customer or system integrator as the auditor,*** *in the context of* ***oversight of software supplier****.*

After specifying the goal, a set of questions was built to provide confirmation, clarity and coverage of the various aspects of the goal. For each question, a set of data was associated to answer it in a quantitative way, i.e., one or more metrics were identified. The figure-4.2 illustrates the final GQM diagram, where M(a…g) represent the set of metrics generated analytically by applying the top-down GQM technique, starting from the goal, and going through the Q(1…6) questions identified.

Figure-4.2: The GQM diagram for metrics generation


Goal: The purpose is to evaluate the software audit result, based on the recorded software audit issues, by the viewpoint of the certifier, customer or system integrator as the auditor, in the context of oversight of software supplier.
Q1: What types of issues can be recorded by the auditor?
Q2: What is the impact of the issue in the final software product?
Q3: What types of root causes can be associated to each issue?
Q4: Is the issue a cause or consequence of other issues?
Q5: What is the impact of the issue in the development and verification activities?
Q6: What is the relation between the issue and the stage of the audit?
Ma:various types according to the purpose of the issue
Mb:level of direct or indirect impact in the executable code
Mc:types according to Reason´s human error model
Md:number of activities impacted by the issue
Me:distance from the activity that originated the issue
Mf: amount of rework to fix the issue
Mg:adequacy of the stage of the audit where the issue was identified

The identified metrics are described as follow:

Metric-Ma - An issue can be of the following types according to the purpose. The following cases have been identified:

a. A suggestion for process improvement detected during the audit. However, the process is considered sufficient for compliance;

b. An issue to correct a punctual process deficiency (or adherence to the process) detected during the audit;

c. An issue to record a non-compliance and request a closure approach;

d. An issue to request additional information, which may drive to a non-compliance that was not conclusive during the audit;

e. An issue to request additional information, but a priori without any impact in items discussed during the audit;

f. An issue related to document evaluation.

Metric-Mb - An issue related to non-compliance may have direct or indirect impact in the executable code. Examples include:

a. If the non-compliance inserted an error in the executable code

b. If the non-compliance has the potential to insert an error in the executable code

c. If the non-compliance failed to detect an error in the executable code

d. If the non-compliance failed to detect an error that has potential to insert an error in the executable code

e. If the non-compliance failed to detect process adherence deficiency, but without clear impact in the executable code

Metric-Mc - An issue may have severity levels, depending on the root cause. The Reason´s human error model has been used to aid in classifying types of root causes. The following explanation was extracted from Howden (2011), with adjustments in the organization of the text. James Reason classifies human errors as follow:

a. *Slip Errors*: The correct solution is formulated, but a slip occurs during its execution.

b. *Rule Errors*: Rules are parts of knowledge in the form of "if condition, then do action". They are solutions established and reused repeatedly. Rule errors are subdivided into:

   i. *Bad rules*: they correspond to bad solution techniques that are wrong and must be unlearned.

   ii. *Misapplied rules*: can occur in diverse ways, such as failure to satisfy all conditions, or incorrect application of the action.

c. *Errors of knowledge*: are associated with the most laborious parts of the solution of the problem, where the solver should resort to reasoning step-by-step from the first principles. They are subdivided into:

   i. *Inaccurate mental model*: correspond to errors results of ignorance (or lack of knowledge).

   ii. *The limited Workspace*: Errors caused by the human brain's span limit that can only handle a small number of things simultaneously.

d. *Memory Prediction Errors*: A situation where there was a conscious intention to do something, but the resolution was lost.

e. *Breach Errors*: situation where the solver knows that a particular action might not be appropriate, but for some reason, such as schedule pressure, it does anyway.

Metric-Md - An issue related to non-compliance may have severity levels, depending on the number of activities impacted. In the examples that follow, measurements are captured by "*NumberOfActivities"*:

a. If the non-compliance does not impact other activities, e.g., isolated coding error (*NumberOfActivities*=0)

b. If the non- compliance has impacted a second activity, e.g., coding error that was not detected by code review (*NumberOfActivities*=1)

c. If the non- compliance has impacted two other activities, e.g., coding error that was not detected by code review, nor by tests (*NumberOfActivities*=2)

Metric-Me - An issue related to non-compliance may have severity levels, depending on the distance from the activity that originate it. In the examples that follow, the suggested measurements (i.e., *DistanceFromOrigin*) are based on a development process composed of the following phases: system requirements specification, software requirements specification, detailed design, coding, integration, testing.

a. Non- compliance detected in tests, but resulting from system requirement error (*DistanceFromOrigin*=5)

b. Non- compliance detected in tests, but resulting from software requirement error (*DistanceFromOrigin*=4)

c. Non- compliance detected in tests, but resulting from detailed architecture design (*DistanceFromOrigin*=3)

d. Non- compliance detected in tests, but resulting from coding (*DistanceFromOrigin*=2)

e. Non-compliance detected in tests, but resulting from integration (*DistanceFromOrigin*=1)

Metric-Mf - An issue may impact the development and verification activities depending on the amount of rework to fix it. Examples include:

a. Corrections in documents, standards or checklists, but without impact in processes;

b. Corrections in documents, standards or checklists, with impact in processes demanding training;

c. Corrections in process with clear impact in artifacts (e.g., requirements);

d. Corrections in process, demanding an analysis to determine impact in artifacts;

e. Corrections in artifacts, but demanding root cause analysis to identify any process deficiency;

f. Corrections in process and related artifact, but added by regression analysis to identify impact in activities already performed and related artifacts generated.

Metric-Mg - The same issue may have different relevancies if found in different stages of the audits. This metric relates to adequacy of the issue regarding to the stage of the audit where the issue was identified. The worse the adequacy, the greater the severity. Examples in aviation oversight include:

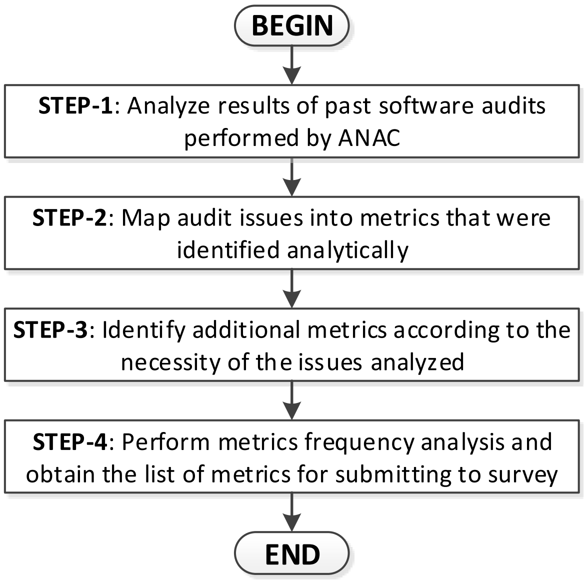
a. Issue identified in adequate Stage (*Adequacy* = 0).
b. Stage#2 scope issue identified in Stage#3 (*Adequacy* = 1);
c. Stage#1 scope issue identified in Stage#3 (*Adequacy* = 2);
d. Stage#1 scope issue identified in Stage#4 (*Adequacy* = 3);

Afterwards, the metrics identified analytically in this section were refined by using the results of ANAC past audits. Some of the metrics did not allow for direct quantitative measurement (i.e., Ma, Mb, Mc, Mf). Therefore, after the refinement they were submitted for survey with software safety senior specialists in order to obtain quantitative values.

4.3 - Using ANAC past audits to refine the analytical metrics

The metrics identified analytically were refined by using results of past audits performed by ANAC. The figure 4.3 shows the flowchart of the process used:

Figure-4.3: The process for metrics refinement using results of past audits


BEGIN
STEP-1: Analyze results of past software audits performed by ANAC
STEP-2: Map audit issues into metrics that were identified analytically
STEP-3: Identify additional metrics according to the necessity of the issues analyzed
STEP-4: Perform metrics frequency analysis and obtain the list of metrics for submitting to survey
END

STEP-1, Analyze results of past software audits performed by ANAC:

Audit results performed since 2006 were analyzed, covering 4 certification programs, 19 different software suppliers from various systems, e.g., flight controls, avionics, landing gear, air management, brake, and electrical system. Those suppliers are mainly from the USA and

Europe, and they supply critical systems to the top-leading aircraft manufacturers. Approximately 1300 issues of 44 audits comprising variety of stages (i.e., Stage#1, Stage#2, combined Stage#2/3, and Stage#3) were analyzed with the focus on the applicability of the analytical metrics. Some audits were performed jointly with the main international certification agencies (e.g., FAA and EASA). The table-4.2 provides a summary of ANAC past audits result:

Table-4.2: Summary of ANAC past audits per certification program

Certification Program	Type of system	Number of audits	Number of issues
Aircraft-A	Avionics, Flight Controls, Air Management	15	594
Aircraft-B	Flight Controls, Electrical, Air Management, Brake, Landing Gear	17	441
Aircraft-C	Flight Controls	8	146
Aircraft-D	Flight Controls	4	120

The information provided in table-4.2 does not cover the whole set of audits performed by ANAC, but can be considered representative in terms of percentage and characteristics. The table-4.3 provides the distribution of the issues per audit and type of system, and the last rows provides the number of audits per stage and the average number of issues per audit:

Table-4.3: Distribution of audits issues per type of system and stages

Type of audit▶ Type of system▼	Stage# 1	Stage# 2	Stage# 2/3	Stage# 3	Total
Avionics	182	26	25	23	256
Air Management	204	54	7	7	272
Brake	-	10	-	8	18
Electrical	151	41	-	32	224
Flight Controls	314	137	30	3	484
Landing Gear	38	5	-	4	47
Total issues▶	889	273	62	77	1301
Number of audits▶	17	16	4	7	44
Issues per audit▶	52.3	17.1	15.5	11	29.6

Remark: as can be noticed from the last row, the tendency is always to decrease the number of issues as the development and verification activities approach the final product.

STEP-2, Map audit issues into metrics that were identified analytically:

The issues of past audits were mapped into the analytical metrics according to the metric applicability. Each issue was mapped to at least one metric. More than one metric may be associated with an issue, but not all metrics are necessarily applicable to each issue. For example, the metric M_a (purpose of the issue) is applicable to all issues, but the metric M_c (root cause) may not be applicable to those issues whose purpose (M_a) is not related to non-compliance.

STEP-3, Identify additional metrics and cases according to the necessity of the issues analyzed:

For analyzed issues that contained characteristics that could not be mapped to any of the analytical metrics or metric cases, additional metrics or metric cases were identified.

STEP-4, Perform metrics frequency analysis and obtain the list of metrics for submitting to survey:

It was performed an analysis of the frequency of occurrence of analytical metrics in relation to the mapping of audit issues. For the metrics that did not obtain any occurrence or are very low, an evaluation was made questioning the applicability and relevance, and if justifiable, they were eliminated. The table-4.4 presents the summary of the metrics refinement.

Table-4.4: Summary of metrics refinement

Analytical Metric	Freq.	Action	Justification	Refined Metric	Qfb?
M_a *(Extra)*- document evaluation	878	Create	See Justification-1	M_1 – document evaluation	NO
M_a- type of issues according to the purpose	408	Adjust	See Justification-2	M_2 – type of issues according to the purpose	NO
M_b- direct or indirect impact in the executable code	385	Adjust	See Justification-3	M_3 - type of artifact impacted by the issue	NO
M_c- the root-cause of the issue	183	Adjust	See Justification-4	M_4 - Root cause of the issue	NO
M_d- number of activities impacted by the issue	13	Remove	See Justification-5	N/A	N/A
M_e- distance from the activity that originate the issue	385	Adjust	See Justification-6	M_5 - Distance from the issue to the final product	yes
M_f- amount of rework to fix the issue	390	Adjust	See Justification-7	M_6 - Amount of artifacts impacted by the issue	yes
M_g- issue adequacy in regard to the stage of the audit	408	Keep	N/A	M_7 - Adequacy of the issue regarding to the audit stage	Yes

Notes: Qfb = Quantifiable, Freq = Frequence

The refinement of the metrics was possible because of the vast amount of issues recorded during ANAC past audits since 2006, which were very representative of types of issues, types of audits stages, types of aviation systems, and types of aircrafts. For every action performed in the refinement, a justification was provided and the description is as follows:

Justification-1: It demands a dedicated metric labeled M_1, applicable only for issues related to documents evaluation, whose cases are as follow:

a. The information contains editorial errors (typos);
b. The information is out of context, e.g., recorded in an inappropriate section or document;
c. The information is inconsistent between sections or documents;
d. The information is confused, ambiguous;
e. The information is clear and complete, but is considered unacceptable;
f. The information is superficial or incomplete;
g. Could not find in the provided documents the required information for compliance.

Justification-2: It is applicable to all issues but was adjusted and re-labeled to M_2 because the purpose "document evaluation" demands a separate metric and was removed.

Justification-3: It was difficult to identify those $\mathrm{M_b}$ cases in the issues evaluated. For example, if an issue is related to a non-compliance of test cases review, the description of the issue would be something like "*If the non-compliance failed to detect an error that has potential to fail to detect an error in the executable code",* which is quite confusing. Changed to a more pragmatic way, which is related to the type of artifact impacted, and re-labeled to M_3 - type of artifact impacted, whose cases are as follow:

a. Issue opened against plans and standards;

b. Issue opened against requirement, design or code (e.g., ambiguous requirement, architecture incompatible with requirements, code does not fully implement the requirement);

c. Issue opened against verification cases and procedures (e.g., defective test cases/procedures, non-representative test environment, insufficient analysis strategy);

d. Issue opened against verification results and related artifacts (e.g., checklist filled with errors, checklist questions insufficient for revision needs, incorrect test result not detected by the review);

e. Issue opened against traceability (e.g., requirement traces to wrong parent requirement, insufficient granularity);

f. Issue opened against tools (e.g., poor qualification report, justification for non-qualification is unacceptable);

g. Issue opened against Problem Reports (PR);

h. Issue opened against software configuration management records;

i. Issue opened against Software Quality Assurance (SQA) records;

j. Issue opened against informal data (e.g., an SQA spreadsheet for informal control not planned for use by the process).

Justification-4: Some Reason's human error classification was not mapped from any evaluated issue, as follows:

a. *Errors of knowledge - Inaccurate mental model*: the personnel qualification criteria are beyond the scope of the metrics for software supplier oversight of this thesis; therefore, no issues had root cause related to this classification;

b. *Breach Errors*: This type of root cause is related to managerial pressure and ethically questionable decisions. Therefore, it is not possible for the auditor to clearly assign this type of root-cause to an issue. Nor is such root-cause expected to be provided to the auditor after supplier analysis.

This metric was adjusted and re-labeled to *M4* - root-cause effectively mapped to Reason's human error model, whose cases are as follow:

a. It was only a slip, an isolated case (Reason´s classification: *Slip Errors)*;

b. The amount and complexity of the information needed for the activity may have contributed to the mistake (Reason´s classification: *Errors of knowledge - The limited Workspace; Memory Prediction Errors);*

c. Similar cases were found involving the same person, raising suspicion of insufficient training (Reason´s classification: *Rule Errors - Misapplied rules)*;

d. The training material was deficient, raising suspicion that the person did not understand the activity to perform (Reason´s classification: *Rule Errors - Misapplied rules)*;

e. The process followed was clear but incorrect, leading the person to the mistake (Reason´s classification: *Rule Errors - Bad rules)*;

f. The process followed was not clear, which may have contributed to the mistake (Reason´s classification: *Rule Errors - Bad rules or Misapplied rules).*

Justification-5: Very few occurrences of this metric were detected in those issues evaluated. The explanation is because usually separated issues are recorded for each activity that was impacted by the initial issue. Therefore, this metric was considered not necessary and was discarded.

Justification-6: The metric is useful for evaluating the effectiveness of the verification by recording the source of error detected by the testing. Such metric is more related to evaluation of software development and verification process (i.e., process metrics), rather than software audits results (i.e., project metrics). Audits are based on samplings, which hardly ever include performing test execution (exception is test witness). Instead, test cases, test procedures, test results, and related reviews are sampled during an audit. Therefore, this metric was adjusted and re-labeled to *M5* - distance from the issue to the final product (i.e., final executable code), expressed by the life cycle phase related to the issue, i.e., planning, requirements, design, coding, integration, unit testing, integrated testing, and final analyzes (e.g., coverage analysis, data and control coupling analysis, timing analysis, memory analysis). The smaller the distance, the greater the severity. The cases identified are:

a. Issue related to final analyzes, e.g., structural coverage analysis (*distance* = 1)
b. Issue related to integrated testing (*distance* = 2)
c. Issue related to unit testing (*distance* = 3)
d. Issue related to the integration phase (*distance* = 4)
e. Issue related to the coding phase (*distance* = 5)
f. Issue related to the design phase (*distance* = 6)
g. Issue related to the requirement phase (*distance* = 7)
h. Issue related to the planning phase (*distance* = 8)
i. Issue related to system level activities (distance = 9)

Justification-7: Some information is not available to the auditor by the audit time, and is more concerned to supplier project management. Moreover, the cases listed are difficult to quantify. The metric was adjusted and re-labeled as *M6* - amount of artifacts impacted, which can be estimated by the auditor during preliminary evaluation of the issue by using the following qualitative reference:

0, 1: No impact or negligible;

2, 3: Low impact, under control;

4, 5, 6: Medium impact, demanding some attention;

7, 8: High impact, raising concerns;

9, 10: Very high impact, can be unacceptable.

The next section describes the survey performed with aviation software safety specialists to obtain quantitative values for those metrics that a priori are not quantifiable (refer to table-4.4), as well as quantitative relevance of each metric.

4.4 - A Survey with aviation software safety specialists

The survey performed with aviation software safety specialists had the following objectives:

a. To obtain quantitative values for some metrics;
b. To obtain quantitative relevance of each metric;
c. To identify new metrics;
d. To identify any dependency among the metrics;
e. To obtain scores for severity of a list of issues generated from ANAC past audits.

The last objective of the survey (bullet "e") is related to the evaluation of the metrics in aeronautics and is addressed in chapter 5. The table-4.5 summarizes the information regarding the participants. The item "*Experience with international auditors*" is related to the question below:

How many software audits (Stage # 1, Stage # 2, Stage # 3, or combined stages) have you ever attended (as an auditor or audited), where there was participation of foreign authority (e.g., FAA, EASA) or international consultants as auditors?

[ ] less than 4 [ ] from 4 to 9 [ ] 10 or more

The question had the objective to capture the representativeness of the participants regarding to the experience of the main international civil aviation auditors (i.e., certification authorities and or international consultants).

Table-4.5: Summary of survey participants

Number of participants in the survey	ANAC	Industry	Total
	5	14	19
Experience with software safety (in years)	Minimum	Maximum	Average
	7	33	16.8
Experience with international auditors (see question above)	Less than 4	From 4 to 9	10 or more
	1	10	8

ANAC is among the four major civil aviation certification agencies and attended the survey with 5 specialists. The aviation industry attended with 14 specialists and is among the major world industries for transport aircraft. The participants average experience with software safety is considerably high (16.8 years), and their participation in the survey can be considered representative of the international auditor's experience (only one participant answered "less than 4"). For further details on description of the process used in the aviation survey, please refer to the appendix D. The survey results can be divided in 4 types according to the 4 objectives previously mentioned, as follow:

a. Quantitative values for those metrics that, a priori, are not quantifiable;
b. Quantitative relevance of each metric;
c. New metrics identified;
d. Discussion on dependency among the metrics.

4.4.1 - Quantitative values for the metrics

The quantitative values were obtained for four metrics: M1-"*document evaluation*", M2-"*purpose of the issue*", M3-"*artifacts impacted*" and M4-"*root cause*". Values from 0 to 3 were chosen according to the severity, being 0 for no severity and 3 for the most severe item in the metric. Each metric must had at least one item scored with 3. The survey results were very positive, as in the analysis of the scores provided by the participants there was always a tendency to converge the values. Exceptions (e.g., high deviation) were discussed in a dedicated workshop to identify possible ambiguities and unclearness that might have generated the problem. The following table-4.6 and chart present the quantitative values obtained by the survey for the metric M1. For the complete results, refer to appendix-D.

Table-4.6: Quantitative values for metric M1 “*document evaluation*”

Item#	Metric "document evaluation"	MEAN	DEVIATION
1.a	The information contains editorial errors (typos);	0.1	0.3
1.b	The information is out of context, e.g., recorded in an inappropriate section or document;	0.8	0.5
1.c	The information is inconsistent between sections or documents;	1.8	0.5
1.d	The information is confused, ambiguous;	2	0.5
1.e	The information is clear and complete, but is considered unacceptable;	2.9	0.3
1.f	The information is superficial or incomplete;	1.9	0.7
1.g	Could not find in the provided documents the required information for compliance.	2.5	0.7

Note: the enumeration format of Item# is in the context of the spreadsheet used in the survey.

By analyzing the results, it is possible to divide them in three severity levels, as follow:

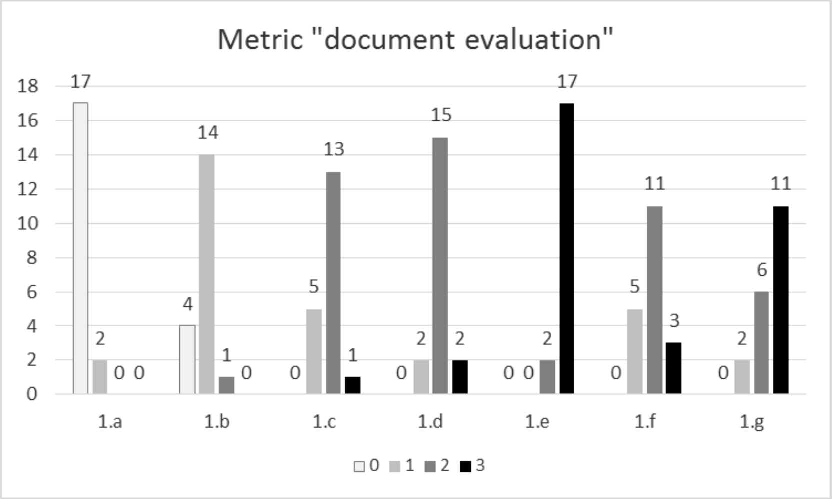
a. **Low** severity: score below 1; items 1.a and 1.b; related to editorial issues without impact in the required information;

b. **Medium** severity: score close to 2; items 1.c, 1.d and 1.f; related to the quality of the information, but without clear impact in compliance;

c. **High** severity: score close to 3; items 1.e and 1.g; clearly related to non-compliance.

The figure-4.4 provides the frequency of the scores for each item of the metric.

Figure-4.4: Chart for metric M1 “*document evaluation*”



The extreme cases almost reached consensus among the participants, i.e., for editorial issues (item 1.a) almost all participants scored zero (two exceptions), and for clear information that does not comply (item 1.e) almost all participants scored 3 (also with two exceptions). The item 1.g, though considered of high severity (average = 2.5), has high deviation, which can be explained because many auditors consider the document evaluation as sampling-based, i.e., not exhaustive. As such, it is usually opened an issue requesting the company to indicate where in the documents provided the information can be found. In that case, only if confirmed the absence of the information the issue would be related to a non-compliance (score 3), otherwise it could also be scored as 2.

4.4.2 - Quantitative relevance for each metric

Regarding the quantitative relevance of each metric, values from 0 to 3 were also chosen according to the relevance, being 0 for no relevance and 3 for the most relevant metric. At least one metric had to be scored with 3. The table-4.7 presents the result of quantitative relevance of each metric.

Table-4.7: Relevance of each metric in quantitative values

Item#	Relevance of each metric	mean	Deviation
5.a	Metric M2: purpose of the issue	2.2	0.8
5.b	Metric M3: type of artifact impacted by the issue	1.8	0.8
5.c	Metric M4: root cause of the issue	2.2	0.7
5.d	Metric M5: distance from the issue to the final product	1.6	0.8
5.e	Metric M6: amount of artifacts impacted by the issue	2.3	0.7
5.f	Metric M7: adequacy of the issue in regard to the stage of the audit	1.6	0.9

Note: the enumeration format of Item# is in the context of the spreadsheet used in the survey.

There is not much difference among the metrics for the quantitative relevance, and all of them had values close to 2. Nevertheless, it is possible to classify them in two levels of relevance:

a. **Medium-high** relevance: score above 2; items 5.a, 5.c and 5.e; related to the essence of the issue (What for? Why it happened? How much damage it caused?);

b. **Medium** relevance: score below 2; items 5.b, 5.d and 5.f; related to process and life cycle data (How far from the mainstream data? How far from the final data? How delayed from the current process?).

Note: The deviation is higher than those cases of quantitative values for each metric.

4.4.3 - Identification of new metrics

Six new metrics were suggested in the survey, together with the score for the quantitative relevance. Those suggested metrics were presented and discussed for acceptance in a dedicated workshop with the survey participants. The table-4.8 presents a summary of the suggested metrics, the proposed score for quantitative relevance, whether accepted by the participants and justification.

Table-4.8: Suggested metrics and evaluation result during workshop

Item #	Suggested metric	Score	Accepted ?	Justification
6.a	Service history of the previous product generated by the same process being audited	3	NO	More applicable to define initial level of involvement in oversight. Difficult to obtain such information if the previous product belongs to another company
6.b	Criticality of the software for flight safety	3	NO	Already being captured by the software criticality level, which is included in the decision table of the metric
6.c	Deadline for certification (the further the deadline is, the lower the impact, and vice-verse	2	NO	This concern is of managerial scope and should not affect the analysis of the audit result
6.d	Estimated rework	2	NO	Information difficult to obtain by the auditor during the audit. Partially captured by the metric “amount of artifacts impacted by the issue”
6.e	Number of systemic deviations identified	2	NO	Captured by the combination of the metrics “purpose of the issue” and “root cause of the issue”
6.f	Severity and difficulty of solution of the issue identified	2	NO	Metric is too open and difficult to quantify. Partially captured by “root cause” and “amount of artifacts impacted by the issue”

Note: the enumeration format of Item# is in the context of the spreadsheet used in the survey.

After presenting the suggested metrics and discussing about their merits, none of them were accepted. In some cases, the metrics were not applicable to the scope of the oversight stages (i.e., 6.a and 6.c), or they were already captured by a combination of other metrics (i.e., 6.b, 6.d, 6.e and 6.f), or too difficult to obtain the necessary information (i.e., 6.a and 6.d), or the criterion was too open and difficult to quantify (i.e., 6.f).

4.4.4 - Discussion on dependency among metrics

During the workshop, some discussion took place on dependency among the metrics, and the summary is as follows:

a. The metric M_1 stands by its own and is applicable to Stage#1, which is specific for document evaluation. All other metrics are applicable to Stage#2 and Stage#3;

b. The metric M_2 - "*purpose of the issue*" set the basic measurement for each type of issues. The other metrics (M_3 to M_7) may affect the basic measurement, but inside its range, i.e., never extrapolating the maximum value;

c. If the issue does not relate to a non-compliance (e.g., metric M_2, items 2.a and 2.e), then some of other metrics may not be applicable to that issue (e.g., metric M_4 – root cause);

d. There is some dependency between the metric M_3 (type of issue) and M_5 (distance to the final product). For example, in M_3 the item 3.a "issue opened against plans and standards" implicitly considers the distance between the planning phase and the final executable code, which is part of the metric M_5. Similar with the item 3.c (verification cases and procedures), which is an artifact applied during the verification of the executable code, implicitly including the metric M_5;

e. Some suspicions arouse about possible dependency between the metric M_2 (type of artifact) and M_5 (amount of artifacts impacted), but after further discussion it was agreed that they are distinct metrics without any overlaps.

f. There was consensus among the participants that the severity level cannot be fully captured by the metrics, i.e., it is not possible to have a complete set of criteria that quantitatively covers all aspects of the severity level of an issue. Therefore, a percentage should be left to the auditor subjectivity, based on his or her "engineering judgment".

4.5 -The metrics equations

This section describes the generation of the equations that express the metrics, which are composed of the metrics M_1 to M_7. Based on the section 4.4.4, discussion on dependency among the metrics, the metric equation is two-folded as follow:

a. An equation specific for documents evaluation, which uses the metric M_1.

b. An equation for process evaluation and process adherence assessment, which uses the metrics M_2 to M_7,

4.5.1 - The metric equations for documents evaluation

According to the workshop discussion on dependency among metrics, "*the metric M1 stands by its own and is applicable to Stage#1, which is specific for document evaluation. All other metrics are applicable to Stage#2 and Stage#3".* For convenience, first the values obtained in the survey (table-4.6) were changed to the range between 0 and 10, as follow:

$$measurement = \frac{\text{SurveyValue}}{\text{MaximumValue}} * 10 \qquad (4.1)$$

Where: MaximumValue = 2.9

The adjusted table in ascending order is presented in table-4.9:

Table-4.9: Quantitative values for metric M1 "*document evaluation*"

Case	measure	Description	Survey value
1	0.3	The information contains editorial errors (typos)	0.1
2	2.8	The information is out of context, i.e., recorded in an inappropriate section or document	0.8
3	6.2	The information is inconsistent between sections or documents	1.8
4	6.6	The information is superficial or incomplete	1.9
5	6.9	The information is confused, ambiguous	2.0
6	8.6	Could not find in the provided documents the required information for compliance	2.5
7	10.0	The information is clear and complete, but is considered unacceptable	2.9

The equation for the measurement of each stage is:

$$DocEvalMeasur = \sum_{i=1}^{n} \text{m(i)} \qquad (4.2)$$

Where:

DocEvalMeasur: the measurement for each audit stage focusing on documents evaluation,

m(i): the measurement for each issue i, by applying the M1 metric (table-4.9),

i=1…n, n: total amount of audit issues

4.5.2 - The metric equations for process evaluation and process adherence assessment

According to the workshop discussion on dependency among metrics, "*the metric M2 - purpose of the issue, set the basic measurement for each type of issues. The other metrics (M3 to M7) may affect the basic measurement, but inside its range, i.e., never extrapolating the maximum value"*. In other words, M2 indicates the maximum possible severity of an issue in percentage, according to the purpose. No matter how high is the impact of the metrics M3 to M7 for that specific issue, the final measurement for that issue will reach at most the percentage defined by the metric M2. The table-4.10 presents the percentage values for the metric M2 in ascending order:

Table-4.10: Percentage values for metric M2 "purpose of the issue"

Case	%	Description	Survey mean
1	7	A suggestion for process improvement detected during the audit. However, the process is considered sufficient for compliance	0.2
2	25	An issue to request additional information, but a priori without any impact in concerns discussed during the audit	0.7
3	47	An issue to correct a punctual process deficiency (or adherence to the process) detected during the audit	1.3
4	75	An issue to request additional information, which may drive to a non-compliance that was not conclusive during the audit	2.1
5	100	An issue to record a non-compliance and request a closure approach	2.8

For the metrics M3 to M7, first the values obtained in the survey were changed to the range between 0 and 10, by applying the equation-4.1. The table-4.11 presents the result in ascending order for the metric M3:

Table-4.11: Quantitative values for metric M3 "*type of artifact impacted*"

Case	measure	Description	Survey mean
1	1.2	Issue opened against informal data (e.g., an SQA spreadsheet for informal control not planned for use by the process)	0.3
2	6.2	Issue opened against Software Quality Assurance (SQA) Records	1.6
3	6.5	Issue opened against Software Configuration Management Records	1.7
4	6.9	Issue opened against plans and standards	1.8
5	6.9	Issue opened against Problem Reports (PR)	1.8

6	7.3	Issue opened against tools (e.g., poor qualification report, justification for non-qualification is unacceptable)	1.9
7	7.7	Issue opened against traceability (e.g., requirement points to wrong parent requirement, insufficient granularity)	2.0
8	9.2	Issue opened against verification data, including reviews, inspections, verification cases and procedures, verification results and related artifacts	2.4
9	10.0	Issue opened against requirement, design, code or configuration data (e.g., ambiguous requirement, architecture incompatible with requirements, code does not fully implement the requirement	2.6

According to the workshop discussion on dependency among metrics (ref. 4.4.4, bullet d), *"there are some dependency between the metric M3 (type of issue) and M5 (distance to the final product). [...] with the item 3.c (verification cases and procedures), which is an artifact applied during the verification of the executable code, implicitly including the metric M5."* As a consequence, in the appendix-D, table-D.3, items 3.c and 3.d were merged into the case 8 of the table-4.11.

The adjusted values for the metric M4 "root cause of the issue" are presented in the table-4.12:

Table-4.12: Quantitative values for metric M4 "*root cause*"

Case	**measure**	**Description**	**Survey mean**
1	2.0	It was only a slip, an isolated case	0.6
2	5.9	Similar cases have been found involving the same person, raising suspicion of insufficient training	1.7
3	6.2	The training material was deficient, raising suspicion that the person did not understand enough the activity to perform	1.8
4	6.6	The amount and complexity of the information needed for the activity may have contributed to the mistake	1.9
5	6.9	The process followed was not clear, which may have contributed to the mistake	2.0
6	10.0	The process followed was clear but incorrect, leading the person to the mistake	2.9
7	6.2	Default value for the case where the root cause cannot be determined at the time the issue is raised (measurement = average of all cases)	N/A
8	0.0	The issue is not related to (potential) non-compliance regarding process adherence. Therefore, the root cause is not applicable.	N/A

Remark: For the case 7, it is assumed the average value as default, which may change during the issue follow-up, once the root cause is identified after further investigation.

The metric M5 is related to the distance from the issue to the final product (i.e., final executable code), expressed by the life cycle phase related to the issue. The smaller the distance, the greater the severity. The table-4.13 presents the adjusted values and is based on a life cycle process with the following phases: system level, planning, requirements, design, coding, integration, unit testing, integrated testing, and final analyzes (e.g., coverage, data and control coupling, timing, memory). For the measurement, it is assumed that the relevance is inversely proportional to the distance.

Table-4.13: Quantitative values for metric M5 "*distance to the final product*"

Case	measure	Description	Distance
1	1.1	Issue related to system level phases	9
2	2.2	Issue related to planning phase	8
3	3.3	Issue related to requirements phase	7
4	4.4	Issue related to design phase	6
5	5.5	Issue related to coding phase	5
6	6.6	Issue related to integration phase	4
7	7.7	Issue related to unit testing	3
8	8.8	Issue related to integrated testing	2
9	10.0	Issue related to final analysis	1
10	5.5	Issue related to most of or all phases	N/A

The case 10 captures those issues that impact or are applicable to various phases, for example, some deficiencies in SQA process or configuration control. The life cycle phases are based on the aviation standard DO-178C. For the case of space domain, the life cycle phases may be different, resulting in different distances and related measurements. Any metric adjustments for space domain are discussed in chapter 6.

For the metric M6, "*amount of artifacts impacted by the issue*", the measurement is estimated by the auditor during preliminary evaluation of the issue, which also considers potential impacts, i.e., throughout the audit follow-up the measurement may change. The table-4.14 provides the values for the metric M6:

Table-4.14: Quantitative values for metric M6 “*amount of artifacts impacted by the issue*”

Case	measure	Description
1	**0**, **1**	No impact or negligible
2	**2**, **3**	Low impact, under control
3	**4**, **5**, **6**	Medium impact, demanding some attention
4	**7**, **8**	High impact, raising concerns
5	**9**, **10**	Very high impact, can be unacceptable
6	**5**	Default value, requiring further Company investigation

The default case is used when it is not possible to estimate at the time the issue is raised, and depends on further company investigation. It is assumed the average value as default, which may change during the issue follow-up.

The table-4.15 provides values for the metric M7, “*adequacy of the issue regarding to the audit stage*”. For the measurement, it is assumed that the relevance is directly proportional to the adequacy.

Table-4.15: Quantitative values for metric M7 “*adequacy of issue regarding to audit stage*”

Case	Measure	Description	Adequacy
1	0.0	Issue identified in adequate audit Stage	0
2	3.3	Stage#1 scope issue identified in Stage#2 Stage#2 scope issue identified in Stage#3 Stage#3 scope issue identified in Stage#4	1
3	6.6	Stage#1 scope issue identified in Stage#3 Stage#2 scope issue identified in Stage#4	2
4	10.0	Stage#1 scope issue identified in Stage#4	3

The number of audit stages is based on the aviation model, which consist of four stages. For the case of space domain, the number of audit stages may be different, resulting in different adequacies and related measurements. Any metric adjustments for space domain are discussed in chapter 6.

For the severity calculation of each audit issue, it is also necessary to obtain the relevance of each metric involved. The quantitative relevance of each metric in percentage was obtained as follow:

$$Wj = \frac{\mathrm{Mj}}{\mathrm{Mt}} * 100 \qquad (4.3)$$

Where:

Wj: the quantitative relevance of the metric j in percentage (weight)

Mj: the relevance of the metric j in survey score

Mt: the total sum of survey scores (Mt = M3 + M4 + M5 + M6 + M7)

j = 3..7

The table-4.16 provides the relevance of each metric in percentage (weight).

Table-4.16: The relevance of each metric in percentage

Metric	W *	Description	Survey mean
M2	N/A **	Purpose of the issue	2.2
M3	19	Type of artifact impacted by the issue	1.8
M4	23	Root cause of the issue	2.2
M5	17	Distance from the issue to the final product	1.6
M6	24	Amount of artifacts impacted by the issue	2.3
M7	17	Adequacy of the issue regarding to the audit stage	1.6

Note *: The weight of the metric relevance in percentage

** The metric M2 set the basic measurement for each type of issues. Therefore, although the scores have been obtained in the survey, the relevance in weight is not applicable for the equation that expresses the final metric calculation.

The final calculation was divided in two equations. The first is an equation to measure the severity of each issue, as follow:

$$m = M(2) * \sum_{j=3}^{7} (\mathrm{M(j)} * \mathrm{W}(j)) \qquad (4.4)$$

Where:

m: the measure of the issue severity

M(2): the percentage related to the purpose of the issue (refer to table-4.10)

M(j) the measurement for the metric Mj (refer to table-4.11 to table-4.15)

W(j): the percent relevance of each metric (refer to table-4.16)

The second is an equation to calculate the final measurement of the audit result, as follow:

$$mAudit = \sum_{i=1}^{n} \mathrm{m(i)} \tag{4.5}$$

Where:

mAudit: the final measurement of the audit result

m(i): the measurement of the issue i, by applying the equation-4.4 for each issue

n: total amount of issues recorded in the audit

The final measurement of the audit result can be applied to a decision table to supporting the managerial decision for next steps. Examples are provided in chapter 7, section 7.6.

4.6 – Summary of chapter 4

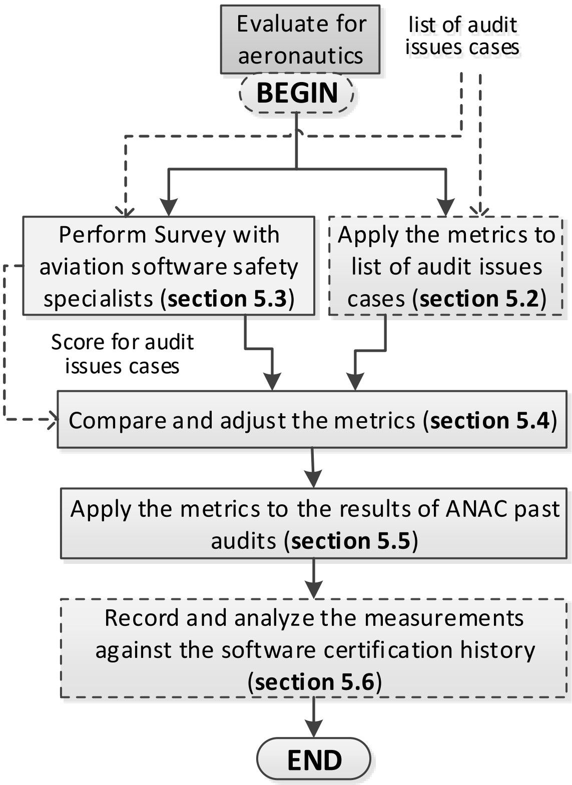
This chapter described the generation of the Aerospace Metrics. First, the metrics were generated analytically by using the consolidated technique GQM and Reason´s human error model. Then, the analytical metrics were refined by using ANAC past audits results, and quantitative values were obtained by a survey with civil aviation software safety specialists. At the end, the metrics equations were specified.

5. THE METRICS EVALUATION FOR AERONAUTICS

5.1 - Overview

This chapter is mainly related to the design cycle of the Design Science. In this chapter, the metrics generated in chapter 4 are applied to a representative set of past aviation software audits and the resultant measurement is evaluated against the software certification history. Surveys and workshops with software safety senior specialists from aviation are also used. The figure-5.1 shows the metric evaluation process for the aeronautics, whose general context is illustrated in the figure-1.1:

Figure-5.1: The metrics evaluation process for aeronautics


Evaluate for aeronautics
list of audit issues cases
BEGIN
Perform Survey with aviation software safety specialists (section 5.3)
Apply the metrics to list of audit issues cases (section 5.2)
Score for audit issues cases
Compare and adjust the metrics (section 5.4)
Apply the metrics to the results of ANAC past audits (section 5.5)
Record and analyze the measurements against the software certification history (section 5.6)
END

Apply the metrics to list of audit issues cases: Generate a list of audit issues with summarized description (based on ANAC past audits), which can exercise the metrics by covering a representative set of audit issues cases. Apply the metrics to the generated list and obtain the measurements that are compared with the result of the survey.

Perform survey with aviation software safety specialists: The same list of issues generated from the results of ANAC past audits is submitted in a survey with software safety senior specialists from the civil aviation to obtaining scores for the issues severity, based on their experience on performing software audits.

Compare and adjust the metrics: The measurements obtained from the metrics are evaluated against the result of the survey (i.e., senior specialists' scores for the issues) to identifying adjustments in the metrics equation. The survey result is used as reference for tuning the metrics equations.

Apply the metrics to the results of ANAC audits: Some software are selected taking into account their representativeness regarding the certification history. The adjusted metrics are applied to the issues of audits performed on those selected software.

Record and analyze the measurements against the software certification history: The measurement obtained from those selected software is evaluated against their certification history. For example, for an audit that has failed it is expected a measurement reflecting a bad result and vice-verse, i.e., for an audit that has passed with merit and has justified the lowering of ANAC involvement it is expected a measurement reflecting a very good result. Those evaluated cases can support building a table to be used for management decision.

5.2 - Generation of list of representative audit issues and submission to the metrics

It was generated a list containing description of audit issues based on issues identified in past software audits performed by ANAC. The list was divided in three groups according to the audit stage: Stage#1, Stage#2 and Stage#3. Care was taken to build a list that was representative of the cases usually recorded and, as much as possible, attempted to cover the metrics generated in chapter 4. For the complete list of the issues refer to the Appendix-D, table-D.6.

The table-5.1 shows the mapping of the generated list of issues against the metrics, in order to have an idea of the coverage, and consequent representativeness of the list for metrics evaluation. For an example of metrics coverage by an issue, please refer to chapter 7, section 7.4.

Table-5.1: The coverage of the metrics by the generated list of audit issues

Stage ▼	metric ▶ issue ▼	M1	M2	M3	M4	M5	M6	M7
Stage#1	1.a	●						
	1.b	●						
	1.c	●						
	1.d	●						
	1.e	●						
	1.f	●						
	1.g	●						
Stage#2	2.a		●	●		●	●	●
	2.b		●	●		●	●	●
	2.c		●	●	●	●	●	●
	2.d		●	●	●	●	●	●
	2.e		●	●		●	●	●
	2.f		●	●	●	●	●	●
	2.g		●	●	●	●	●	●
	2.h		●	●	●	●	●	●
	2.i		●	●				●
Stage#3	3.a		●	●		●	●	●
	3.b		●	●	●	●	●	●
	3.c		●	●	●	●	●	●
	3.d		●	●		●	●	●
	3.e		●	●	●	●	●	●
	3.f		●	●	●	●	●	●
	3.g		●	●	●	●	●	●
	3.h		●	●	●	●	●	●
	3.i		●	●	●	●	●	●

The Stage#1 has 7 issues (1.a to 1.g) and they are all mapped to the metric M1 because in the civil aviation the first process is the planning whose outputs are the planning documents; therefore, the audit stage is basically composed of documents evaluation. Although not shown in the table-5.1, it is important to mention that all the M1 cases (see Table-4.9) have been covered by the Stage#1 issues to ensuring the representativeness of the list regarding to exercising the metrics. For the Stage#2 and Stage#3 where both have 9 issues each (2.a to 2.i, 3.a to 3.i), the metrics almost had full coverage, with few exceptions. Most cases of non-coverage are in the M4 (root cause) because it does not make sense to request the software supplier to identify the root cause for issues whose purpose is not related to non-compliance.

The measurements of the whole list of issues is provided in section 5.4 and compared with the scores obtained from the survey with software safety senior specialists. The comparison can provide subsidies for adjustments in the metrics calculation.

5.3 - A survey with aviation software safety specialists

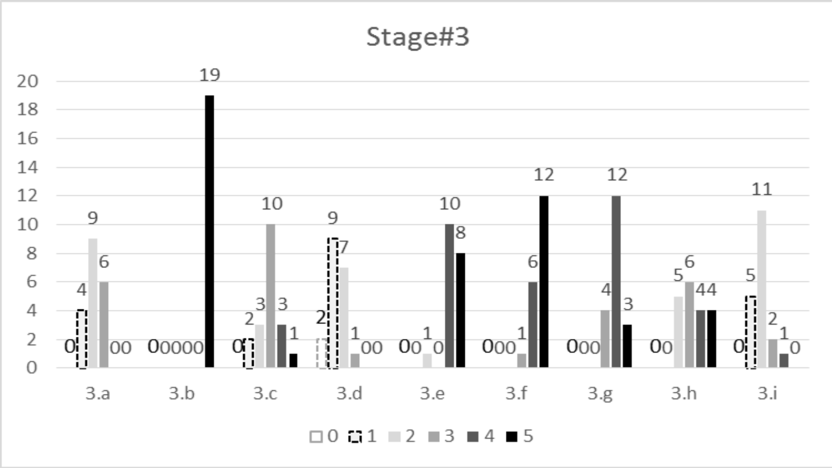
The same list of issues generated in section 5.2 was applied to the survey with software safety senior specialists to obtaining scores for the issues severity based on their experience with civil aviation software audits. The following was used as qualitative reference for score assignment:

0: no severity;
1: very low severity, negligible;
2: low severity, summarized follow-up is enough;
3: medium severity, detailed follow-up needed;
4: high severity, demanding attention;
5: very high severity, requiring priority follow-up

The survey results were collected and analyzed to identifying possible unclear or ambiguous instructions that may have led to misunderstandings, with consequence prejudice to the survey results. Those cases were addressed in a dedicated workshop with the survey participants. Care was taken to focus in clarifying the instructions and not to influence the participants in revising the score. The detailed description of the process used in the survey as well as the complete result is provided in appendix-D.

The following descriptions are samplings of the survey result prior to the workshop, with the purpose of explaining the role of the survey in the metrics evaluation for aeronautics. The figure 5.2 shows the mean, deviation, and distribution graphic of survey scores from Stage#3 scope:

Figure-5.2: Distribution for Stage#3 survey scores prior to the workshop



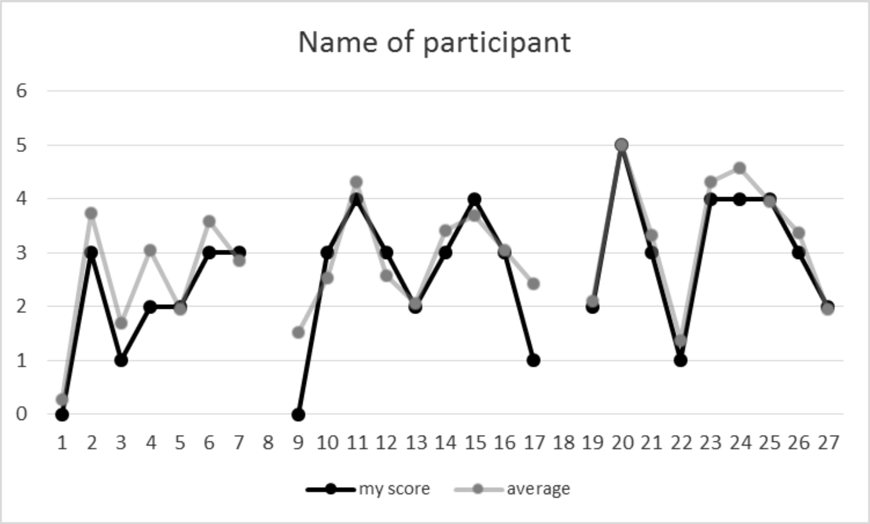
Stage#3	mean	deviation
3.a	2,1	0,7
3.b	5	0
3.c	2,9	1
3.d	1,4	0,8
3.e	4,3	0,7
3.f	4,6	0,6
3.g	3,9	0,6
3.h	3,4	1,1
3.i	1,9	0,8

It can be noticed that item 3.b obtained unanimity in the score (score = 5, very high severity, requiring priority follow-up). The item 3.c, despite the high deviation, still shows a tendency to converge, but the same does not happen with item 3.h that does not indicate any tendency. This last one was selected for workshop discussion. (Ambiguous text? Misaligned concepts?

Controversial points?). One can perceive that in general the deviation is quite high, and assuming that the issues were described clearly (with few exceptions), this suggests that there is some subjectivity in the issues evaluation among the auditors, and the metrics would have the potential to mitigate such subjectivity.

The next descriptions focus on the performance of some participants in comparison to the average of the group of participants. The figure-5.3 shows a case of scores very close to the group average:

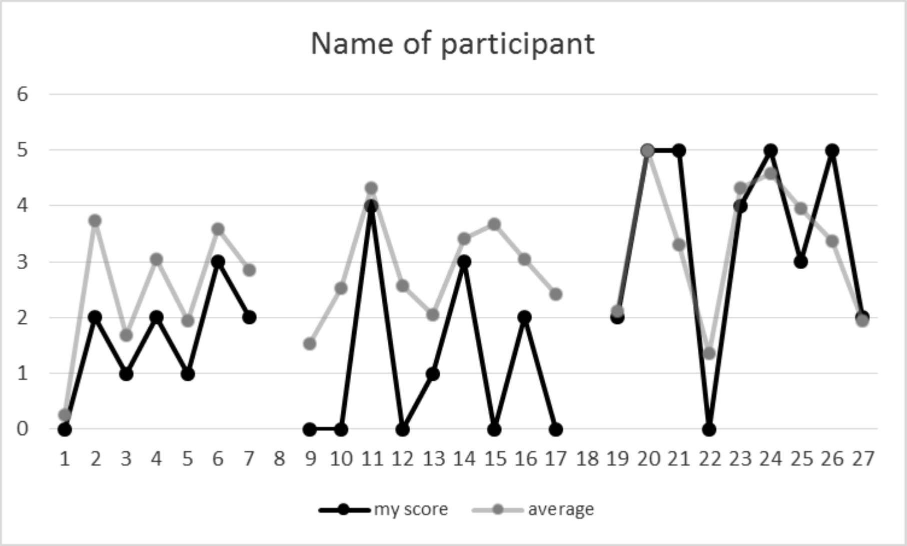
Figure-5.3: Case of scores close to the average



The specific participant's scores are represented in black, while the group average scores are in grey. The three groups of segments represent the three audit stages, i.e., Stage#1 (abscissa 1 to 7), Stage#2 (abscissa 9 to 17) and Stage#3 (abscissa 19 to 27). The Stage#4 was not included because ANAC has not formally performed any Stage#4 so far due to the stage scope. Considering that the score is always an integer value (i.e., 0 to 5), whereas the average can be fractional, almost all scores are inside the average, except the abscissas 9 and 17, which are both one unit below. Moreover, one can notice that the scores of the specific participant follow the group tendency, i.e., both lines are synchronized in ascending and descending sequence.

The figure-5.4 shows a case where the participant's scores are far from the average, and with tendency to less rigor.

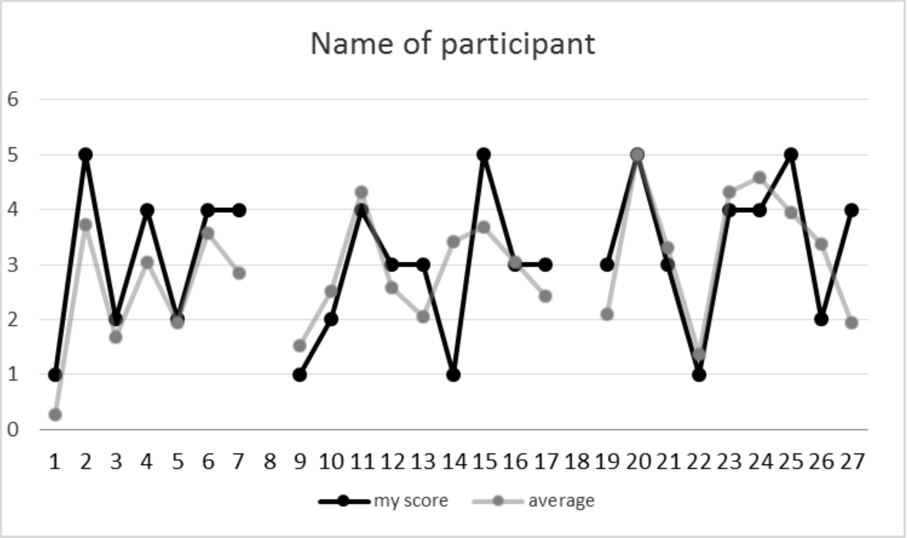
Figure-5.4: Case of scores showing tendency to less rigor



The scores that are far from the average are all below it (i.e., abscissas 2, 9, 10, 12, 15, 17, 21, 22, except 26), which show a tendency to less rigor than the average of participants. Nevertheless, the scores follow the average tendency (i.e., both lines are synchronized) with few exceptions.

The two cases presented so far, i.e., close to average and tendency to less rigor, are both normal cases expected in any survey, and do not invalidate the survey result. The next two cases are examples that questioned the survey result and demanded some analysis and adjustments during the workshop. The figure-5.5 shows a case of participant's scores very close to the group average, but with a specific score very far from the average.

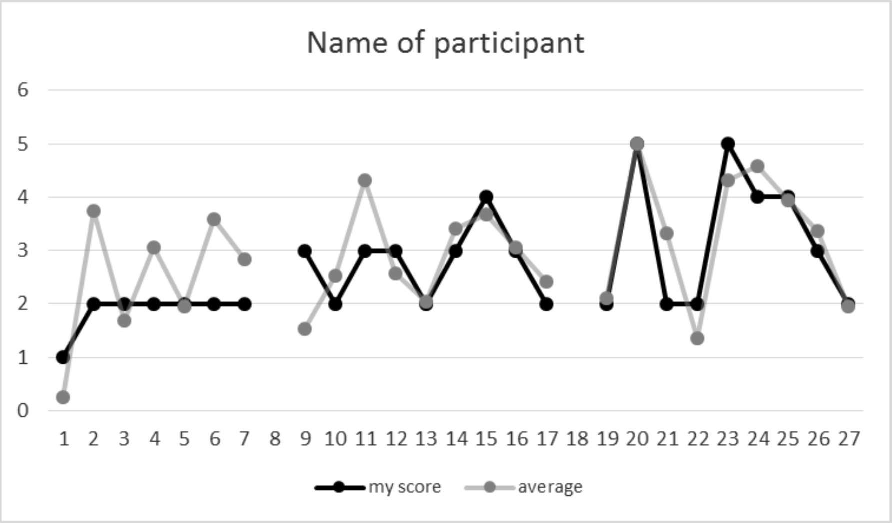
Figure-5.5: Case of scores close to the average, but with one score very distant



Almost all scores are inside the average or very close to it, and both lines are synchronized. However, one specific score (i.e., abscissa 14) is two units below the average and in opposition to the average tendency, i.e., the first is descending whereas the latter is ascending. Is it a case where the participant misunderstood the issue? Or does the participant have a peculiar interpretation of this issue severity? This case was selected for workshop discussion.

The figure-5.6 shows a case of a participant assigning scores with fixed values during an interval, without following the average tendency.

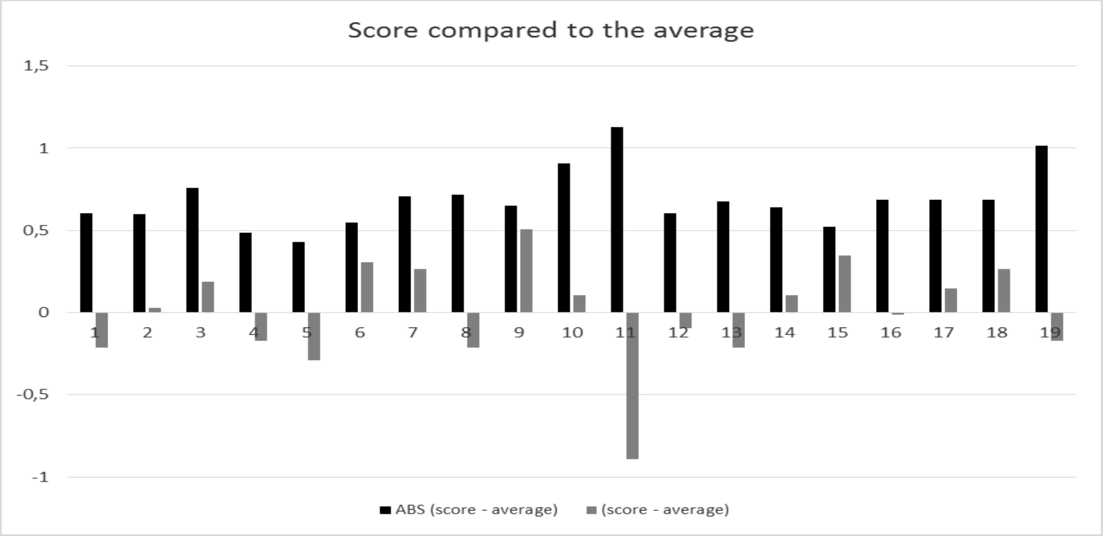
Figure-5.6: Case of scores with fixed values, not following the average tendency



Similar to the previous case, with the exception of the mentioned interval, almost all scores are inside the average or very close to it and both lines are synchronized. In the specific interval between abscissas 2 and 7, there are 6 sequential fixed scores equal 2, not following the average tendency at all. Is it a case of misunderstanding the instructions for the Stage#1 group? Or does the participant have a fixed criterion for this interval? A mind set? This case was also selected for workshop discussion.

The figure-5.7 shows the overall performance of the participants in comparison to the group average.

Figure-5.7: Participants performance comparing to the average



The vertical bars in black represent the absolute mean difference between the participant's score and the group average. In other words, it measures the average distance between the participant's scores and the group scores average. The vertical grey bars take into account the signal, i.e., the average difference between the participant's score and the group score average can be either positive or negative, which indicates the tendency to more rigor (i.e., positive grey bar) or less rigor (i.e., negative grey bar). The participant 5 is the closest to the group average, the participant 9 has the tendency to be more rigorous, the participant 11 to be less rigorous, and the participant 19 is one of the most distant from the average, but without any tendency (i.e., grey bar close to zero).

A major contribution of the survey that was not originally planned was to serve as a tool for the self-assessment of the software safety specialist and for the alignment of concepts and rigor among the specialists (auditors). It has been also studied the possibility of applying the survey within aviation industry. In this case, the survey would also be used as a tool to aid in the training of future software auditors.

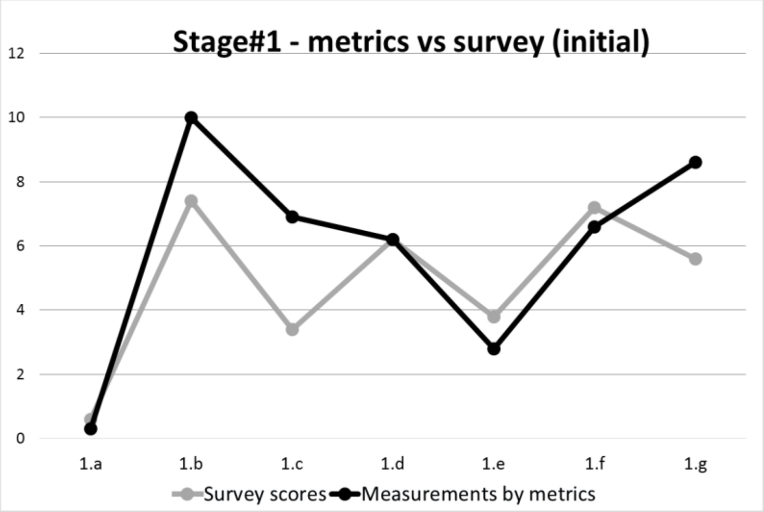
5.4 - Compare and adjust the metrics

This section compares the measurements generated by the metrics (see section 5.2) against the scores from the survey (see section 5.3), to identifying necessity for adjustments in the metrics.

5.4.1 - Metrics related to documents evaluation

The figure-5.8 shows both, the measurements obtained by applying the metrics and the survey scores for the Stage#1 issues:

Figure-5.8: Measurements and survey scores for Stage#1 issues



For the cases 1b, 1c and 1g, the resultant measurements are far from the survey scores. More specifically, in all those cases the measurements are much higher. Considering the survey scores as reference, those cases should be evaluated for necessity of adjustments in the metrics. According to the metric M1 (see table-4.11), the case 1b was measured with the value 10, “*The information is clear and complete, but is considered unacceptable”,* which is the maximum possible value. That means all M1 measurements assume the cases are related to information of high technical relevance, because all other measurements listed in table-4.9 took as reference the highest value 10. However, in the survey there are cases related to information with varieties of technical relevance, which the software safety specialists must probab (5.1) considered for assigning the score. For instance, the case 1b describes a non-compliance related to model coverage, which can be considered of medium technical relevance, but the metric M1 assigned the maximum severity value to it. Similar situation applies to the cases 1c and 1g, which are also related to information of medium technical relevance. As an adjustment in the metrics, all items of Stage#1 scope should have an additional consideration regarding the technical relevance of the information associated to the issue, which is based on qualitative judgment of the auditor. The equation-4.2 was changed as follows:

$$DocEvalMeasur = \sum_{i=1}^{n} \mathrm{m(i)} * \mathrm{R(i)} \qquad (5.1)$$

Where:

DocEvalMeasur: the final measurement for audits focusing on documents evaluation;

m(i): the measurement for each issue “i”, by applying the table-4.9 to be adjusted (see table-5.2);

R(i): the technical relevance of the information related to each issue “i”, which is based on qualitative judgment of the auditor (see table-5.3 for possible values);

i=1…n, n: total amount of audit issues

The possible values for R(i), i.e., the technical relevance of the information related to each issue “i", can be estimated by analyzing the data from the figure-5.8. By equation-5.1, the measurement of each issue “i” is calculated by:

$$\mathrm{IssueMeasurem(i)} = \mathrm{m(i)} * \mathrm{R(i)}$$

And the technical relevance is as follows:

$$R(i) = \frac{\mathrm{IssueMeasurem(i)}}{\mathrm{m(i)}} \qquad (5.2)$$

The IssueMeasurem(i) should be coherent with the survey scores because it is supposed to capture the technical relevance of the information related to the issue, in addition to the M1 cases from the table-5.2.

As explained, the cases 1b, 1c, and 1g are related to information with medium technical relevance. Assuming that the survey score for those cases are acceptable approximation of the issue measurement, and with the medium technical relevance as R=1, then by equation-5.2:

$$\mathrm{IssueMeasurem(i)} = \mathrm{m(i)}, \quad \mathrm{i=1b,\ 1c,\ 1g}$$

Which means the survey score can represent the M1 measurements for the cases 1b, 1c, and 1g, but adjusted to the related information of medium technical relevance. As the M1 measurements of table-4.9 reflect the maximum technical relevance of the information, the maximum value for R(i) for the case 1b can be calculated by:

$$\text{MaximumR}(1\text{b}) = \frac{\text{M1measure}(1\text{b})}{\text{SurveyScore}(1\text{b})} = \frac{10}{7.4} = 1.35$$

Applying the same logic to cases 1c and 1g, and calculating the average, the result is:

$$\text{MaximumR} = \frac{1.35 + 2.03 + 1.54}{3} = 1.64$$

The table-4.9 should be adjusted, assuming that the issue case is related to information of medium technical relevance. The factor for M1 adjustment from table-4.9 is:

$$\text{M1adjust} = \frac{1}{\text{MaximumR}} = 0.61$$

The table-5.2 shows the adjusted measurements:

Table-5.2: Adjusted quantitative values for metric M1 "*document evaluation*"

Case	Measure	Description	Survey value
1	0.2	The information contains editorial errors (typos)	0.1
2	1.7	The information is out of context, i.e., recorded in an inappropriate section or document	0.8
3	3.8	The information is inconsistent between sections or documents	1.8
4	4.0	The information is superficial or incomplete	1.9
5	4.2	The information is confused, ambiguous	2.0
6	5.2	Could not find in the provided documents the required information for compliance	2.5
7	6.1	The information is clear and complete, but is considered unacceptable	2.9

The measurement from the table-5.2 is assigned to the factor m(i) from the equation-5.1, and represents the possible cases of an issue opened during a document evaluation, but without considering the technical relevance of the related information.

The table-5.3 shows the possible values for the technical relevance of the information related to a Stage#1 issue, the qualitative meaning, and the related measurement to be assigned to the R(i) in the equation-5.1:

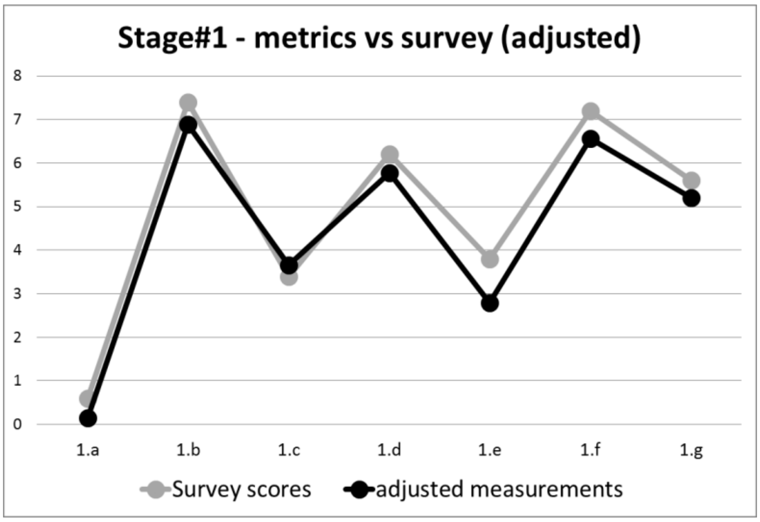
Table-5.3: The technical relevance (R) of the information related to a Stage#1 issue

Measure	Qualitative technical relevance	Relevance
0.36	Very low or negligible	0
0.48		1
0.61	Low	2
0.74		3
0.87	Medium	4
1.00		5
1.13		6
1.26	High	7
1,39		8
1.52	Very high	9
1.64		10

For the medium technical relevance it was assigned the value 5 with the related measurement equal 1, which reflects the measurements of the table-5.2. The lowest relevance was obtained by symmetry from the highest in relation to the medium relevance. And other values in between were calculated linearly.

After applying the adjusted M1 metrics, the figure-5.9 shows the new measurements in comparison with the survey scores:

Figure-5.9: New measurements and survey scores for Stage#1 issues

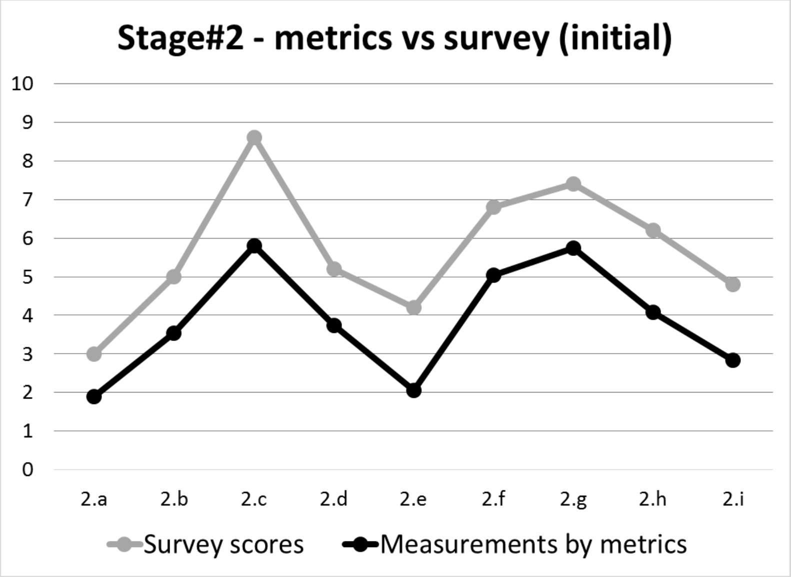


Comparing to the initial measurement shown in figure-5.10, the new measurement is much closer to the scores obtained from the survey, because the adjusted M1 metrics now also consider the technical relevance of the information related to the issue. The adjusted M1 metrics are composed of the table-5.2 instead of table-4.9, the new table-5.3, and the equation 5.1 instead of equation 4.2.

5.4.2 - Metrics related to process evaluation and process adherence assessment

The figure-5.10 shows both, the measurements obtained by applying the metrics and the survey scores for the Stage#2 issues:

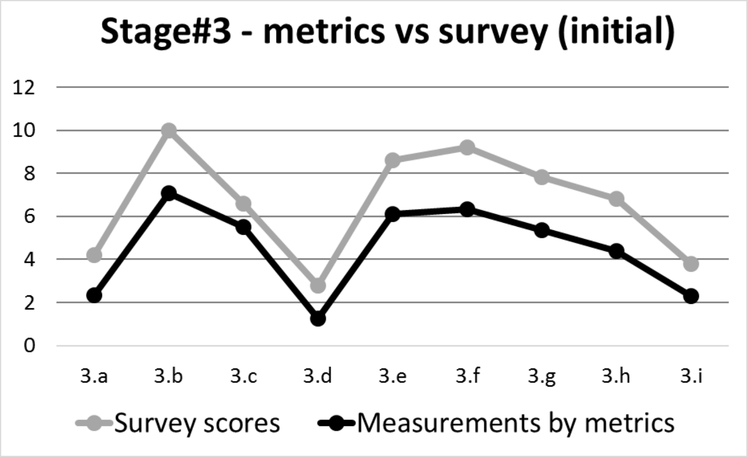
Figure-5.10: Measurements and survey scores for Stage#2 issues



The measurements obtained from applying the metrics follow the same tendency of the survey scores, but all values are lower than the survey scores, and the average difference is equal 1.83. By analyzing the applicability of the metrics, it is detected that for the metric M7 (see table-4.15) the case applied was always 1, *"Issue identified in adequate audit Stage",* with the measurement equal zero. Considering that the metric M7 accounts for 17% of the total issue measurement, this fact contributes to reduce the resultant measurement. Moreover, by analyzing the issues description and related survey scores, it is possible to conclude that the evaluation of the software safety specialists does not reduce the severity of the issue if it is identified in adequate audit stage, i.e., an issue could be evaluated with the highest score even if identified in adequate audit stage.

The figure-5.11 shows both, the measurements obtained by applying the metrics and the survey scores for the Stage#3 issues:

Figure-5.11: Measurements and survey scores for Stage#3 issues



Similar to the Stage#2 issues, the measurements obtained from applying the metrics follow the same tendency of the survey scores. All values are lower than the survey scores, and the average difference is equal 2.13. The M7 metric analysis for the Stage#2 is also applicable for the Stage#3 issues, though the case 3e had M7 measurement different from zero.

As an adjustment in the metrics, the M7 does not contribute directly to the measurement calculation. Instead, it is an additional consideration that may increase the severity. The equation-5.3, which changes the equation-4.4, shows the expression that calculates the measurement for each issue:

$$m = \left(M(2) * \sum_{j=3}^{6} W(j) * M(j) \right) * M(7) \qquad (5.3)$$

Where:

m: the measure of the issue severity

W(j): the percent relevance of each issue by applying the table-4.16 to be adjusted (refer to table-5.4);

M(j) the measurement for the metric Mj (refer to table-4.11 to table-4.14);

M(2): the percentage related to the purpose of the issue (refer to table-4.10)

M(7): the measurement for the metric M7 by applying the table-4.15 to be adjusted (refer to table-5.5).

The table-5.4 shows the adjusted table-4.16, with the new percent relevance of each issue:

Table-5.4: The adjusted relevance of each metric in percentage

Metric	W *	Description	Survey mean
M2	N/A **	Purpose of the issue	2.2
M3	23	Type of artifact impacted by the issue	1.8
M4	28	Root cause of the issue	2.2
M5	20	Distance from the issue to the final product	1.6
M6	29	Amount of artifacts impacted by the issue	2.3
M7	20***	Adequacy of the issue regarding to the audit stage	1.6

Note *: The adjusted weight of the metric relevance in percentage

**: The metric M2 set the basic measurement for each type of issues. Therefore, although the scores have been obtained in the survey, the relevance in weight is not applicable for the equation that expresses the final metric calculation.

***: The metric M7 is an additional consideration that may increase the measurement (the weight is captured by the table-5.5).

For the new role of the metric M7 in the new equation-5.3, it is necessary to adjust the table-4.15, which is shown in the table-5.5:

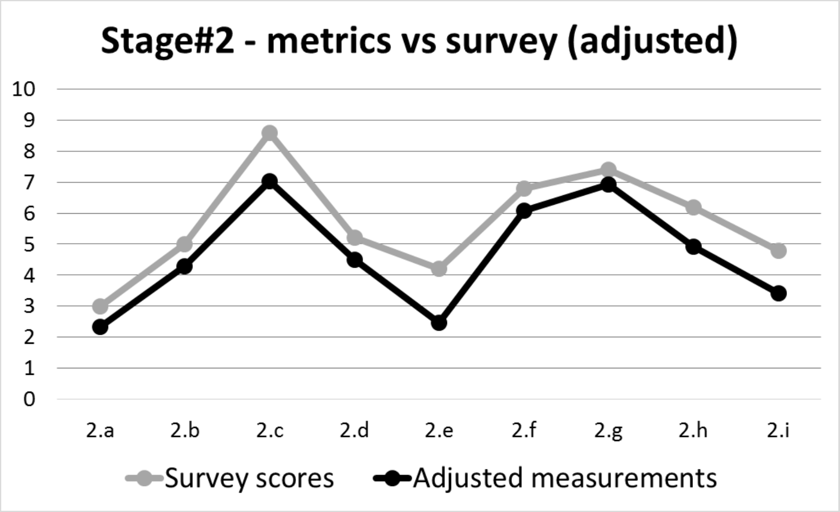
Table-5.5: Adjusted values for metric M7 "*adequacy of issue regarding to audit stage*"

case	Measure	Description	Adequacy
1	1.00	Issue identified in adequate audit Stage	0
2	1.06	Stage#1 scope issue identified in Stage#2 Stage#2 scope issue identified in Stage#3 Stage#3 scope issue identified in Stage#4	1
3	1.13	Stage#1 scope issue identified in Stage#3 Stage#2 scope issue identified in Stage#4	2
4	1.20	Stage#1 scope issue identified in Stage#4	3

The new measurements are used as multiplicative terms to increase the severity of the issue, depending on the adequacy provided by the metric M7. For the best adequacy (i.e., adequacy equal zero), the metric M7 does not change the calculated severity; therefore, the M7 measurement is equal 1. For the worst adequacy (i.e., adequacy = 3), the measurement increases the severity by 20%, which is in line with the M7 relevance obtained from the survey (see table-5.4). Measurements in between were calculated linearly.

After applying the adjusted metrics, the figure-5.12 shows the new measurements in comparison with the survey scores for the Stage#2 issues:

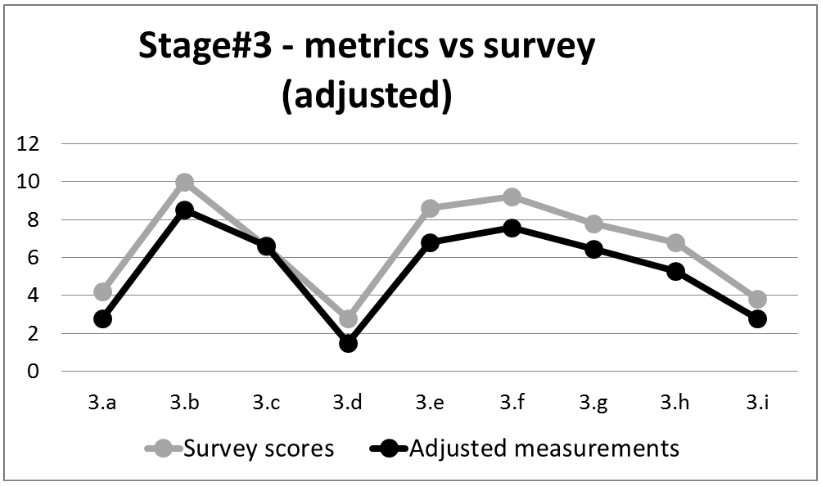
Figure-5.12: New measurements and survey scores for Stage#2 issues



The new measurements still follow the tendency of the survey scores and with all values below them. However, comparing to the results prior to the metrics adjustment, some improvement is perceived because the average difference decreased from 1.83 to 1.52.

The figure-5.13 shows the new measurements and survey scores for the Stage#3:

Figure-5.13: New measurements and survey scores for Stage#3 issues



Similar to Stage#2, the new measurements still follow the tendency of the survey scores. Almost all values are below the survey scores (except the case 3c), but the average difference has decreased from 2.13 to 1.28 after the metric adjustment.

The adjusted metrics generated measurements that are closer to the survey scores, if compared to the original ones. However, measurements are still below the survey scores and the representativeness could be questionable. By analyzing the Stage#2 and Stage#3 issues from the survey, it can be noticed that the scores assigned by the participants tend to represent the severity of the issue in the context of the list provided by the survey. Differently, the adjusted metrics propose to generate representative measurements from the universe of all possible issues of an audit Stage#2 or Stage#3. That can be an acceptable explanation for the differences detected between the measurements from the adjusted metrics and the survey scores. It is important to mention that the measurements follow the same tendency as the survey scores, which is an indication that they are representative of the audit issues severity.

5.5 - Apply the metrics to the results of ANAC audits

The generated metrics were applied to results of ANAC selected audits, as part of the evaluation process. The audits cover representative cases of average performance (normal case), audits not passed (hard case), or passed with merit (merit case). For each software selected, the Stage#1, Stage#2 and Stage#3 (or combinations) were submitted to the metrics and a final measurement was obtained. For examples of applying the metrics, refer to chapter-7, section 7.4. The table-5.6 summarizes the result:

Table-5.6: Summary of audit result of software selected for metrics evaluation in aviation

Case ▼	First audit result				Second audit result				Third audit result			
	F	A	O	Measure	F	A	O	Measure	F	A	O	Measure
Normal	0	33	4	133.57	4	15	3	51.53	0	15	4	40.49
Merit	2	46	4	121.11	1	7	3	22.85	0	4	3	13.12
Hard	0	22	0	122.95	24	2	0	114.18	1	7	2	61.31

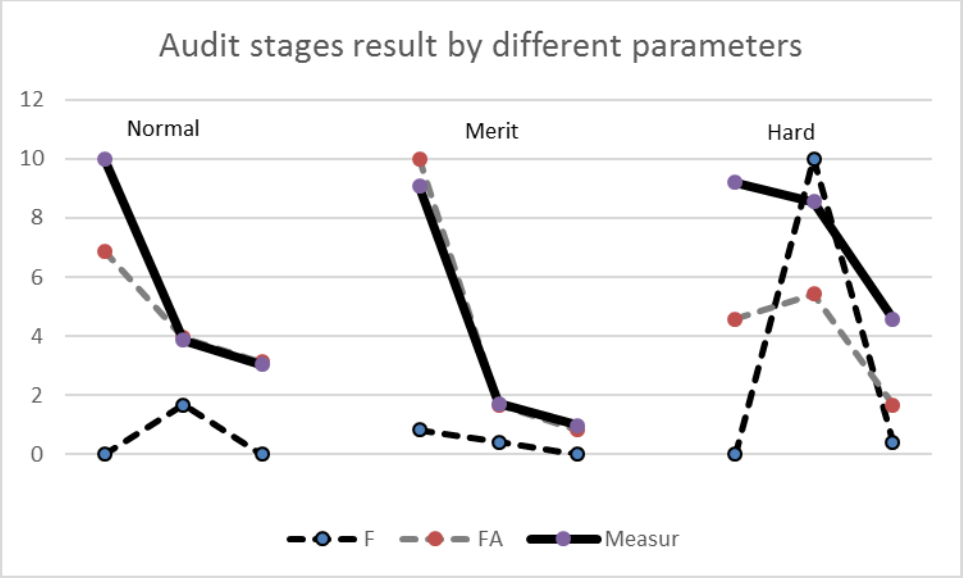
Note: F = number of findings; A = number of actions; O = number of observations
Measure = measurement obtained by applying the Aerospace Metrics

Usually three parameters are used to informally judge the audit result:

1. Number of Findings (F);
2. Number of relevant issues, i.e., Findings plus Actions (FA);
3. Total number of issues, i.e., Findings plus Actions plus Observations (FAO).

The third parameter is the less considered, because the audited company is not expected to address the Observations. In fact, there is a tendency to no longer record the Observations in the audits. Therefore, only the first two parameters were used in the evaluation. The figure-5.14 shows the results expressed by the first two parameters, and by the measurements obtained from the Aerospace Metrics.

Figure-5.14: Audit results expressed by different parameters



The three groups of segments represent the audit stages for normal, merit, and hard cases. Each group has three values in the abscissa representing the three stages, and three segments representing the first two parameters above described plus the measurement obtained by applying the Aerospace Metrics. For facilitating the analysis, the parameters were normalized between 0 and 10. Comparing the measurements among the three cases and for every stage, the analysis is as follows:

- *Stage#1*: The measurements of the three cases were high and very similar among them. That happens very often because companies have difficulties in documenting their processes, regardless of the quality of the processes implemented.

- *Stage#2*: there are differences in the measurements because the Stage#2 assesses the quality of the actual implemented process in the companies' facilities, and cases of normal, merit and hard audits appear. In the normal and merit cases, the abrupt drop in the curve indicates that the problem was more of documentation rather than processes. Differently, the slight drop in the hard case curve indicates low quality processes.

- *Stage#3*: The normal and merit cases curves fall slightly because as the development approaches the final product, companies increase the rigor of process execution due to clearer perception of the impact in the executable code. For the hard case, the curve drop was accentuated, indicating a significant improvement, though not reaching a low value. This is explained by the evaluation against the certification history in the next section.

Comparing the three parameter types, the measurement behaves quite differently from the number of Findings for all three cases. Concerning the number of relevant items (i.e., FA), it behaves similar to the measurement for the normal case, and almost identical for the merit case. However, it differs for the hard case. In Stage#2, the FA of the hard case is much smaller than the other cases, but the measurement is similar. In Stage#3, the FA for hard case is smaller than for the normal case and slightly above the merit case, but the measurement is much larger than the other two cases.

5.6 - Record and analyze the measurements against the software certification history

For every software selected, the final measurements for each audit stage obtained in section 5.5 were compared against the related certification history, which included among others: delays in schedule, re-run of stages, decision for skipping or merging stages due to good performance of the previous one, and difficulties in final compliance. The table-5.7 provides the measurement and related certification history for every selected software. The Stage#4, though not formally executed, is represented in the table for recording the status at the time of the certification issuance (e.g., post-certification pendency, lowering the level of involvement for the next certification).

The measurements analysis of the three cases against the certification history for each stage is as follows:

- *Stage#1*: The measurements of the three cases are similar, and the related certification histories are also equivalents and coherent with the measurements. For the hard case, the measurement reflects the result of the Stage#1, thought initially the plan was to perform a combined Stage#1/2. Besides, the certifier decided for direct involvement in the next stage not because of the stage#1 result, but due to the deficient audit follow-up.

Table-5.7: Measurement and certification history for every selected software

Case▼	Stage	Measure	Certification history
Normal	#1	133.57	Stage passed; Concerns due to new development approach and use of variety of tools; Some issues closed in the next stage; Next stage performed after 7 months;
	#2	51.53	Stage passed; Negative impact due to wrong interpretation of results; Issues closed in 9 months; Next stage performed after 12 months;
	#3	40.49	Stage passed; Issues closed 2 months prior to certification.
	#4	N/A	No post-certification pending; Next certification, possibility of decreasing level of involvement.
Merit	#1	121.11	Stage passed; Many issues raised due to variety of tools usage and two different development approaches for each software component; Experienced consultant hired for next stage; Issues closed in 4 months; Next stage performed after 5 months;
	#2	22.85	Stage passed; Smooth follow-up and issues closed in 7 months; No more consultant for next stage; Next stage performed after 12 months;
	#3	13.12	Stage passed; Smooth follow-up and issues closed in 3 months prior to certification.
	#4	N/A	No post-certification pending; Next certification, involvement decreased.
Hard	#1/2	122.95	Stage passed **with restrictions**; Combined Stage#1/2 due to airplane category, but only Stage#1 performed and passed; Some issues closed in next stage; Supplier not addressing timely; Audit performed by the integrator, but certifier decided to involve in next stage; Next stage performed after 17 months;
	#2/3	114.18	Stage **NOT** passed; Combined Stage#2/3 due to airplane category, but Stage#2 failed and Stage#3 not performed; Some issues not closed even in next audit; Supplier hired experienced consultant; Integrator with permanent staff at supplier site; Bi-weekly basis follow-up by certifier; Same stage re-performed after 8 months;
	#2/3	61.31	Stage passed **with restrictions**; Weekly-basis follow-up by certifier; Some issues remained open for final stage; Informal final stage after 2 months, around certification;
	#4	N/A	Some certification pendency; Problems to be solved post-certification but prior to entry-into-service; certifiers decided to involve in first post-certification change.

- *Stage#2*: The certification histories were also coherent with the related measurements for all three cases. The merit case had audit follow-up smoother than the normal case, with faster audit issues closure and positive managerial impact, whereas the normal case had negative managerial impact. The related merit case measurement was little less than half of the normal case. As for the hard case, the measurement was more than double of the normal case, the Stage#2 was not approved, and the negative managerial impact was very significant, including external consulting, direct involvement of the integrator, and periodic supervision of the certifier.

- *Stage#3*: The certification histories were coherent with the related measurements, with one exception, i.e., comparing the three histories, the normal case measurement was expected to be closer to the merit case than to the hard, but it did not happen. A possible explanation is that the hard case had primary and relevant deficiencies that could be detected in few audit issues, and no further assessment was needed. Differently, the normal case had very detailed process, which facilitated the assessment to detect many discrepancies, though not relevant. Such case may demand future adjustments in the metrics for the Stage#3.

- *Stage#4*: The managerial impact of the three cases described in the certification history is coherent with the measurements from the previous stages. The merit case decided for reducing the involvement of the certifier in the next certification. For the normal case, the possibility of reducing should be evaluated at the beginning of the next certification. Finally, for the hard case the certifier decided to closely supervise any post-certification software change.

The final measurements of audit stages can be used as reference for defining the interval of the decision support table (see table-4.17). To obtain more representative values, it would be necessary to do the above evaluation for all audits performed so far, and continuously evaluating the future ones for refinement. For the thesis, it was restricted to the above selected cases to illustrate the process. The certification history is briefly described, though more information is available for evaluation. Details were omitted due to confidentiality policy.

5.7 – Summary of chapter 5

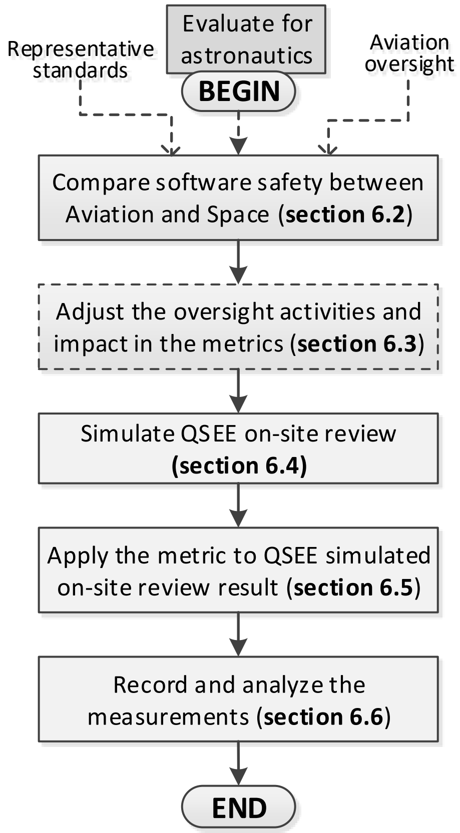
This chapter described the evaluation of the Aerospace Metrics for the aeronautics, more specifically the civil aviation domain. The metrics were applied to a representative set of past aviation software audits and the resultant measurement was evaluated against the software certification history. Surveys and workshops with senior software safety specialists from aviation were also used.

6 – THE METRICS EVALUATION FOR ASTRONAUTICS

6.1 – Overview

This chapter is mainly related to the design cycle of the Design Science. It describes the evaluation of the Aerospace Metrics for the space domain. First, a systematic comparison between aviation and space is performed to identify adjustments in oversight activities and impact in the metrics generated in chapter 4 due to space specific necessities. Then, software audits based on civil aviation are performed in a space project called QSEE (Qualidade do Software Embarcado em Aplicações Espaciais), the results are submitted to the metrics and the resultant measurement is evaluated. The figure-6.1 shows the metrics evaluation process for the astronautics, whose general context is illustrated in the figure-1.1.

Figure-6.1: The metrics evaluation process for astronautics


Evaluate for astronautics
Representative standards
Aviation oversight
BEGIN
Compare software safety between Aviation and Space (section 6.2)
Adjust the oversight activities and impact in the metrics (section 6.3)
Simulate QSEE on-site review (section 6.4)
Apply the metric to QSEE simulated on-site review result (section 6.5)
Record and analyze the measurements (section 6.6)
END

Compare software safety between aviation and space: a systematic comparison is performed between aviation and space domains in the software safety scope focusing on a representative set of standards from both domains. The purpose is to identify reuses of oversight activities and adjustments due to specific necessities of the space oversight, rather than differences and similarities among standards.

Adjust the oversight activities and evaluate the impact in the metrics: the result of the systematic comparison is used as input for identification of adjustments in the oversight activities in order to be applied to space projects. The systematic comparison results and oversight adjustments are both evaluated for impact in the metrics and consequent adjustments.

Simulate QSEE on-site review: The INPE project QSEE is used as case study. As the project has already finished, software audits are simulated by using the civil aviation oversight activities adjusted for space application. The agenda and procedure are adapted to the characteristics and present status of the QSEE project. However, the essence of civil aviation software audits is preserved in terms of allocated time and activities performed, to obtain a representative audit result for evaluation of the metrics.

Apply the metrics to QSEE simulated on-site review result: the simulated audits are divided in 5 stages of development, and for each stage a list of issues identified is produced and submitted for applying the metrics. The produced measurement should reflect the performance of the QSEE software supplier for each audit stage.

Record and analyze the measurements: The measurements obtained from the issues of the simulated audits are evaluated against the performance of the development phase to which the audit stage is related. Information related to the RIDs (record of deviation item) of the joint reviews, as well as comments captured during the debriefing session of the simulated audits are used as source for the measurement evaluation. Moreover, a coverage analysis of the metrics is performed to verify the representativeness of the case study.

6.2 – Systematic software safety comparison between aviation and space

This section presents a systematic comparison between aviation and space domains in the software safety scope focusing on a representative set of standards from both domains.

6.2.1 - Comparison overview

The purpose was to identify reuses of oversight activities from aviation best practices and adjustments due to specific necessities of the space oversight, rather than differences and similarities among standards. To have confidence that the systematic comparison provides a representative result, works on software safety comparison were evaluated (refer to section 2.4.5), where it was possible to identify some assumptions and limitations.

Considering the limitations identified in section 2.4.5, it was specified the *Systematic Comparison Process* that must cover the following four concerns:

Concern-1: Ensure domains' comparison at adequate level, regardless of standards scope;

Concern-2: Clearly identify differences and similarities between both domains that impact the level of reuse of aviation best practices;

Concern-3: Ensure software safety coverage of the chosen scope from both domains;

Concern-4: Facilitate identifying reuses and adjustments from aviation.

For facilitating analysis of reuse and adjustment, the comparison results were classified by taking the *Aviation Oversight* (see section 3.2.6) as reference.

6.2.2 - The Systematic Comparison Process

The *Systematic Comparison Process* comprises five steps as follow:

- **STEP-1:** Identify assumptions and comparison criteria **(**step related to all concerns)
- **STEP-2:** Select domains' items to compare (step related to concern-1)
- **STEP-3:** Perform and record the comparison **(**step related to concern-2)
- **STEP-4:** Perform coverage analysis **(**step related to concern-3)
- **STEP-5:** Classify the systematic comparison results **(**step related to concern-4)

The starting point for constructing the space oversight activities is the *Aviation Oversight*. The *Systematic Comparison Process* provides subsidies for identification of possible reuse of aviation best practices, as well as adjustments due to space oversight necessities. The following classification was adopted for the comparison results:

- **Type-A1, Aviation-only not reusable:** items that, though covered by the *Aviation Oversight*, do not have correspondence in space; for those cases, the aviation best practices are not reusable because are not applicable to the space oversight;
- **Type-A2, Aviation-only outside the *Aviation Oversight***: items that only exist in aviation but are not covered by the *Aviation Oversight*; there are no aviation best practices to consider for reuse;

- **Type-AS1, partially reusable:** items covered by the *Aviation Oversight* but without clear correspondent items in space; they depend on adjustments to allow for reuse of aviation best practices;
- **Type-AS2, fully reusable:** items covered by the *Aviation Oversight,* and with correspondent items in space which should be covered by the space oversight; therefore, can allow for reuse of aviation best practices without adjustments;
- **Type-AS3, similar but outside the *Aviation Oversight***: items that are not covered by the *Aviation Oversight,* though have correspondent items in space; therefore, unlikely to be covered by the space oversight;
- **Type-S1, Space-only but in the intent of the *Aviation Oversight*:** Items that only exist in space, but they should be covered by the space oversight with punctual adjustments, preserving the basic intent of the *Aviation Oversight*.
- **Type-S2, Space-only beyond the intent of the *Aviation Oversight*:** Items that only exist in space, but a cost-benefit analysis should be performed to decide whether to extend the scope of oversight activities to cover them.

Note: A detailed description of the *Systematic Comparison Process* is provided in the appendix-B, and for an illustration of the above classification refer to figure-B.4.

6.2.3 - Summary of the result based on impact in space

The summary of the comparison result focusing on the impact in space is as follows:

- **Type-A1** and **Type-A2:** do not have impact in space;
- **Type-AS1** cases are as follow:
  a. The concept of Low-Level Requirement (LLR), from where the source code is directly produced. For space, the code is produced from the software units which are defined at detailed design phase.
  b. The tests are all based on requirements (i.e., no white box testing). Differently, for space the software units can be tested based on the code structure.
  c. The concept of derived requirements, which are those that are not directly traceable to higher level requirements. Space does not have such concept.

d. The concept of architecture as related to LLR. For the space, the architecture is related to software technical requirements (i.e., similar to HLR), and hierarchically below comes the detailed design (i.e., equivalent to architecture in aviation domain) which is related to the software units.

e. The traceability between HLRs and LLRs, where the architecture should be compatible with (but not traced to) the HLRs. For space, the traceability is between the elements of the architecture (i.e., components) and elements of the detailed design (i.e., units).

f. The criteria for code coverage. For space, it is not required 100% statement coverage for level C, and for some other cases the percentage can be agreed with the customer.

g. The objectives and activities of the certification liaison process. For space, the customer-supplier relationship needs similar activities and can partially reuse from aviation.

h. The planning process, mandatory at the beginning to plan all activities to be performed throughout the development. For space, it is not mandatory to plan all activities at the beginning, but during the development at the suitable time. For example, development plan is required for SRR, but verification plan is required for PDR and maintenance plan for QR.

- **Type-AS2** summary result is as follow:

a. The *Aviation Oversight* activities can be reused by the space oversight to assess through samplings the quality of, and adherence to the process of development and respective verification, covering from the space system requirements allocated to software until the executable code, including the requirement-based testing in the representative environment. The quality, configuration control and traceability of the generated life-cycle data, the nonconformity records and actions for solution, and the quality assurance records, among others, are used as evidences.

- **Type-AS3:** no cases have been found;

- **Type-S1** cases are as follow:

a. Space standards can be tailored based on technical, operational, managerial, conditional requirements, and customer-supplier agreement, which affect the mandatory set of

ECSS-requirements, and should be captured by the space oversight process prior to starting the audit assessment.

b. For space, the customer specifies the requirements baseline and provides them to the supplier. However, ECSS allows the supplier to specify the requirements baseline under support of the customer. Therefore, those activities that are typical of system scope are also addressed in the software scope.

c. Due to the customer-supplier approach, space has the delivery and acceptance process, which delimitates the end border between supplier and customer.

d. Due to some spacecraft operational characteristics, space software requires the possibility of maintenance inflight, high integrity communication with Ground, protection against single-event upset (SEU), and concerns on disposal phase.

e. Space allows the customer to require an independent organization to perform verification and validation.

f. Space provides a separate process for maintenance.

g. Space requires the use of model to provide behavioral view in order to support the verification of requirements, architecture and detailed design.

h. Space requires the use of computational models for the dynamic architecture design.

i. Space requires mission and configuration dependent data to segregate from the software, e.g., a separate database.

j. Space requires the specification of software quality requirements.

- **Type-S2** cases are as follow:

  a. Processes of Procurement and Retirement;
  b. Organization-related guidance including qualification and training program;
  c. Process assessment for capability and maturity level;
  d. Ground software development assurance.
  e. A process for Operation phase prior to launching.

Note: Detailed information on comparison result can be found in the appendix-B.

6.3 – Adjustment of oversight activities and impact in the metrics

6.3.1 – Adjusting the aviation oversight activities for space application

The adjusted activities of the aviation oversight become the embryo for the Space Oversight Framework, whose overview is provided in the appendix-C. Some differences identified by the systematic comparison affect the main structure of the framework while others affect specific procedures, and are described as follow:

a. Tailoring of ECSS standards (Type-S1, case "a"): demands an additional adjustment in the initial risk assessment, to determine the set of applicable ECSS requirements for the specific project that will be under oversight.

b. Different stakeholders (Type-S1, cases "b" and "c"): the first stage (Stage#1) covers activities performed mainly by the customer and, consequently, the Stage#5 also includes customer activities for acceptance;

c. Communication with Ground (Type-S1, case "d"): demands a specific oversight procedure for assessment of the validation due to Ground environment necessities.

d. Independent V&V (Type-S1, case "e"): demands a separate oversight procedure focusing on a specific stakeholder other than customer and supplier, which is responsible for the independent V&V.

e. Different processes (Type-S1, case "f"): it would demand an additional process specific for maintenance, after the acceptance process. However, for this thesis it was decided to keep the maintenance under responsibility of the supplier, in order to maintain similarity with the civil aviation approach.

The other Type-S1 cases as well as the Type-AS1 affect only the spreadsheet used for compliance checking of ECSS-requirements called *Software Compliance Checklist*. The Type-AS2, by definition comprises those requirements whose related oversight activities are fully reusable; therefore, does not demand any adjustment in the aviation oversight activities.

The table-6.1 provides the mapping of the impact of comparison result in the space oversight framework. The Type-S2 demands a cost-benefit analysis, but for this thesis the framework described in appendix-C does not include the activities necessary to assess the Type-S2 ECSS-requirements. Therefore, they are considered out of scope.

Table-6.1: Impact of the comparison result in the space framework

Type ▼	Framework ▶ case ▼	Main Structure	Procedure	Checklist	No impact	Out of Framework scope
AS1	a	-	-	●	-	-
	b	-	-	●	-	-
	c	-	-	●	-	-
	d	-	-	●	-	-
	e	-	-	●	-	-
	f	-	-	●	-	-
	g	-	-	●	-	-
	h			●		
AS2	a	-	-	-	●	-
S1	a	-	●	-	-	-
	b	●	-	-	-	-
	c	●	-	-	-	-
	d	-	●	-	-	-
	e	-	●	-	-	-
	f	●	-	-	-	-
	g	-	-	●	-	-
	h	-	-	●	-	-
	i	-	-	●	-	-
	j	-	-	●	-	-
S2	a	-	-	-	-	●
	b	-	-	-	-	●
	c	-	-	-	-	●
	d	-	-		-	●
	e	-	-	-	-	●

6.3.2 – Evaluating the impact in the metrics

The table-6.2 presents the impact of the comparison result in the metrics. The Type-AS2 comprises, by definition, those requirements whose related oversight activities are fully reusable; therefore, does not demand any adjustment in the aviation oversight activities and consequently does not impact the metrics.

The Type-AS1, case "h", the metric M1 "*document evaluation*" is not changed (Table-4.9 still applicable), but there are changes in scope of the related equation because ECSS does not require an initial process for planning all activities. Hence, the initial planning has a restrict scope and the follow-on processes can include specific planning, i.e., all stages must include the M1 metric for document evaluation, which means the equation-5.1 can be used in all stages, not only for stage#1. Therefore, for space the final measurement of the audit result should consider the result of equation-4.5 and equation-5.1, as follow:

$$mSpaceAudit = DocEvalMeasur + mAudit \tag{6.1}$$

Table-6.2: The impact of the comparison results in the metrics

Type ▼	metric ▶ case ▼	*M1*	*M2*	*M3*	*M4*	*M5*	*M6*	*M7*
AS1	a	-	-	-	-	-	-	-
	b	-	-	-	-	-	-	-
	c	-	-	-	-	-	-	-
	d	-	-	-	-	-	-	-
	e	-	-	-	-	-	-	-
	f	-	-	-	-	-	-	-
	g	-	-	-	-	-	-	-
	h	**YES**						
AS2	a	-	-	-	-	-	-	-
S1	a	-	-	-	-	-	-	-
	b	-	-	-	-	**YES**	-	**YES**
	c	-	-	-	-	-	-	-
	d	-	-	-	-	**YES**	-	**YES**
	e	-	-	-	-	-	-	-
	f	-	-	-	-	-	-	**YES**
	g	-	-	-	-	-	-	-
	h	-	-	-	-	-	-	-
	i	-	-	-	-	-	-	-
	j	-	-	-	-	-	-	-
S2	a	TBE	TBE	TBE	TBE	TBE	TBE	TBE
	b	TBE	TBE	TBE	TBE	TBE	TBE	TBE
	c	TBE	TBE	TBE	TBE	TBE	TBE	TBE
	d	TBE	TBE	TBE	TBE	TBE	TBE	TBE
	e	TBE	TBE	TBE	TBE	TBE	TBE	TBE

Note: TBE = to be evaluated

The Type-S1, cases “b” and “d”, impacts both the metrics M5 and M7. For M5 those differences include requirements baseline at the beginning (replacing system level phases) with related validation at the end, plus delivery and acceptance test, as illustrated in the table-6.3 for the revised metric MS5. The impact in the metric M7 is because the adopted space oversight framework has 5 audit stages, i.e., one extra stage if comparing with the aviation oversight.

Table-6.3: Metric MS5 "*distance to the final product*" adjusted for space domain

Case	measure	Description	Distance
1	1.0	Issue related to requirements baseline phase	11
2	1.9	Issue related to initial supplier planning phase	10
3	2.8	Issue related to requirements and architecture phase	9
4	3.7	Issue related to detailed design phase	8
5	4.6	Issue related to coding phase	7
6	5.5	Issue related to integration phase	6
7	6.4	Issue related to unit and integration testing	5
8	7.3	Issue related to validation of the technical specification	4
9	8.2	Issue related to final analyzes (e.g., coverage analysis)	3
10	9.1	Issue related to validation of the requirements baseline	2
11	10.0	Issue related to delivery and acceptance phase	1
12	5.5	Issue related to most of or all phases	N/A

The table-6.4 illustrates the revised metric MS7:

Table-6.4: Metric MS7, "adequacy of the issue regarding to audit stage" adjusted for space

case	Measure	Description	Adequacy
1	1.00	Issue identified in adequate audit Stage	0
2	1.05	Stage#1 scope issue identified in Stage#2 Stage#2 scope issue identified in Stage#3 Stage#3 scope issue identified in Stage#4 Stage#4 scope issue identified in Stage#5	1
3	1.10	Stage#1 scope issue identified in Stage#3 Stage#2 scope issue identified in Stage#4 Stage#3 scope issue identified in Stage#5	2
4	1.15	Stage#1 scope issue identified in Stage#4 Stage#2 scope issue identified in Stage#5	3
5	1.20	Stage#1 scope issue identified in Stage#5	4

The type-S1, case "f", would demand an additional process specific for maintenance, after the acceptance process, which would impact the metric M7. However, for this thesis it was decided to keep the maintenance under responsibility of the supplier, in order to maintain similarity with the civil aviation approach.

The Type-S2 demands a cost-benefit analysis for possible scope extension of the space oversight framework. Hence, the table-6.2 indicates TBE (to be evaluated). As an example, if after the cost-benefit analysis it is decided to include the operation process (Type-S2, case “e”), then the number of stages would increase to 6, which would impact the metric M7, and the distance to the final product also would change, impacting the metric M5.

The following metrics have not been impacted by either the comparison result and or related oversight adjustments:

- Metric M2, “purpose of the issue” not changed (Table-4.10 still applicable)
- Metric M3, “*type of artifact impacted*” not changed (Table-4.11 still applicable)
- Metric M4, “*root cause*” not changed (Table-4.12 still applicable)
- Metric M6, “*amount of artifacts impacted by the issue*” not changed (Table-4.14 still applicable)

6.4 – Case study - QSEE project

The QSEE project was used as a case study with the purpose of exercising the metrics, and also to identify gaps in the metrics coverage that demand additional activities, such as additional case studies, surveys and analyzes.

6.4.1 – The QSEE project – Quality of Space Application Embedded Software

According to Santiago et al. (2007), the QSEE project was conceived to achieve three objectives:

a. Transfer to Brazilian software industry INPE’s knowledge in software for space application, particularly V&V tools, methods and techniques used for payload embedded software on-board of scientific satellites and balloon applications;

b. Update the software development methodology for scientific satellites and balloon payloads;

c. Create a methodology so that INPE can accept software developed by private companies.

Software for payload data-handling computer (SWPDC) was specified by INPE as a pilot project, using the X-ray Monitor and Imager (MIRAX) satellite as case study. MIRAX is a small X-ray astronomy satellite mission designed to monitor a large region around the central galactic plane for transient phenomena. Two versions of the software were developed by different suppliers, but using the same set of requirements as input. The QSEE had the following main stakeholders:

a. An INPE team as the customer, responsible for the specification of the pilot project which includes the Requirements Baseline (RB);

b. Another INPE team as in-house software supplier, responsible for the development of one software version of the pilot project;

c. DBA Engenharia de Sistemas LTDA as an outsourced software supplier, also responsible for the development of one software version of the pilot project;

d. An IVV group comprising specialists from INPE and UNICAMP, focusing on the acceptance tests of both software versions of the pilot project.

The QSEE project execution comprised three phases:

- Phase 1: teams' constitution and training; study and tailoring of ECSS standards;
- Phase 2: pilot project specification; construction of software acceptance process by INPE applying IVV approach;
- Phase 3: development of the SWPDC software, and validation of the two versions of the pilot project using the acceptance methodology.

The following documents were produced by the stakeholders and reviewed by the planned joint reviews (i.e., SRR, PDR, DDR, CDR):

a. By INPE, as the customer: RB document and communication protocol specification document;

b. By INPE, as a supplier: development plan, Technical Specification (TS) document, design document, test plan, test report document, and user manual;

c. By the DBA: same set of documents as INPE-supplier;

d. By the IVV group: IVV plan, subsystem IVV plan, subsystem TS document.

Remark: a joint group produced report documents of SRR, PDR, DDR and CDR joint reviews.

For details on the architecture adopted by QSEE refer to Santiago et al. (2007), for the IVV refer to Ambrosio et al. (2008), and for the methodology used for validation tests refer to Pontes et al. (2014).

6.4.2 – The QSEE project adapted for case study

Even considering that the QSEE was a pilot project with adjustments if compared to a typical INPE project, the evaluation during the audit considered aspects of a typical project. Moreover, although safety concerns of the QSEE pilot project were considered irrelevant with negligible adverse impact, the case study applied the highest assurance rigor. For convenience, the scope of the simulated audits was limited to INPE as a customer, INPE as a supplier, and the IVV group. The external supplier DBA was excluded due to presumed difficulties in obtaining further information, if deemed necessary.

Mattiello et al. (2007) describes an analysis of the planned activities of the QSEE stakeholders regarding to the ECSS applicable requirements, aiming at supporting the stakeholders to have processes capable of complying with the ECSS. Despite that, for this case study the simulated audits have verified compliance to the ECSS standards covered by the Space Framework.

For obtaining representative results to exercising the metrics, the agenda of the simulated audit was based on the aviation, but with the following adaptation:

a. Day 1 was a meeting with QSEE team simulating the first day of a typical aviation on-site review;

b. Day 2, 3 and 4 were a desktop evaluation of QSEE lifecycle data, simulating the on-site assessment of company’s artifacts;

c. Day 5 was an on-site evaluation of the development environment;

d. Day 6 was a meeting with QSEE team simulating the closing day of a typical on-site review (i.e., debriefing).

Remark: A typical aviation on-site review stage usually takes 3 to 4 days. This simulated on-site review took longer as it covered the whole development, comprising 5 audit stages. Moreover, unlike the aviation where the company´s artifacts are accessible only on-site, the

QSEE artifacts were fully accessible, allowing for detailed assessment in a desktop basis, with the QSEE developers available through e-mails or telephone for any questions. Such facility made it possible to simulate the on-site assessment of all stages in those 3 days of desktop evaluation.

6.4.3 – Summary results of the simulated audit performed in the QSEE project

The purpose of the simulated audit was to assess the planning and implementation of the development process through examination of the software life cycle data of QSEE regarding to the compliance with ECSS applicable standards. The audits were performed by the author of the thesis, and the auditees were two members of the QSEE pilot project: the project manager and the IVV responsible. The summary results are presented in the table-6.5 below.

Table-6.5: Summary of simulated audit issues per stages

Stage▶	#1	#2	#3	#4	#5
Number of issues▶	20	15	3	9	4

The following strengths were identified:

a. Quality of documentation, more specifically the TS document, software design document, and subsystem TS document;

b. Quality of TS activities, i.e., the generation of software requirements, architecture, and detailed design;

c. Quality of acceptance tests based on models (i.e., IVV group) and the relevance of the obtained results.

Some points of concern were also identified:

a. Missed tailoring of ECSS standards in terms of applicable requirements;
b. RB without enough information;
c. Absence of software safety and dependability analysis;
d. Absence of validation tests against RB;
e. Low coverage of TS validation tests;
f. Supplier verification activities are unclear;
g. Stakeholders' roles and responsibilities are unclear;

h. Some deficiencies in QA activities.

The majority points of concerns identified is consequence of the QSEE project characteristics adapted for case study, as described in 6.4.1 and 6.4.2 (e.g., concerns “a”, “c”, “d”, “f”, “g”). The concern “b” was also identified by the QSEE project joint reviews, but the concerns “e” and “h” were identified only by the simulated audits and agreed by the QSEE members during the debriefing.

6.5 – Applying the metrics to the issues raised in simulated audits

The resultant measurement for all five stages of the simulated QSEE audit is presented in the table-6.6:

Table-6.6: The measurements of the simulated QSEE audit

Stage▶	#1	#2	#3	#4	#5	Total
Number of issues▶	20	15	3	9	4	51
Measurements ▶	104.7	62.0	21,6	49.8	26.7	264.8
Issues percentage▶	39.2	29.4	5.9	17.6	7.8	100
Measurements percentage▶	39.5	23.4	8.2	18.8	10.1	100
Measurements per issue▶	5.24	4.13	7.2	5.53	6.68	5.19

The measurements, as well as the number of issues, tend to decrease throughout the software development, which is also a tendency in aviation. As explained in chapter-3, the number of issues does not necessarily reflect the result of the audit stage. For example, the number of issues raised in Stage#2 accounts for 29.4% of the total issues, but reflects for 23.4% of the total measurements, which means the issues severity is lower than the average. Stage#3 and Stage#5 are on the other side, while Stage#1 and Stage#4 show a balance between the number of issues and measurements. For an example of metrics applied to space audit issue to obtaining the related measurement, please refer to chapter 7, section 7.5.

Concerning the representativeness of the QSEE project as case study for the metrics evaluation, the table-6.7 shows a mapping of the issues raised during the simulated audits against the metrics, for having an idea of the metrics coverage:

Table-6.7: The coverage of the metrics by the issues identified in the simulated audit stages

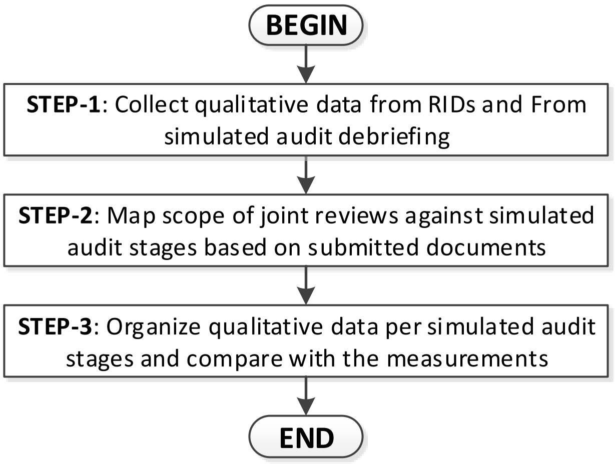
Stage ▼	metric ▶ #issues ▼	*M1*	*M2*	*M3*	*M4*	*M5*	*M6*	*M7*
#1	20	7	13	13	13	12	13	13
#2	15	4	11	11	8	11	11	11
#3	3	2	1	1	1	1	1	1
#4	9	1	8	8	7	8	8	8
#5	5	2	3	3	2	3	3	3

Unlike aviation, the M1 metric is applicable to all stages, though Stage#1 exercises it the most. The other metrics, which are applicable to process evaluation and process adherence assessment, were also exercised by all stages. Some issues did not exercise the M4 metric (root cause) because those issues were not related to non-compliances. It is important to mention that an issue is either a document or process related; therefore, the sum “M1+M2” is always equal the total issues for every stage.

6.6 –The measurements analysis

This section presents the analysis of the measurements obtained from applying the metrics to the issues raised during the QSEE simulated audit. Figure-6.2 illustrates the process used:

Figure-6.2: The process for QSEE measurements analysis


BEGIN
STEP-1: Collect qualitative data from RIDs and From simulated audit debriefing
STEP-2: Map scope of joint reviews against simulated audit stages based on submitted documents
STEP-3: Organize qualitative data per simulated audit stages and compare with the measurements
END

The three steps process is described below:

STEP-1: Collect qualitative data from RIDs and from simulated audit debriefing

The initial intention was to apply the metrics to the RIDs, but after further investigation, the strategy was changed due to the following reasons:

a. There is a strong relation between the issues evaluated by the metrics and the oversight activities performed to identify those issues;

b. The activities performed by the joint reviews focus on documents evaluation, whereas the oversight activities focus on process evaluation and process adherence assessment;

c. The format of the RIDS is not in line with the space oversight framework. For example, the RIDs provide for a solution proposal by the reviewer, whereas the oversight philosophy does not recommend the reviewer to propose any solution, i.e., the solution must be under the developer responsibility, and the reviewer role is to identify issues and obtain the developer understanding and agreement on them.

Consequently, it was concluded that applying the metrics to the RIDs would obtain measurements that are not representative for the metrics evaluation. Nevertheless, considering that the joint reviews and space oversight activities are just different approaches to assess the same project, both results can be comparable if the format of the records is disregarded and, hence, the comparison can add value to the metrics evaluation. The table-6.8 provides a summary of the RIDs recorded during the QSEE joint reviews:

Table-6.8: Number of RIDs produced during the QSEE joint reviews

review▶ stakeholder ▼	SRR	PDR	DDR	CDR	Total
Customer	6	16	-	-	22
Supplier	19	15	16	17	67
IVV group	4	-	-	21	25
Total	29	31	16	38	114

Source: adapted from Ambrosio et al. (2008)

The table accounts only the INPE as supplier, i.e., RIDs raised against DBA artifacts were not considered. Ambrosio et al. (2008) states that "*although many of the RIDs indicated minor problems, some critical problems were pointed out in the different reviews*", and provides a summary of those critical problems as follow:

- SRR: protocols specification and a description of the operation modes were not provided by the customer;
- PDR: Deliverables and deadlines were not specified by the supplier. TS did not include interruptions for data acquisition and faults treatment.

- DDR: Software design did not include behavior of how to deal with commands sequencing. Message sequencing was misunderstood.
- CDR: TS did not include performance testing and test coverage analysis.

The minor problems are related to quality of the information, i.e., information not clear, incomplete, conflicting to each other, with editorial errors, which is typical of reviews focused on documentation.

Audit debriefing: During the 6th day of the simulated audit a debriefing with the QSEE members took place, where the results were presented and discussed. A summary of the main comments of the QSSE members are provided below:

a. Due to the QSEE been a pilot project focusing on software scope, system level activities were almost absent (e.g., safety and dependability analysis were not performed), which may have contributed for deficient RB specification;

b. Due to human resources constraints, some independence was not followed as expected. For example, a QSEE member played the role of both customer and supplier, which may have contributed for TS requirements with specification belonging to system level scope;

c. Due to human resources constraints, the INPE as supplier focused more on development and testing activities, whereas less emphasis was put on other verification activities (e.g., reviews, inspections, analysis) as well as configuration management and quality assurance;

d. The supplier DBA performed the complete verification because the company as CMMI-3 provides for that process. However, the INPE as customer did not perform any formal oversight on the suppliers because that activity was not the focus of the QSEE project;

e. INPE as customer did not emphasize SQA role because that was not the focus of the QSEE project. Consequently, SQA activities related to RB specification as well as supervision of suppliers and IVV group were deficient;

f. Due to the lack of detailed specification at system level (e.g., superficial RB requirements), the IVV group had to use design information of both suppliers, resulting in two different models for supporting the automatic test cases generation as part of the acceptance process.

STEP-2: Map scope of joint reviews against simulated audit stages based on submitted documents

Considering the documents that were submitted for each joint review, the table-6.9 provides a mapping of the simulated audit stages against the joint reviews in terms of scope, for the case of QSEE project:

Table-6.9: The mapping of the simulated audit stages against the joint reviews for the QSEE

review ▶ Stage ▼	SRR	PDR	DDR	CDR
#1	●●	●	-	-
#2	-	●●	●	-
#3	-	-	●●	●
#4	-	-	●	●●
#5	-	-	-	●●●

The Stage#1 maps to SRR but also to PDR, because some SRR input documents (i.e., RB document and Software Development Plan) are again inputs for the PDR with revised versions contemplating the SRR results. Similar situation happens to Stage#2 and Stage#3. Regarding the Stage#4, the supplier test plan is input for the DDR, but the TS validation is reviewed during CDR. The mapping shown is slightly different if compared with the Space Oversight Framework presented in appendix-C. According to Ambrosio et al. (2008), *”the two last reviews QR and AR are performed after the integration of the equipment embedding the software”*, but as a pilot project with scope restricted to the software, the QSEE did not go through those phases and the QR and AR were not performed. Consequently, validation against the RB and acceptance tests fell under the scope of CDR.

STEP-3: Organize qualitative data per simulated audit stages and compare with the measurements

The table-6.10 provides the measurement calculated for each simulated audit stage, and related qualitative evaluation summarized from data obtained from RIDs and audit debriefing:

Table-6.10: The measurement for each stage and related qualitative evaluation

Stage	Measure	Qualitative evaluating data from RIDs and audit debriefing
#1	104.7	System level activities almost absent; Superficial RB specification; Protocol specification and description of operation modes not provided by customer; Customer SQA was deficient; Deliverables and deadlines not specified by supplier; RB requirements not verified by review; Requirements standard non-existent; Roles among stakeholders unclear and may have overlaps; Plans not reviewed against contract requirements;
#2	62.0	TS did not include interruptions for data acquisition and faults treatment; TS requirements contain information that should have been specified at system level; TS requirements and architecture not verified by review; Architecture standard non-existent; Traceability not enough to support verification of TS against RB;
#3	21,6	Software design did not include behavior of how to deal with commands sequencing; Detailed design not verified by review; Design standard non-existent; Traceability from design elements to code not clear;
#4	49.8	Test specification did not include performance testing and test coverage analysis; Test cases and procedures were not reviewed as part of supplier verification process; Some TS requirements have inappropriate verification method; Test cases do not fully cover the requirements for normal cases, and robustness cases are non-existent; traceability between test cases and TS requirements is not enough to support ensuring verification coverage; RB were not testing-validated by supplier;
#5	26.7	Acceptance tests were specified by IVV group using methodology that generates test cases from models that contains supplier design information; Delivery and acceptance by supplier not clear; Acceptance test cases not reviewed against RB and traceability not clear;

An overall analysis of the measurements against the respective qualitative evaluation for each stage has not shown any unacceptable discrepancy. By comparing the description of the qualitative evaluation among the stages, one can conclude that the Stage#1 is the most deficient, i.e., the scope of system level activities including RB specification and overall planning is deficient. The related measurement is by far the highest (i.e., 104.7), confirming the qualitative evaluation. On the other side, the qualitative evaluation shows the Stage#3 as the least deficient, which is also confirmed by the lowest measurement (i.e., 21.6), and followed by the Stage#5 (i.e., 26.7). In between the extreme cases, there are Stage#2 and Stage#4. The qualitative evaluation for the Stage#4 seems to describe a worse scenario than the Stage#2. However, the respective measurements do not confirm it, i.e., measurement for Stage#4 is lower than Stage#2 (i.e., 49.8 and 62.0). An explanation could be because the Stage#2 is directly impacted by the input from the Stage#1, which is the most deficient. Hence, the number of Stage#2 issues tends to be high, which is confirmed by the table-6.7, affecting the final measurement.

It is important to mention that the deficiencies recorded in table-6.10 are not from the QSEE as project, but just consequence of adapting the QSEE to be used as case study. The QSEE project has successfully reached its purpose, and several relevant papers have been published.

For the use of QSEE as case study, this thesis has adapted the original QSEE project (see section 6.4.2) and has taken some assumptions in order to obtain acceptable representativeness in the case study.

6.7 – Summary of chapter 6

This chapter described the evaluation of the metrics for the astronautics, more specifically the space domain. The evaluation started with a systematic comparison between aviation and space to identifying adjustments in the oversight activities and impact in the generated metrics due to space necessities. Then, software audits were performed in the QSEE space project by applying the adjusted oversight activities. The audits results were submitted to the adjusted metrics and the resultant measurement was evaluated.

7 – METRICS FOR OVERSIGHT OF SOFTWARE SUPPLIER OF SAFETY-CRITICAL AEROSPACE SYSTEMS – THE RESULTS

7.1 - Overview

This chapter is related to the design cycle of the Design Science. It provides a summary of the Aerospace Metrics, whose concept, generation and evaluation were presented in the previous chapters of this thesis. The metrics are summarized in terms of equations and related tables that capture quantitative values. Equations and tables are applicable to both aeronautics (i.e., aviation) and astronautics (i.e., space), unless otherwise specified. Examples are provided for metrics applied to aviation and space audit issues, and use for management decision support.

7.2 - Metrics related to documents evaluation

The equation-7.1 relates to metrics of documents evaluation:

$$DocEvalMeasur = \sum_{i=1}^{n} \mathrm{m(i)} * \mathrm{R(i)} \qquad (7.1)$$

Where:

DocEvalMeasur: the final measurement for audits focusing on documents evaluation;

m(i): the measurement for each issue “i”, by applying the table-7.1;

R(i): the technical relevance of the information related to each issue “i”, which is based on qualitative judgment of the auditor (see table-7.2 for possible values);

i=1…n, n: total amount of audit issues

For aviation, the above equation is applicable only during the Stage#1, because the planning of all phases is concentrated at beginning and comprises the production of a set of planning documents (scope of Stage#1). Other Stages are concentrated in process evaluation and process adherence assessment, which uses the equations presented in section 7.3.

For space domain, ECSS does not require an initial process for planning all activities. Hence, the initial planning does not cover all phases and the follow-on processes can include specific planning, i.e., all stages must include the M1 metric for document evaluation, which means the equation-7.1 can be used in all stages, not only for Stage#1.

The table-7.1 provides description of possible cases that can be identified during a document evaluation, and related measurements.

Table-7.1: Quantitative values for metric M1 *"document evaluation"*

case	Description	measure
1	The information contains editorial errors (typos)	**0.2**
2	The information is out of context, i.e., recorded in an inappropriate section or document	**1.7**
3	The information is inconsistent between sections or documents	**3.8**
4	The information is superficial or incomplete	**4.0**
5	The information is confused, ambiguous	**4.2**
6	Could not find in the provided documents the required information for compliance	**5.2**
7	The information is clear and complete, but is considered unacceptable	**6.1**

The table-7.2 provides the technical relevance of the information related to each issue raised during a document evaluation, which is based on qualitative judgment of the auditor, and related measurements.

Table-7.2: The technical relevance (R) of the information related to document evaluation

Relevance	Qualitative technical relevance	Measure
0	Very low or negligible	**0.36**
1		**0.48**
2	Low	**0.61**
3		**0.74**
4	Medium	**0.87**
5		**1.00**
6		**1.13**
7	High	**1.26**
8		**1,39**
9	Very high	**1.52**
10		**1.64**

7.3 - Metrics related to process evaluation and process adherence assessment

The final measurement calculation of the audit result, in the scope of process evaluation and process adherence assessment, is divided in two equations. The first equation (Equation-7.2) shows the expression that calculates the measurement for each issue:

$$m = \left(M(2) * \sum_{j=3}^{6} M(j) * W(j) \right) * M(7) \qquad (7.2)$$

Where:

m: the measure of the issue severity

M(2): the percentage related to the purpose of the issue (refer to table-7.4)

M(j) the measurement for the metric Mj (refer to table-7.5 until table-7.9);

W(j): the percent relevance of each metric (refer to table-7.3);

M(7): the measurement for the metric M7 (refer to table-7.10 and table-7.11).

The second equation (Equation-7.3) calculates the final measurement of the audit result, as follow:

$$mAudit = \sum_{i=1}^{n} \mathrm{m(i)} \qquad (7.3)$$

Where:

mAudit: the final measurement of the audit result

m(i): the measurement of the issue i, by applying the equation-7.2 for each issue

n: total amount of issues recorded in the audit

For aviation, the above equations are not applicable in the Stage#1, where a set of planning documents is produced and demands only the equation-7.1. For space the equations are applicable to all stages, and the final measurement of the audit result should consider all three equations as follow:

$$mSpaceAudit = DocEvalMeasur + \ mAudit \qquad (7.4)$$

The table-7.3 presents the relevance of each metric in percentage (M3, M4, M5 and M6), which are used in the equation-7.2.

Table-7.3: The relevance of each metric in percentage

Metric	Description	Relevance W
M3	Type of artifact impacted by the issue	**23**
M4	Root cause of the issue	**28**
M5	Distance from the issue to the final product	**20**
M6	Amount of artifacts impacted by the issue	**29**

The following tables present cases description and related measurements for the metrics M2 "purpose of the issue", M3 "*type of artifact impacted*", M4 "*root cause*", M5 "*distance to the final product*", M6 "*amount of artifacts impacted by the issue*" and M7 "*adequacy of the issue regarding to the audit stage*". The metrics M5 and M7 have separate tables for aviation and space, while all others are equally applicable to both domains.

Table-7.4: Percentage values for metric M2 "purpose of the issue"

Case	Description	Percentage
1	A suggestion for process improvement detected during the audit. However, the process is considered sufficient for compliance	**7**
2	An issue to request additional information, but a priori without any impact in concerns discussed during the audit	**25**
3	An issue to correct a punctual process deficiency (or adherence to the process) detected during the audit	**47**
4	An issue to request additional information, which may drive to a non-compliance that was not conclusive during the audit	**75**
5	An issue to record a non-compliance and request a closure approach	**100**

Table-7.5: Quantitative values for metric M3 "*type of artifact impacted*"

Case	Description	measure
1	Issue opened against informal data (e.g., an SQA spreadsheet for informal control not planned for use by the process)	**1.2**
2	Issue opened against Software Quality Assurance (SQA) Records	**6.2**
3	Issue opened against Software Configuration Management Records	**6.5**
4	Issue opened against plans and standards	**6.9**
5	Issue opened against Problem Reports (PR)	**6.9**
6	Issue opened against tools (e.g., poor qualification report, justification for non-qualification is unacceptable)	**7.3**
7	Issue opened against traceability (e.g., requirement points to wrong parent requirement, insufficient granularity)	**7.7**
8	Issue opened against verification data, including reviews, inspections, verification cases and procedures, verification results and related artifacts	**9.2**
9	Issue opened against requirement, design, code or configuration data (e.g., ambiguous requirement, architecture incompatible with requirements, code does not fully implement the requirement	**10.0**

Table-7.6: Quantitative values for metric M4 "*root cause*"

Case	Description	measure
1	It was only a slip, an isolated case	**2.0**
2	Similar cases have been found involving the same person, raising suspicion of insufficient training	**5.9**
3	The training material was deficient, raising suspicion that the person did not understand enough the activity to perform	**6.2**
4	The amount and complexity of the information needed for the activity may have contributed to the mistake	**6.6**
5	The process followed was not clear, which may have contributed to the mistake	**6.9**
6	The process followed was clear but incorrect, leading the person to the mistake	**10.0**
7	Default value for the case where the root cause cannot be determined at the time the issue is raised (measurement = average of all cases)	**6.2**
8	The issue is not related to (potential) non-compliance regarding process adherence. Therefore, the root cause is not applicable.	**0.0**

Note: For the case 7, it is assumed the average value as default, which may change during the issue follow-up, once the root cause is identified after further investigation.

Table-7.7: Quantitative values for metric MA5 "*distance to the final product*"

Case	Description	measure
1	Issue related to system level phases	**1.1**
2	Issue related to planning phase	**2.2**
3	Issue related to requirements phase	**3.3**
4	Issue related to design phase	**4.4**
5	Issue related to coding phase	**5.5**
6	Issue related to integration phase	**6.6**
7	Issue related to unit testing	**7.7**
8	Issue related to integrated testing	**8.8**
9	Issue related to final analyzes (e.g., structural coverage analysis)	**10.0**
10	Issue related to most of or all phases	**5.5**

Note1: applicable only for Aviation domain;

Note2: The case 10 captures those issues that impact or are applicable to various phases, for example, some deficiencies in SQA process or configuration control.

Table-7.8: Metric MS5 "*distance to the final product*" adjusted for space domain

Case	Description	measure
1	Issue related to requirements baseline phase	**1.0**
2	Issue related to initial supplier planning phase	**1.9**
3	Issue related to requirements and architecture phase	**2.8**
4	Issue related to detailed design phase	**3.7**
5	Issue related to coding phase	**4.6**
6	Issue related to integration phase	**5.5**
7	Issue related to unit and integration testing	**6.4**
8	Issue related to validation of the technical specification	**7.3**
9	Issue related to final analyzes (e.g., structural coverage analysis)	**8.2**
10	Issue related to validation of the requirements baseline	**9.1**
11	Issue related to delivery and acceptance phase	**10.0**
12	Issue related to most of or all phases	**5.5**

Note1: applicable only for Space domain

Note2: The case 12 captures those issues that impact or are applicable to various phases, for example, some deficiencies in SQA process or configuration control.

Table-7.9: Quantitative values for metric M6 “*amount of artifacts impacted by the issue*”

Case	Description	Measure
1	No impact or negligible	**0**
		1
2	Low impact, under control	**2**
		3
3	Medium impact, demanding some attention	**4**
		5
		6
4	High impact, raising concerns	**7**
		8
5	Very high impact, can be unacceptable	**9**
		10
6	Default value, requiring further Company investigation	**5**

Note: The case 6 (default value) is used when it is not possible to do any estimation at the time the issue is raised, and depends on further Company investigation. It is assumed the average value as default, which may change during the issue follow-up.

Table-7.10: Quantitative values for metric MA7 “*adequacy of issue regarding to audit stage*”

Case	Description	Measure
1	Issue identified in adequate audit Stage	**1.00**
2	Stage#1 scope issue identified in Stage#2 Stage#2 scope issue identified in Stage#3 Stage#3 scope issue identified in Stage#4	**1.06**
3	Stage#1 scope issue identified in Stage#3 Stage#2 scope issue identified in Stage#4	**1.13**
4	Stage#1 scope issue identified in Stage#4	**1.20**

Note: applicable only for Aviation domain

Table-7.11: Metric MS7, "adequacy of issue regarding to audit stage" adjusted for space

Case	Description	Measure
1	Issue identified in adequate audit Stage	**1.00**
2	Stage#1 scope issue identified in Stage#2 Stage#2 scope issue identified in Stage#3 Stage#3 scope issue identified in Stage#4 Stage#4 scope issue identified in Stage#5	**1.05**
3	Stage#1 scope issue identified in Stage#3 Stage#2 scope issue identified in Stage#4 Stage#3 scope issue identified in Stage#5	**1.10**
4	Stage#1 scope issue identified in Stage#4 Stage#2 scope issue identified in Stage#5	**1.15**
5	Stage#1 scope issue identified in Stage#5	**1.20**

Note: applicable only for Space domain

7.4 - Example of use of metrics in Civil Aviation

The following example illustrates the metrics applied to an issue raised in civil aviation software audit:

Issue description: (audit stage = Stage#2) "*The source code that traces to the stack monitor requirement (ReqId-2574) does not implement the function in the same step sequence specified by the requirement. Although in this case the functional implementation is compliant to the requirement, Company-X is requested to investigate if there are other cases in order to identify whether or not it is a systemic issue.*"

Does the issue exercise the following metrics?

a. M1, document evaluation? No, it is not an issue related to document evaluation, but process or adherence problems, i.e., according to the issue, "*The source code ...does not implement the function in the same step sequence specified by the requirement*". (Table-7.1 not applicable, measurement = 0.0);

b. M2, purpose of the issue? Yes, clearly it has the purpose to request additional information, i.e., according to the issue, "*Company-X is requested to investigate ...*" (Table-7.4, case 4, measurement = 75%);

c. M3, type of artifact impacted? Yes, the issue describes a case of impact in the source code (Table-7.5, case 9, measurement = 10.0);

d. M4, root cause? Not yet, because it is not a case of non-compliance, i.e., according to the issue "*the functional implementation is compliant to the requirement*". But depending on the result of the Company-X further investigation, it may become a non-compliance case and will require identification of the root cause, if related to process adherence (Table-7.6 not applicable yet, measurement = 0.0);

e. MA5, distance to the final product? Yes, the distance to the executable code can be identified: the problem is in the source code (Table-7.7, case 5, measurement = 5.5);

f. M6, amount of artifacts impacted? Yes, it can be estimated by the auditor because the concern is applicable to all requirements that specify some algorithm, and the amount can be provided by the Company-X during the audit (Table-7.9, case to be selected by the auditor, let's suppose measurement = 3.0);

g. MA7, adequacy regarding to the audit stage? Yes, the issue has been raised during the Stage#2 and it is related to the coding phase, which is in the Stage#2 scope (Table-7.10, case 1, measurement = 1.0).

The issue in the above example can exercise the metrics M2, M3, M5, M6 and M7. The equation-7.2 calculates the severity of the issue of the example above, where the measurements are:

M(2) = 75%, M(3) = 10, M(4) = 0.0, M(5) = 5.5, M(6) = 3.0, M(7) = 1.0

And the weights by applying table-7.3 are:

W(3) = 23%, W(4) = 28%, W(5) = 20%, W(6) = 29%

Substituting the values in the equation, the final measurement m of the above issue is:

m = (75% * (23% * 10.0 + 28% * 0.0 + 20% * 5.5 + 29% * 3.0)) * 1.0 = 3.2

In order to calculate the measurement of the whole stage (Stage#2), it should apply the equation-7.3, which adds-up the measurements of all issues identified in that stage.

7.5 - Example of use of metrics in Space

The following example illustrates the application of the metrics to an issue raised during the

QSEE simulated audit:

Issue#12, Stage#2: "*The requirement RFUND007 (requirements baseline - RB) is too generic and traces down to several requirements from the technical specification (TS). Some of those TS requirements have information that should have been specified at the RB level as they are clearly at system scope, but that information does not exist at upper level. (non-compliance to ECSS-E-ST-40C, sections 5.2.2.1 and 5.8.3.2, and ECSS-Q-ST-80C, section 6.3.2.1 and 6.3.2.4*)"

Does the issue exercise the following metrics?

a. M1, document evaluation? No, it is not an issue related to document evaluation, but process or adherence problems, i.e., according to the issue, *the requirement is too generic and TS requirements have information that should have been specified at system level.* (Table-7.1 not applicable, measurement = 0.0);

b. M2, purpose of the issue? Yes, clearly it has the purpose to record a non-compliance, i.e., according to the issue, non-compliance to ECSS-E-ST-40C and ECSS-Q-ST-80C (Table-7.4, case 5, measurement = 100%);

c. M3, type of artifact impacted? Yes, the issue describes a case of impact in the RB requirement and TS requirement (Table-7.5, case 9, measurement = 10.0);

d. M4, root cause? Yes, because it is a case of non-compliance related to process adherence. But, as the result of the Company-X further investigation will identify the root cause, the M4 measurement will be the default value (Table-7.6 default measurement = 6.2);

e. MS5, distance to the final product? Yes, the distance to the executable code can be identified: the problem is in the RB requirement and TS requirements (Table-7.7, case 3, measurement = 2.8);

f. M6, amount of artifacts impacted? Yes, it can be estimated by the auditor. However, for this specific issue the estimation depends on further Company investigation. Therefore, a default case is selected (Table-7.9, default measurement = 5);

g. MS7, adequacy regarding to the audit stage? Yes, the issue has been raised during the Stage#2, but it is related to both, the RB requirement (stage#1, most severe case) and TS requirement (stage#2) (Table-7.11, case 2, measurement = 1.05).

The issue in the above example can exercise the metrics M2, M3, M4, M5, M6 and M7. The equation-7.2 calculates the severity of the issue of the example above, where the measurements are:

M(2) = 100%, M(3) = 10, M(4) = 6.2, M(5) = 2.8, M(6) = 5.0, M(7) = 1.05

And the weights by applying table-7.3 are:

W(3) = 23%, W(4) = 28%, W(5) = 20%, W(6) = 29%

Substituting the values in the equation, the final measurement m is:

m = (100% * (23% * 10.0 + 28% * 6.2 + 20% * 2.8 + 29% * 5.0)) * 1.05 = 6.35

In order to calculate the measurement of the whole stage (Stage#2), it should apply the following equations:

a. Equation-7.1, which adds-up the measurements of all issues related to documents evaluation identified in that stage;

b. Equation-7.3, which adds-up the measurements of all issues related to process evaluation and process adherence assessment identified in that stage;

c. Equation-7.4, which calculates the measurement of the whole stage, i.e., the final measurement of the audit result.

7.6 - The metrics supporting management decision

The final measurement of the audit result is applied to a decision table for supporting the decision for next steps. The table-7.12 provides an example:

Table-7.12: Decision support table using the measurement of audit result

Audit cases ▶ Risk assessment ▼	**Audit not passed** mAudit >= x	**Passed with concerns** x > mAudit >= y	**Passed with merit** mAudit < y
High risk	-Increase oversight by continuous supervision; -Perform complete follow-up of the issues; -Re-execute the audit;	-Perform complete follow-up of the issues; -Execute next audit stage;	-Perform summarized follow-up of the issues; -Lower to medium-high risk; -Reduce the agenda of the next audit;
Medium-high risk	-Perform periodic meetings; -Perform complete follow-up of the issues; -Raise to high risk; -Re-execute the audit;	-Perform complete follow-up of the issues; -Execute next audit stage;	-Perform summarized follow-up of the issues; -Lower to medium risk; -Combine the next two audit stages;
Medium risk	-Perform complete follow-up of the issues; -Raise to medium-high risk; -Re-execute the audit stage combined with the next stage;	-Perform summarized follow-up of the issues; -Combine the next two audit stages;	-Perform summarized follow-up of the issues; -Skip next audit stage;

The use of the metric is in the context of oversight activities, which is described in appendix-C for the space domain, and in section 3.2.6 for aviation. An initial risk assessment is performed to obtaining the risk level of the supplier and related software. The example of table-7.12 shows three possible risk assessment outputs (high, medium-high and medium), but it could have a fourth output (low-risk) which in the example it is assumed to not demand any audit as oversight activity; therefore, it has been omitted in the table. The risk assessment is continuously evaluated by the measurement of each audit result and may change the risk level. Values of x and y, which determine the three intervals related to possible audit results (not passed, passed with concerns, passed with merit), are more organization-dependent and can be obtained with historical data. For the case of this thesis, the chapter 5 – "metric validation for aeronautics" uses de historical data from ANAC. However, the metric validation does not include the next step activities of the table-7.12, which is in the scope of the organization responsible for the oversight, and intrinsically related to the organization necessities. It is important to mention that the values for x and y may differ according to the audit stage.

The obtained measurement for each issue can also be used to decide for the level of involvement in the follow-up activities of the audit. The table-7.13 provides an example:

Table-7.13: Table to support deciding the level of involvement in audit follow-up

Measurements interval	Level of auditor involvement in follow-up activities
m >= a	Complete follow-up by the auditor – attention required
a > m >= b	Simplified follow-up by the auditor
b > m >= c	Supplier can close the issue and provide summarized visibility to the auditor
m < c	Supplier not required to address the issue – auditor follow-up not need

Remark: Values of a, b and c are also organization-dependent that can be obtained from historical data, and may differ according to the audit stage.

7.7 - Summary of chapter 7

This chapter provided a summary of the Aerospace Metrics in terms of equations and related tables. Examples were also provided for metrics applicability to aviation and space audit issues, and use for management decision support.

8. CONCLUSION

8.1 – Overview

This chapter presents the conclusion of the thesis. It starts with a summary of the work followed by the thesis evaluation, limitation, contribution, future works, and concluding remarks.

8.2 – Summary of the work

This work presented metrics for oversight of software supplier of safety-critical aerospace systems, called "Aerospace Metrics". First, a bibliographic review on related works was performed to ensure the relevance and innovation. Then, the Aerospace Metrics were generated analytically by using the GQM technique combined with the Reason's human error model, and further refined by using the civil aviation past twelve years software audits results, together with a survey with software safety specialists from the civil aviation. For evaluation in aeronautics, the generated Aerospace Metrics were applied to selected cases of aviation software audits, and evaluated against the related software certification history. For evaluation in astronautics, software safety systematic comparison between space and aviation domains was performed to identifying adjustments in both the metrics and the oversight activities due to space specific necessities. As case study in space domain, the adjusted oversight activities were applied to the QSEE project simulating software audits, and the results submitted to the adjusted Aerospace Metrics.

8.3 – Thesis evaluation

The thesis proposition is repeated below:

Considering the presumed inherent risk of systems outsourcing, it is feasible to construct metrics for evaluating oversight's result of software supplier of safety-critical aerospace system, which can be used for managerial decision.

And it is supported by:

a. The use of GQM and the Reason's human error model;
b. Material gathering 12 years of ANAC practical experience in software audits;
c. A software safety systematic comparison between aviation and space;
d. Workshops and surveys with aviation software safety senior specialists;

e. A space project as case study.

For the confirmation of the proposition, the evaluation focused on the three cycles of Design Science (see section 1.7), and is provided in the next subsections. The key points of each cycle are identified in italics and in quotation marks.

8.3.1 – Evaluation on the Relevance Cycle

Concerning the *"inputs from the contextual environment into the research"*, applicable problems and limitations from aerospace were captured in section 1.1, section 3.2.3, and section 3.3.4, and are listed and evaluated for coverage by the thesis as follow:

Problem/Limitation-1: *It is presumed an inherent risk on outsourcing software-critical space system, which demands an oversight of software supplier to identifying project problems and product nonconformities at earlier stages of development.*

An important artifact generated by the thesis is the Space Oversight Framework described in appendix C, which can be used as tool for performing oversight of software supplier. The Framework was used in this thesis to performing simulated audits in the QSEE space project and the result was satisfactory. Although those simulated audits could not be conclusive regarding to identifying problems at earlier stages of development, the Space Oversight Framework was built based on the civil aviation best practices in software audits, which has enough service history to ensure that efficacy. Therefore, the problem/limitation-1 can be considered addressed by this thesis.

Problem/Limitation-2: *In the civil aviation software audit, the criteria used for issue classification are not adequate for evaluating the audit result and may lead to inappropriate interpretations that can adversely affect managerial decisions.*

The Aerospace Metrics of this thesis consider the relevance of the audit issues. The metrics construction and adjustment (see chapter 4) used consolidate technique/model, vast material from civil aviation past audits, and expertise of senior aviation software safety specialists. The metrics were applied to civil aviation past audits (see chapter 5), obtaining better results than the current criteria. Hence, the proposed metrics are believed to be adequate for evaluating the audit result, which address the problem/limitation-2.

Problem/Limitation-3: *The joint reviews described in ECSS-E-ST-40C focus on documents evaluation, rather than process evaluation and process adherence assessment.*

The Space Oversight Framework of this thesis is divided in two sets of activities for distinct purposes: (1) to evaluate the company processes mainly by desktop review of documents that describe those processes; (2) to evaluate the actual implementation of the processes and level of adherence to them by on-site review (audit) at company installation. The evaluation is supported by a set of metrics (see chapter 7) that measure those audits´ results. The framework and metrics are believed to address the identified problem/limitation-3.

Problem/Limitation-4: *The audits described in ECSS-M-ST-10C are in line with the software supplier oversight of this thesis. However, the requirements provided are general and specific guidelines for software audits are lacking.*

The appendix-C describes the Space Oversight Framework, which can be used as tool for performing audits in software supplier. The framework comprises description of the oversight activities (including the audits), procedures and checklists, which can be considered guidelines for software audits, addressing the problem/limitation-4.

Problem/Limitation-5: *For process assessment and improvement, the effort needed to implement the S4S described in ECSS-Q-HB-80-02A may not be adequate considering the current maturity level of Brazilian space industry or small companies in general.*

The Space Oversight Framework is based on civil aviation experience in software audits and has reduced set of activities, which are for assessment of the process quality and adherence. Those activities are driven by samplings of representative process artifacts, do not require compliance to any capability/maturity standard or model, and can be tailored based on DAL, type of space mission, customer-supplier contract, and even based on company size, though not explicitly stated in the framework. The framework characteristics are believed to address the problem/limitation-5.

Concerning the "*research artifacts into environmental field testing*", the ultimate evaluation of the Aerospace Metrics in aeronautics (i.e., aviation) showed better performance than the current parameters usually used (see section 5.5 and 5.6), which satisfies the acceptance criteria described in section 1.8. For astronautics (i.e., space), the case study using the QSEE space project showed results which are qualitatively coherent with the project status (see section 6.4). Moreover, the bibliographic review (see section 2.4) did not find any feature in space domain that could be similar with the proposed Aerospace Metrics. Therefore, it can be stated that the ultimate evaluation also satisfies the acceptance criteria for the space.

Still concerning the "*research artifacts into environmental field testing*", but about the practicality of the application, for the case of civil aviation the use of metrics in past audits described in chapter 5 (section 5.5 and 5.6) was quite straightforward, though a tool integrated to the oversight activities to automate the measurement would be very helpful. Besides, for more representative values to defining the intervals of the decision-support tables (see table-7.12 and table-7.13), the metrics should be applied to all past audits, and also to the forthcoming ones to ensure refining and updating. For the case of space domain, the tailoring and use of the Space Oversight Framework described in chapter 6 (section 6.4 and 6.5) was also straightforward. However, as it was applied to simulated audits on QSEE project adapted for case study, it cannot be conclusive regarding to practicality. For better evaluation of the practicality in space, it deems necessary the use on typical space project that follows the actual schedule. According to section 1.8 (thesis evaluation criteria), due to time constraints inherent to a doctoral program it is possible to demonstrate the practicality of the concepts to a certain level. Therefore, it can be stated that the practicality of the thesis application is considered acceptable, and limitations are described in section 8.4.

8.3.2 – Evaluation on the Rigor Cycle

Concerning the "*theories, methods and domain expertise from the foundations knowledge*", the main ones have been evaluated for concept solidity and considered acceptable, as follow:

a. Essential concepts for critical software (section 2.2): The terms *error*, *fault*, *failure*, *reliability*, *accident*, *hazard*, *risk*, and *safety* are consolidated in the main domains where safety is a concern, but with slight differences that, if properly managed, do not compromise the end result.

b. Software assurance and supplier oversight as a mean to mitigate adverse safety impact (section 2.3): The approach is adopted by the main civil aviation agencies in their respective policies. For space, the document ECSS-M-ST-10C recommends audits for monitoring suppliers.

c. Software safety approach in space domain (section 3.2): The approach is captured in a set of ECSS documents. According to ECSS standards foreword, "*ECSS is a cooperative effort of the European Space Agency, national space agencies and European industry associations for the purpose of developing and maintaining common standards*".

d. Software safety approach in civil aviation domain (section 3.3): The approach is documented in standards from either the RTCA or SAE. The RTCA is a non-profitable corporation and develops consensus-based recommendations on contemporary aviation issues. SAE is a globally active professional association and standards developing organization for engineering professionals in various industries, including aerospace.

e. The GQM technique (section 4.2): It is a popular approach to software metrics by the University of Maryland and the Software Engineering Laboratory at NASA, and well-succeeded in industries for decades.

f. The Reason´s human error model (section 4.2): Since the release of the book in 1990, the model has been used in many scientific works in medicine, aviation, nuclear, automotive, etc. James Reason is also the author of the Swiss Cheese Model (SCM), the accident model adopted by ICAO in aviation.

Concerning the *"past knowledge to ensure research innovation"*, the bibliographic review presented in section 2.4 investigated works on software metrics and related subjects (i.e., software outsourcing, oversight, and compliance demonstration approaches). It was prioritized works from well-known publishers (e.g., IEEE, ACM, Elsevier, Springer), and after extensive investigation no works with same characteristics of the proposed Aerospace Metrics were found. Therefore, it can be stated that the bibliographic review provides enough confidence in the research innovation of this thesis.

Concerning the *"new research knowledge to the knowledge base"*, section 8.5 presents the thesis contribution. The practicality of some contribution could not be fully evaluated, and there are papers still under production process. However, as it is due to time constraints inherent to a doctoral program, the above key point of the Rigor Cycle can be considered addressed by the thesis.

8.3.3 – Evaluation on the Design Cycle

Concerning the *"tighter loop of research activity for the construction and evaluation of design* *artifacts and processes*", the consistency of the research steps used for the design artifacts construction has been evaluated as follow:

a. *The Aerospace Metrics generation (chapter 4):* The use of GQM and Reason's human error model allowed the identification of the metrics candidates, and the analysis of the civil aviation past audits results allowed the concretization of these candidates. In addition, the experience of software safety senior specialists, captured by survey and workshops enabled the assignment of quantitative measures and relevancies to the metrics. The survey and workshops were also able to identify potential dependency among the metrics, which was essential for specifying the metrics equations. The generation process comprised four steps consistently specified, which supported achieving a high pedigree for the metrics.

b. *The metrics evaluation in aeronautics (chapter-5):* The civil aviation past audits results also allowed construction of representative cases of audit issues, which were subsequently submitted to the metrics as well as to the survey for evaluation by the senior specialists. The comparison between the measurements obtained by the metrics and the evaluation by the senior specialists showed an acceptable consistency, with punctual discrepancies passive of adjustments. For ultimate evaluation, the adjusted metrics were applied to selected cases of aviation software audits and the measurements were qualitatively evaluated against the related software certification history, obtaining a consistent result. The evaluation process comprised five steps consistently specified.

c. *The metrics evaluation in astronautics (chapter-6):* The systematic comparison between the two domains was able to identify the necessary adjustments to the space oversight activities and consequent impacts in the metrics. The QSEE project adapted for case study, together with the simulated audits based on civil aviation, generated representative results that made it possible to exercise the metrics comprehensively. The evaluation process comprised five steps consistently specified.

d. *The Aerospace Metrics result (chapter-7):* Actually, that section does not have steps for constructing design artifacts, but rather gathers the final metrics result in terms of tables and equations, as well as examples of use in both domains. The content is consistent with the steps of chapters 4, 5 and 6.

e. *Systematic comparison between aviation and space (appendix-B):* The systematic comparison covered four concerns that were identified by evaluation of existent works on software safety comparison. It comprised five steps that were related to the concerns. The comparison results were classified to facilitate identifying adjustments in space oversight activities and related metrics. As the name suggests, all steps were systematically (and consistently) specified.

f. *A survey with aviation software safety senior specialists (appendix-D)*: The survey as well as the workshops related to it were organized in seven steps, starting with a workshop to disseminate the general idea, and ending with compiling the results and distributing to the participants and stakeholders. Care was taken with the questionnaire to have simplicity and clarity, to not let the instructions or sequence of questions to influence the answer, to allow for a conclusive result compilation, etc. Adopted recommendation on how to perform a survey has been obtained from open material available in internet.

8.4 – Thesis limitation

It was not possible to have a typical space project as case study. The main reason was the timescale incompatibility between the doctoral program and a representative software project, i.e., long and complex enough to exercise all oversight stages. Other reasons that may have contribute, though not clearly experienced, were the difficulties of the outsourced software supplier to grant access to information for research purpose that was not previously agreed by contract, and also possible management resistance due to the nature of the thesis application, whose result have potential to high management impact.

8.5 – Thesis contribution

The thesis contributions are summarized below:

a. Metrics for evaluation of oversight results of software suppliers of safety-critical aerospace systems (see chapter 7);

b. A systematic comparison process between space and civil aviation domains regarding to software safety of embedded systems (see appendix-B);

c. A framework for oversight of software supplier of critical space systems comprising a reduced set of activities to better suit the current maturity level of Brazilian space industry (see appendix-C);

d. Supporting material for software audits improvements comprising training, evaluation and self-evaluation of software safety specialists, as well as alignment of criteria for judgment of audit issues severity and relevance (see appendix-D);

e. The following papers were produced or are in process of:

    i. *"A Framework for Oversight of Software´s Suppliers of Safety-Critical Space Systems Based on Civil Aviation Best Practices"*, which is related to the appendix-C; presented in the International Astronautics Congress - IAC-2016 (SAKUGAWA et al., 2016);

    ii. *"Towards oversight on software suppliers of safety-critical space systems based on aviation best practices*", which is related to the appendix-B; currently under Safety Science journal review process;

    iii. A paper to describe the construction of the Space Oversight Framework by using the Systematic Comparison results, and use of QSEE project as case study;

    iv. A paper to describe the survey performed with software safety specialists from the civil aviation, which is related to the appendix-D;

    v. A paper to describe metrics for evaluation of software audits in civil aviation, which is related to chapters 4 and 5;

8.6 – Future works

The future works related to this thesis are suggested as follow:

a. The Aerospace Metrics and the Space Oversight Framework could be applied to future space projects, but some adjustments are needed. For instance, the relevance and necessity of the thesis application in space was based on the current space scenario (see section 1.1) and aviation best practices and service history. Therefore, a survey with space software specialists should be performed for further evidence, as well as to support any adjustments in metrics and oversight framework. Besides, concerning the practicality of the application, currently there are no collected data at INPE that could be used for determining the intervals of the decision-support tables. A possible solution could be to initially customize the aviation intervals through analysis, and gradually adjust them by applying to forthcoming space projects.

b. The Aerospace Metrics could be applied to future aviation projects, but some adjustments are needed. For instance, the recent revision of the Order 8110.49 has allowed flexibility in conducting software reviews. Therefore, adjustments are necessary to enabling the Aerospace Metrics to support a continuous oversight, a set of SOIs not necessarily in four stages, or even a single full coverage review at the end of development. Besides, examples of normal, merit and hard cases audits were provided in sections 5.5 and 5.6, but the actual intervals of the decision-support tables (table-7.12 and 7.13) were not calculated. In order to obtain those intervals, it is necessary to apply the Aerospace Metrics to all ANAC past software audits, as well as to the forthcoming ones for refinement and updates.

c. The survey generated artifacts that could be further organized to become a tool for software audits improvements in aerospace domain comprising training of current and future software auditors, evaluation and self-evaluation of software auditors, as well as alignment of criteria for judgment of audit issues severity and relevance. It has been also studied the possibility of applying the survey within aviation industry, and the proposed tool would be very useful.

d. Concerning practicality in both domains, an integrated tool could be created to support performing the software audit (e.g., samplings, specific assessments, recording artifacts´ configuration information and non-conformities, checking coverage) and automatically generating the issues measurements by using the Aerospace Metrics. The tool could be configured according to the audit stage and project characteristics, provide an interactive interface with audit-guiding instructions, and be able to generate a report in document format (i.e., meeting minutes) at the end of the audit, as well as a spreadsheet comprising the audit issues for follow-up activities.

e. The work of this thesis could be extended for both domains to hardware in the scope of custom micro-coded device, e.g., FPGA, ASIC, PLD. Similar to software, ANAC also has material on AEH audits and guidelines documents, as well as ANAC and aviation industry expertise. The AEH approach should start with a comparison between software and AEH in aviation domain to identifying the necessary adjustments in software metrics and oversight activities to apply in AEH. Youn and Yi (2014) presents a useful comparison by reviewing and summarizing DO-178B and DO-254. Then, a comparison between aviation and space based on the Systematic Comparison Process of this thesis (appendix-C, but for AEH) should be performed to extend the aviation artifacts to space.

f. The work of this thesis could also be extended for both domains to system engineering (or system assurance), with emphasis in safety and requirements engineering. ANAC also has material on system assurance audits and guidelines documents, though much less than software and AEH. The AEH approach described above may not be applicable, because the main concern seems to be in the abstraction level differences. System assurance has higher abstraction level and concerns on functional and safety aspects captured in requirements and architecture; hence, the requirements validation as well as the safety assessment are of fundamental importance. Software (or AEH) assurance has lower abstraction level and concerns more on the implementation aspects of the technology; hence, the focus is the requirements verification through analysis, reviews and testing.

8.7 – Concluding remarks

The three cycles of Design Science were evaluated and the results obtained are considered enough for satisfying the criteria established in section 1.8. Considering that the three cycles aimed at providing evidences to prove the thesis proposition, it can be concluded that:

> ***Considering the presumed inherent risk of systems outsourcing, it is feasible to construct metrics for evaluating oversight's result of software supplier of safety-critical aerospace system, which can be used form managerial decision.***

The content of this thesis may be partially applicable to any kind of software, not only restricted to safety-critical ones. However, depending on the cost-benefit the effort spent may not be justifiable.

The audit material available at ANAC (along with the software safety expertise) was considered sufficient for the Aerospace Metrics evaluation in aeronautics. However, concerning the practicality of the application it is desirable (though may not be feasible) to extend the evaluation to other relevant certification agencies and aviation industries.

New technologies (e.g., MBD) are increasing their role in aviation software development. Although care was taken for the Aerospace Metrics to be based on properties unaffected by the technology, new technologies may impact some thesis artifacts (e.g., values obtained from the survey, intervals of the decision-support tables), and will demand continuous evaluation to keep those artifacts updated.

There is a wrong perception that software development process is about documentation safely stored "somewhere" in the organization. The more documents, the better the process. Hopefully, this thesis can contribute to change that perception. Software development process should be seen as something alive, dynamic, taking part of everyday activities of those that are producing the software product.

REFERENCES

ALBRECHT, A.J. Measuring application development productivity. In: IBM APPLICATION DEVELOPMENT SYMPOSIUM, 1979, Monterey. **Proceedings…** Monterey, 1979.

AMBROSIO, A. M.; MATTIELLO-FRANCISCO, M. F.; MARTINS, E. An independent software verification and validation process for space-applications. In: CONFERENCE ON SPACE OPERATIONS, 9. (SPACEOPS 2008). Hidelberg. **Proceedings...** 2008. p. 9. (INPE-15303-PRE/10112).

ASTHANA, A., OLIVIERI, J.; Quantifying software reliability and readiness. In: IEEE INTERNATIONAL WORKSHOP TECHNICAL COMMITTEE ON COMMUNICATIONS QUALITY AND RELIABILITY – CQR-2009, Naples, FL, USA. **Proceedings…** Naples: IEEE, 2009.

AXELROD, C.W. Assuring software and hardware security and integrity throughout the supply chain. In: IEEE INTERNATIONAL CONFERENCE ON TECHNOLOGIES FOR HOMELAND SECURITY. Waltham, MA, USA. **Proceedings…** 2011.

BASILI, V.R.; CALDIERA, G.; ROMBACH, D. **The goal question metric approach**. John Wiley & Sons, Inc, 1994.

BAUFRETON, P; BLANQUART, J.P.; BOULANGER, JL.; DELSENV, H.; DERRIEN, JC.; GASSINO, J.; LADIER, G.; LEDINOT, E.; LEEMAN, M.; MACHROUH, J.; QUÉRÉ, P.; RICQUE, B. Multi-domain comparison of safety standards. In: CONGRESS ON EMBEDDED REAL TIME SOFTWARE AND SYSTEM, 5., 2010, Toulouse, France. **Proceedings…** 2010.

BLANQUART, JP.; ASTRUC, JM.; BAUFRETON, P; BOULANGER, JL.; DELSENV, H.; GASSINO, J.; LADIER, G.; LEDINOT, E.; LEEMAN, M.; MACHROUH, J.; QUÉRÉ, P.; RICQUE, B. Criticality categories across safety standards in different domains. CONGRESS ON EMBEDDED REAL TIME SOFTWARE AND SYSTEM, 5., 2012, Toulouse, France. **Proceedings…** 2012.

BOER, R.C.; VLIET, H. Constructing a reading guide for software product audits. In: THE WORKING IEEE/IFIP CONFERENCE ON SOFTWARE ARCHITECTURE, WICSA'07, 2007, Mumbai, India. **Proceedings…** 2007.

CARVALHO, F.; MEIRA, S.R.L.; FREITAS, B.; EULINO, J. Embedded software component quality and certification. In: EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS, SEAA'09, 35., 2009, Patras, Greece. **Proceedings…** 2009.

CECCARELLI, A.; SILVA, N. Qualitative comparison of aerospace standards: an objective approach. In: IEEE INTERNATIONAL SYMPOSIUM ON SOFTWARE RELIABILITY ENGINEERING WORKSHOPS, ISSREW-2013, 2013, Pasadena, CA, USA. **Proceedings…** IEEE, 2013.

CLELAND, G.L.; BLANQUART, J.P.; CARRANZA, J.M.; FROOME, P.K.D.; JONES, C.C.M.; MULLER, J.F. A framework for the software aspects of the safety certification of a space system. In: JOINT ESA-NASA SPACE-FLIGHT SAFETY CONFERENCE, ESTEC, 2002, Noordwijk, NL. **Proceedings…** ESA-NASA, June 2002.

COTRONEO, D.; NATELLA, R. Fault injection for software certification. **IEEE Security & Privacy**, v. 11, n. 4, p. 38-45, July-Aug 2013.

CRUICKSHANK, K.J.; MICHAEL, J.B.; SHING, M.T. A validation metrics framework for safety-critical software-intensive system. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEM OF SYSTEMS ENGINEERING, SoSE-2009, 2009, Albuquerque, MN, USA. **Proceedings…** 2009.

CURY, E.; SAKUGAWA, B.M. Certificação de software embarcado na aviação civil - experiência brasileira. In: **2ª Safety Workshop**, EPUSP, 2004.

DANIELS, D. Thoughts from the DO-178C committee. In: IET INTERNATIONAL CONFERENCE ON SYSTEM SAFETY, 6., 2011, Birmingham, UK. **Proceedings…** IET, 2011. p. 31

DEPARTMENT OF DEFENSE. **DoD-MIL-STD-882D**: standard practice for system safety. USA, 2000.

DODD, I.; HABLI, I. Safety certification of airborne software: an empirical study. **Reliability Engineering and System Safety**, Elsevier, v. 98, n. 1, p. 7-23, Febr. 2012.

DOMIS, D.; FÖRSTER, M.; SÖREN, K.; TRAPP, M. Safety concept trees. ANNUAL RELIABILITY AND MAINTENANCE SYMPOSIUM, RAMS-2009, 2009, Fort Worth, TX, USA. **Proceedings…** 2009.

DUPUY, A.; LEVESON, N. An empirical evaluation of the MC/DC coverage criterion on the HETE-2 satellite software. In: DIGITAL AVIONICS SYSTEMS CONFERENCE, DASC-2000, 19., 2000, Philadelphia, PA, USA. **Proceedings…** 2000.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-E-ST-10C**: space engineering – system engineering general requirements. Noordwijk, The Netherlands, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-E-ST-40C**: space engineering – software. Noordwijk, The Netherlands, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-E-ST-70C:** Space Engineering – Ground Systems and operations. Noordwijk, The Netherlands, 2008.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-M-ST-10C:** project planning and implementation. Noordwijk, The Netherlands, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-M-ST-80C:** risk management. Noordwijk, The Netherlands, 2008.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-P-00A:** standardization policy. Noordwijk, The Netherlands, 2000.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-Q-HB-80-02A:** space product assurance – software process assessment and improvement. Noordwijk, The Netherlands, 2010.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-Q-HB-80-03A:** space product assurance – software dependability and safety. Noordwijk, The Netherlands, 2012.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-Q-HB-80-04A:** space product assurance – software metrication programme definition and implementation. Noordwijk, The Netherlands, 2011.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-Q-ST-30C:** space product assurance – dependability. Noordwijk, The Netherlands, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-Q-ST-40C**: space product assurance – safety. Noordwijk, The Netherlands, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-Q-ST-80C:** space product assurance – software product assurance. Noordwijk, The Netherlands, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-S-ST-00-01C:** ECSS system – glossary of terms. Noordwijk, The Netherlands, 2012.

ESPOSITO, C.; COTRONEO, D.; SILVA, N. Investigation on safety-related standards for critical systems. In: INTERNATIONAL WORKSHOP ON SOFTWARE CERTIFICATION, WoSoCER-2011, 1., 2011, Hiroshima, Japan. **Proceedings…** 2011. P. 49-54.

FALESSI, D.; SABETZADEH, M.; BRIAND, L.; TURELLA, E.; COQ, T.; WALAWEGE, R.K.P. Planning for safety standards compliance: a model-based tool-supported approach; **IEEE Software**, v. 29, n. 3, p. 64-70, May-June 2012.

FEDERAL AVIATION ADMINISTRATION (FAA). OFFICE OF AVIATION RESEARCH. **An investigation of three forms of the Modified Condition Decision Coverage (MCDC) Criterion**. Washington, D.C.: FAA, 2001.

FEDERAL AVIATION ADMINISTRATION (FAA). **System design and analysis** – Advisory Material Joint AC/AMJ 1309. Arsenal Version, 2002.

FEDERAL AVIATION ADMINISTRATION (FAA). **Advisory Circular AC-20-152, RTCA Inc**. document RTCA/DO-254, Design Assurance Guidance for Airborne Electronic Hardware, 2005.

FEDERAL AVIATION ADMINISTRATION (FAA). **Aircraft certification service**. Job Aid, Conducting Software Reviews prior to certification, 2004.

FEDERAL AVIATION ADMINISTRATION (FAA). **Order 8110.49 chg1**. Software Approval Guidelines, 2011.

FEDERAL AVIATION ADMINISTRATION (FAA). **Order 8110.49 chg2**. Software Approval Guidelines, 2017.

FELDT, R.; TORKAR, R.; AHMAD, E.; RAZA, B. Challenges with software verification and validation activities in the space industry. In: INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION- ICST-2010, 3., 2010, Paris, France. Proceedings… 2010.

GERLACH, M.; HILBRICH, R.; WEISSLEDER, S. Can cars fly? From avionics to automotive: comparability of domain specific safety standards. In: EMBEDDED WORLD CONFERENCE, 2011, Nuremberg, Germany. **Proceedings… 2011.**

HAVELUND, K.; HOLZMANN, G.J. Software certification – coding, code, and coders. In: ACM INTERNATIONAL CONFERENCE ON EMBEDDED SOFTWARE, EMSOFT-2011, 9., 2011, Taipei, Taiwan. **Proceedings…** Taipei, Taiwan: Association for Computing Machinery – ACM, 2011. p. 205-210.

HEVNER, A.R. A three-cycle view of design science research. **Scandinavian Journal of Information Systems**, University of South Florida, USA, v. 19, n. 2, article 4, 2007, p. 87-92.

HILL, J.; TILLEY, S. Creating safety requirements traceability for assuring and recertifying legacy safety-critical systems. In: IEEE INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE, NE-2010, 18., 2010, Sydney, NSW, Australia. **Proceedings…** IEEE, 2010.

HILL, J.; VICTOR, D. The product engineering class in the software safety risk taxonomy for building safety-critical systems. In: AUSTRALIAN CONFERENCE ON SOFTWARE ENGINEERING, ASWEC-2008, 19., 2008, Perth, WA, Australia. **Proceedings…** 2008.

HOWDEN, W.E. error models and software certification. In: INTERNATIONAL WORKSHOP ON SOFTWARE CERTIFICATION, WoSoCER-2011, 1., 2011, Hiroshima, Japan. **Proceedings…** 2011.

THEINTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO-26262** - road vehicles -- functional safety. International Standard ISO/FDIS 26262, 2011.

THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC-15504** - information technology – process assessment. Geneva, Switzerland, 2004.

JACKLIN, S.A. Certification of safety-critical software under DO-178C and DO-278A. In: AIAA INFOTECH@ AEROSPACE CONFERENCE, June 2012, Garden Grove, CA, USA. **Proceedings…** AIAA, 2012.

KORNECKI, A.; ZALEWSKI, J. Software certification for safety-critical systems: a status report. INTERNATIONAL MULTICONFERENCE ON COMPUTER SCIENCE AND INFORMATION TECHNOLOGY, 2008, Wisła, Poland. **Proceedings…** IEEE, 2008. P. 665-672.

KUMAR, N.; KOLEY, J.; KRISHNAMURTHY, P.R.; RAO, S.N. Regulatory review of computer based systems: Indian perspectives. INTERNATIONAL CONFERENCE ON RELIABILITY, SAFETY AND HAZARD (ICRESH-2010), 2010, Mumbai, India. **Proceedings…** IEEE, 2010.

LAHOZ, C. H. N.; ROMANI, M. A. S.; Yano, E. T. Dependability attributes for space computer systems: quality factors approach. In: SPACE OPERATIONS CONFERENCE, SPACEOPS, 12., 2012, Stockholm. **Proceedings…** ALTEC, 2012.

LAYMAN, L.; BASILI, V.R.; ZELKOWITZ M.V.; FISHER, K.L. A case study of measuring process risk for early insights into software safety. IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE-2011), 33., 2011, Honolulu, HI, USA. Proceedings…IEEE, 2011.

LEDINOT, E.; GASSINO, J.; BLANQUART, JP.; BOULANGER, JL.; QUÉRÉ, P.; RICQUE, B. A cross-domain comparison of software development assurance. In: CONGRESS ON EMBEDDED REAL TIME SOFTWARE AND SYSTEM, ERTS^2-2012, 6., 1-3 February 2012, Toulouse, France. **Proceedings… 2012.**

LEMES, M.J.R. et al. Certificação de software embarcado de emprego aeronáutico: processo e desafios. In: **1º Safety Workshop**. EPUSP, 2003.

LEVESON, N.G. **Safeware -** system safety and computers. University of Washington, Addison-Wesley, 1995.

LEVESON, N.G. **White paper on approaches to safety engineering**. Massachusetts Institute of Technology, 2003.

LEVESON, N.G. The role of software in spacecraft accidents. **Journal of Spacecraft and Rockets**, American Institute of Aeronautics and Astronautics, v. 41, n. 4, p. 564-575, July 2004.

LEVESON, N.G. A systems-theoretic approach to safety in software-intensive systems. **IEEE Transactions on Dependable and Secure Computing**, v.1, n.1, p. 66-86, Jan-Mar. 2005.

LUTZ, R.; HINE, A.P. Using fault modeling in safety cases. In: INTERNATIONAL SYMPOSIUM ON SOFTWARE RELIABILITY ENGINEERING, ISSRE-2008, 19., Seattle, WA, USA. **Proceedings…** 2008.

McCABE, T.J. A complexity measure. **IEEE Transaction on Software Engineering**, v. SE-2, n. 4, p. 308–320, Dec. 1976.

MACHROUH, J.; BLANQUART, JP.; BAUFRETON, P.; BOULANGER, JL.; DELSENY, H.; GASSINO, J.; LADIER, G.; LEDINOT, E.; LEEMAN, M.; ASTRUC, JM.; QUÉRÉ, P.; RICQUE, B. Cross domain comparison of system assurance. In: CONGRESS ON EMBEDDED REAL TIME SOFTWARE AND SYSTEM, ERTS^2-2012, 6., 2012, 1-3 Feb. 2012, Toulouse, France. **Proceedings…** 2012.

MARQUES, J.C.; YELISETTY, S.M.H.; DIAS, L.A.V.; CUNHA, A.M. Using model-based development as software low-level requirements to achieve airborne software certification. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY- NEW GENERATIONS, 2012, 9., 2012, Las Vegas, NV, USA. **Proceedings…** 2012.

MARTIN, L.; SCHATALOV, M.; HAGNER, M.; GOLTZ, U.; MAINBAUM, O. A methodology for model-based development and automated verification of software for aerospace systems. In: IEEE AEROSPACE CONFERENCE, 2013, Big Sky, MT, USA. **Proceedings…** 2013.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY (MIT). **Accidents report website**. Available in: < http://sunnyday.mit.edu/accidents/index.html>.

MATTIELLO-FRANCISCO, M.F.; ARIAS, R., HIRATA, C.M.; YANO, E.T.; SAKUGAWA, B.M. A Comparative study between PMBoK/DoD and ECSS management process for software acquisition. DATA SYSTEM IN AEROESPACE CONFERENCE, 9. (DASIA)., 2005, Edinburg, Scotland. **Proceedings...** 2005. p. 17-21. (INPE-14078-PRE/9247).

MATTIELLO-FRANCISCO, M.F; SAKUGAWA, B.M.; YANO, E.T. Safety in a Web-based satellite flight plan supporting system. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS, 9., 2006, Rome. **Proceedings…** Rome, Italy: American Institute of Aeronautics and Astronautics, 2006,

MATTIELLO-FRANCISCO, M. F.; SANTIAGO, V.; AMBROSIO, A.M.; JOGAIB, L.; COSTA, R. A Brazilian software industry experience in using ECSS for Space Application Software Development. ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 14. (CE 2007), São José dos Campos. **Proceedings...** São José dos Campos: Springer, 2007. p. 163-170. CD-ROM; On-line. ISBN 978-1-84628-975-0. (INPE-15260-PRE/10081).

McDERMID, J.A. Software safety: where´s the evidence? In: AUSTRALIAN WORKSHOP ON SAFETY-CRITICAL SYSTEMS AND SOFTWARE, 6., 2001, Brisbane, Australia. **Proceedings…** Brisbane, Australia: ACM Digital Library, 2001. v. 3, p. 1-6.

MICHAEL, J.B.; SHING, M.T.; CRUICKSHANK, K.J.; REDMOND, P.J. Hazard analysis and validation metrics framework for system of systems software safety. **IEEE Systems Journal**, v.4, n. 2, June 2010.

MINISTRY OF DEFENCE. DEFENCE STANDARD. **DEF STAN 00-55** - Requirements for safety related software in defence equipment; August 1997.

NAIR, S.; DELAVARA, J.L.; SABETZADEH, M.; BRIAND, L. Classification, structuring, and assessment of evidence for safety: a systematic literature review. In: INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION, ICST-2013, 6., 2013, Luxembourg, Luxembourg. **Proceedings…** 2013.

NASA LANGLEY RESEARCH CENTER. **Software safety**. NASA, 1997. Available in: <http://satc.gsfc.nasa.gov/assure/distasst.pdf>

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **NASA-GB-8719.13**, software safety guidebook. NASA, 2004.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **NASA-STD-8719.13B**, software safety standard. NASA, 2004.

THE NATIONAL AVIATION SAFETY DATA ANALYSIS CENTER, NASDAC website, **FAA Accident and Incident Data System (AIDS)**, Available in: http://www.nasdac.faa.gov. Access on 2017.

National Transportation Safety Board (NTSB) website, **Aviation Accident Database**. Available in: <http://www.ntsb.gov>. Access on 2017.

OWENS, B.D.; HERRING, M.S.; DULAC, N.; LEVESON, N.G.; INGHAM, M.D.; WEISS, K.A. Application of a safety-driven design methodology to an outer planet exploration. In: IEEE AEROSPACE CONFERENCE, 2007, Big Sky, MT, USA. **Proceedings…** IEEE, 2007

PLEEGER, S. L. Software metrics: progress after 25 years? **IEEE Software**, v. 25, n. 6, p. 32-34, Nov/Dec 2008.

AGÊNCIA ESPACIAL BRASILEIRA (AEB), MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO (MCTI). **Programa Nacional de Atividades Espaciais** - PNAE : 2012 – 2021. Brasília, 2012**.**

PETERSON, K. **A wing and a prayer:** outsourcing at Boeing. Thomson Reuters special report, 2011.

PONTES, P. R.; AMBROSIO, A.M.; VILLANI, E. Contributions of model checking and CoFI methodology to the development of space embedded software. **Empirical Software Engineering**, v. 19, n. 1, p. 39-68, Springer, Feb. 2014.

PRESSMAN, R. S.; MAXIM, B. R. **Software engineering** – a practitioner´s approach. 8. ed. McGraw-Hill, 2015.

REASON, J. **Human error**. Cambridge UK, Cambridge University Press, 1990.

REGAN, G.; McCAFERY, F.; McDAID, K.; FLOOD, D. Traceability-Why do it? In: INTERNATIONAL CONFERENCE ON SOFTWARE PROCESS IMPROVEMENT AND CAPABILITY DETERMINATION, 17., 2012, Palma, Spain. **Proceedings…** 2012. p. 161-172.

RIERSON, L. **Developing safety-critical software**. CRC Press, Taylor & Francis Group, 2013.

RODRÍGUEZ, M.; PIATTINI, M. Systematic review of software product certification. IBERIAN CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY, CISTI, 7., 2012, Madrid, Spain. **Proceedings…** 2012.

ROMANSKI, G. Combined safety and security certification. In: IET INTERNATIONAL CONFERENCE ON SYSTEM SAFETY, INCORPORATING THE CYBER SECURITY CONFERENCE, 7., 2012, Edinburgh, UK. **Proceedings…** 2012.

RTCA, INC. **Special Committee SC-205**: software considerations in airborne systems. Washington, DC, 2008.

RTCA, INC. **RTCA/DO-178B**: software considerations in airborne systems and equipment certification. Washington, DC, 1992.

RTCA, INC. **RTCA/DO-178C**: software considerations in airborne systems and equipment certification. Washington, DC, 2011.

RTCA, INC. **RTCA/DO-248C**: supporting information for DO-178C and DO-278A. Washington, DC, 2011.

RTCA, INC. **RTCA/DO-254**: design assurance guidance for airborne electronic hardware. Washington, DC, 2000.

RTCA, INC. **RTCA/DO-278A**: software integrity assurance considerations for Communication, Navigation, Surveillance and air traffic management (CNS/ATM) systems. Washington, DC, 2011.

RTCA, INC. **RTCA/DO-297**: Integrated Modular Avionics (IMA) development guidance and certification consideration. Washington, DC, 2005.

RTCA, INC. **RTCA/DO-330:** software tool qualification considerations. Washington, DC, 2011.

RTCA, INC. **RTCA/DO-331:** model-based development and verification supplement to DO-178C and DO-278A. Washington, DC, 2011.

RTCA, INC. **RTCA/DO-332:** object-oriented technology and related techniques supplement to DO-178C and DO-278A. Washington, DC, 2011.

RTCA, INC. **RTCA/DO-333:** formal methods supplement to DO-178C and DO-278A. Washington, DC, 2011.

RUSHBY, J. **Partitioning in avionics architecture: requirements, mechanism, and assurance**. NASA Langley Technical Report, ACM Digital Library, 1999.

THE SOCIETY OF AUTOMOTIVE ENGINEERS, AEROSPACE RECOMMENDED PRACTICE. **SAE/ARP4754A**: guidelines for development of civil aircraft and systems, 2010.

THE SOCIETY OF AUTOMOTIVE ENGINEERS, AEROSPACE RECOMMENDED PRACTICE. **SAE/ARP4761**: guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, 1996.

SAKUGAWA, B.M.; YANO, E.T.; CURY, E. Airborne software concerns in civil aviation certification. In: LATIN AMERICAN SYMPOSIUM ON DEPENDABLE COMPUTING – LADC, 2., 2005, Salvador, Brazil. **Proceedings…** Springer-Verlag, 2005,

SAKUGAWA, B.M.; AMBROSIO, A.M.; LOUREIRO, G.; LAHOZ, C.H.N.; A Framework for oversight of software's suppliers of safety-critical space systems based on civil aviation best practices. In: THE INTERNATIONAL ASTRONAUTICAL CONGRESS, IAC-2016, 2016, Guadalajara, Mexico. **Proceedings…** 2016.

SANTIAGO, V.; MATTIELLO-FRANCISCO, M. F.; COSTA, R.; SILVA, W. P.; AMBROSIO, A. M. QSEE project: an experience in outsourcing software development for space applications. INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING (SEKE'07), 9., 2007, Boston. **Proceedings...** 2007.

SCHUMANN, J.; DENNEY, E. Automatic generation of certifiable space communication software. In: IEEE AEROSPACE CONFERENCE, 2007, Big Sky, MT, USA. **Proceedings…** IEEE, 2007.

SHARMA, D.C. Indian IT outsourcing industry: future threats and challenges. **The journal of policy, planning and futures studies**. Elsevier, v. 56, p. 73-80, Feb. 2014.

SHARMA, A., KUSHWAHA, D.S. A Metric suite for early estimation of software testing effort using requirement engineering document and its validation. In: INTERNATIONAL CONFERENCE ON COMPUTER & COMMUNICATION TECHNOLOGY (ICCCT), 2., 2011, Allahabad, India. **Proceedings…** 2011.

SIMON, H. The Sciences of Artificial, 3. ed. Cambridge, MA: MIT Press, 1996.

SOZEN, N., MERLO, E. Adapting software product lines for complex certifiable avionics software. In: INTERNATIONAL WORKSHOP ON PRODUCT LINE APPROACHES IN SOFTWARE ENGINEERING, 3., PLEASE-2012, Zurich, Switzerland. **Proceedings…** 2012.

SPENDLA, L.; TANUSKA, P.; SMOLARIK, L. Metric proposal for system testing model verification for safety critical systems. In: IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT SYSTEMS AND INFORMATICs (SISY), 11., 2013, Subotica, Serbia. **Proceedings…** IEEE, 2013.

STENSRUD, E.; SKRAMSTAD, T.; LI, J., XIE, J. Towards goal-based software safety certification based on prescriptive standards. In: INTERNATIONAL WORKSHOP ON SOFTWARE CERTIFICATION, WoSoCER,2011, 1., 2011, Hiroshima, Japan. **Proceedings…** 2011.

STETSON, H.K.; KNICKERBOCKER, G.; CRUZEN, C.A.; HADDOCK, A.T. The HAL 9000 space operating system. In: IEEE AEROSPACE CONFERENCE, 2012, Big Sky, MT, USA. **Proceedings…** 2012.

STOREY, N. **Safety-critical computer systems**. Addison-Wesley Longman, 1996.

TOKGOZ, N.; ERDOGAN, D. Information technology outsourcing reasons in aviation industry. In: INTERNATIONAL BUSINESS RESEARCH CONFERENCE, 39., Dec 2016, Tokyo, Japan. **Proceedings…** 2016.

VAN AKEN, J.E.; ROMME, G. Reinventing the future: adding design science to the repertoire of organization and management studies. **Organization Management Journal**, v. 6, n. 1, p. 4-12, 2009.

VAN AKEN, J.E., ROMME, A.G.L. A design science approach to evidence-based management. In: D. ROUSSEAU (ed.). **The Oxford handbook of evidence-based management**, Oxford: Oxford University Press, 2012. p. 43-57.

VIEIRA, M.; MADEIRA, H.; CRUZ, S.; COSTA, M.; CUNHA, J.C. Integrating GQM and data warehousing for the definition of software reuse metrics. In: IEEE SOFTWARE ENGINEERING WORKSHOP, 34., 2011, Limerick, Ireland. **Proceedings…** IEEE, 2011.

WALAWEGE, R.K.P.; SABETZADEH, M.; BRIAND, L.; COQ, T. Characterizing the chain of evidence for software safety cases: a conceptual model based on the IEC 61508 standard. In: INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION, ICST-2010, 3., 2010, Paris, France. **Proceedings…** 2010.

WEAVER, R.A. **The safety of software** – constructing and assuring arguments. PhD thesis, University of York, Department of Computer Science, September 2003.

WONG, W.E.; DEMEL, A.; DEBROY, V.; SIOK, M.F. Safe software: does it cost more to develop? In: IEEE INTERNATIONAL CONFERENCE ON SECURE SOFTWARE INTEGRATION AND RELIABILITY IMPROVEMENT (SSIRI-2011), 5., 2011, Jeju Island, South Korea. **Proceedings…** IEEE, 2011. P. 198-207.

WONG, W.E.; GIDVANI, T.; LOPEZ, A.; GAO, R.; HORN, M. Evaluating software safety standards: a systematic review and comparison. In: IEEE INTERNATIONAL CONFERENCE SOFTWARE SECURITY AND RELIABILITY-COMPANION (SERE-C), 80., 2014, San Francisco, CA, USA. **Proceedings…** IEEE, 2014.

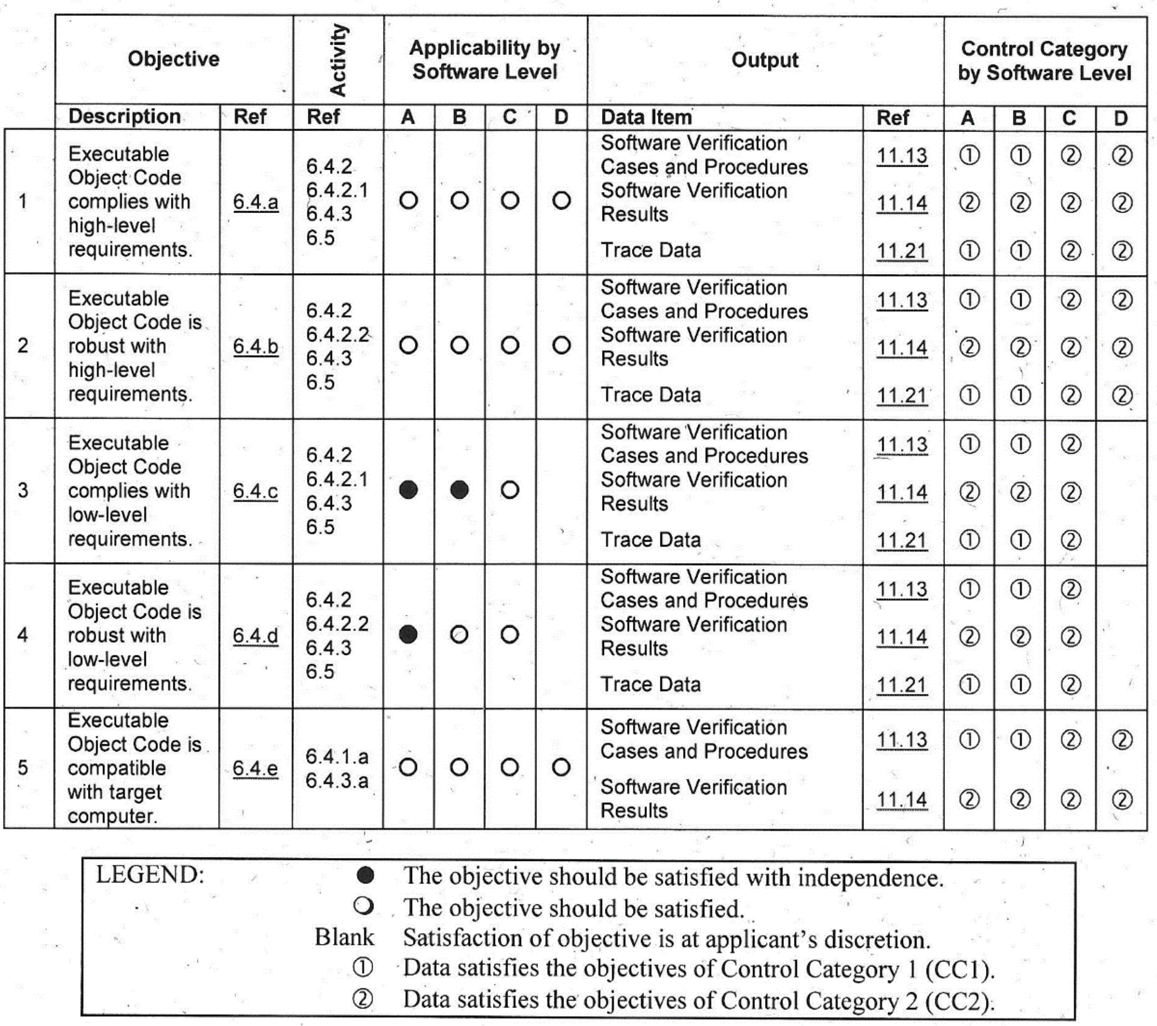
YAJING, M.; DEYING, F. Risk factors analysis of IT outsourcing from the distance-based perspective. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE, ICSESS-2011, 2., 2011, Beijing, China. **Proceedings…** IEEE, 2011. p. 78-87.

YOUN, W.; YI, B.J. Software and hardware certification of safety-critical avionic systems: A comparison study. **Computer Standards & Interfaces**, Elsevier, v. 36, n. 6, p. 889-898, 2017.

APPENDIX A: SUMMARY OF DO-178C OBJECTIVES

This appendix provides a brief description of the ten tables (A-1 to A-10) of DO-178C, Annex A, which contain a summary of the objectives to achieve. As an example, Table A-6 is presented in full and with explanation of its fields, as illustrated in the figure-A.1:

Figure-A.1: DO-178C, Table-A.6, Testing of Outputs of Integration Process



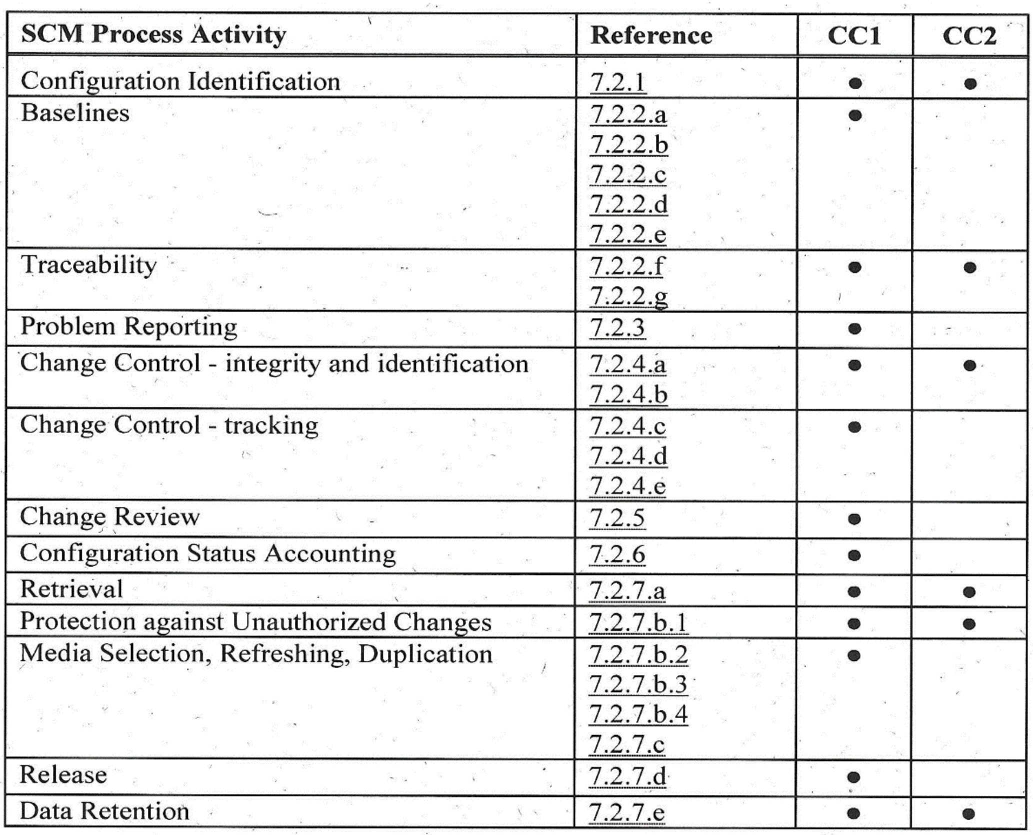
	Objective		Activity	Applicability by Software Level				Output		Control Category by Software Level			
	Description	Ref	Ref	A	B	C	D	Data Item	Ref	A	B	C	D
1	Executable Object Code complies with high-level requirements.	6.4.a	6.4.2 6.4.2.1 6.4.3 6.5	○	○	○	○	Software Verification Cases and Procedures	11.13	①	①	②	②
								Software Verification Results	11.14	②	②	②	②
								Trace Data	11.21	①	①	②	②
2	Executable Object Code is robust with high-level requirements.	6.4.b	6.4.2 6.4.2.2 6.4.3 6.5	○	○	○	○	Software Verification Cases and Procedures	11.13	①	①	②	②
								Software Verification Results	11.14	②	②	②	②
								Trace Data	11.21	①	①	②	②
3	Executable Object Code complies with low-level requirements.	6.4.c	6.4.2 6.4.2.1 6.4.3 6.5	●	●	○		Software Verification Cases and Procedures	11.13	①	①	②	
								Software Verification Results	11.14	②	②	②	
								Trace Data	11.21	①	①	②	
4	Executable Object Code is robust with low-level requirements.	6.4.d	6.4.2 6.4.2.2 6.4.3 6.5	●	○	○		Software Verification Cases and Procedures	11.13	①	①	②	
								Software Verification Results	11.14	②	②	②	
								Trace Data	11.21	①	①	②	
5	Executable Object Code is compatible with target computer.	6.4.e	6.4.1.a 6.4.3.a	○	○	○	○	Software Verification Cases and Procedures	11.13	①	①	②	②
								Software Verification Results	11.14	②	②	②	②

LEGEND:		
	●	The objective should be satisfied with independence.
	○	The objective should be satisfied.
	Blank	Satisfaction of objective is at applicant's discretion.
	①	Data satisfies the objectives of Control Category 1 (CC1).
	②	Data satisfies the objectives of Control Category 2 (CC2).

Source: extracted from *DO-178C*

Table A-6 contains 5 objectives to achieve. These objectives are achieved through testing by using the executable program (see "description" column), where the test cases are generated from the requirements (HLR and LLR) for normal or abnormal situations, and to meet the objective 5 the program must run on the target computer. The "Ref." column references the sections in the standard that describe the objectives. The “Activity” column references the sections that describe the activities necessary to meet the objectives. The column "Applicability by Software Level" indicates, for DAL A to D, if the objective is or is not required and with or without independence. For the output, the "description" column summarizes the life cycle data and the “Ref.” column indicates the related section. The column "Control Category by SW Level", for DAL A to D, indicates the control rigor of the configuration items, according to the criterion established in figure-A.2 below:

Figure-A.2: DO-178C, Table-7-1, SCM Process Associated with CC1 and CC2 Data



SCM Process Activity	Reference	CC1	CC2
Configuration Identification	7.2.1	●	●
Baselines	7.2.2.a 7.2.2.b 7.2.2.c 7.2.2.d 7.2.2.e	●	
Traceability	7.2.2.f 7.2.2.g	●	●
Problem Reporting	7.2.3	●	
Change Control - integrity and identification	7.2.4.a 7.2.4.b	●	●
Change Control - tracking	7.2.4.c 7.2.4.d 7.2.4.e	●	
Change Review	7.2.5	●	
Configuration Status Accounting	7.2.6	●	
Retrieval	7.2.7.a	●	●
Protection against Unauthorized Changes	7.2.7.b.1	●	●
Media Selection, Refreshing, Duplication	7.2.7.b.2 7.2.7.b.3 7.2.7.b.4 7.2.7.c	●	
Release	7.2.7.d	●	
Data Retention	7.2.7.e	●	●

Source: extracted from *DO-178C* (2011]

The remainder of this appendix provides a brief description of the objectives of the other tables in Annex A of the DO-178C. Table A-1 contains 7 objectives to achieve during the software planning process, as described below:

1. The activities of the software life cycle processes are defined;
2. The software life cycle(s), including the inter-relationship between the processes, their sequencing, feedback mechanisms, and transition criteria, is defined;
3. Software life cycle environment is selected and defined;
4. Additional considerations are addressed;

5. Software development standards are defined;
6. Software plans comply with this document;
7. Development and revision of software plans are coordinated.

Table A-2 contains 7 objectives to achieve during the software development process, as described below:

1. High-level requirements are developed;
2. Derived high-level requirements are defined and provided to the system processes, including the system safety assessment process;
3. Software architecture is developed;
4. Low-level requirements are developed;
5. Derived low-level requirements are defined and provided to the system processes, including the system safety assessment process;
6. Source code is developed;
7. Executable Object Code and Parameter Data Item Files, if any, are produced and loaded in the target computer.

Table A-3 contains 7 objectives to achieve during the verification of outputs of software requirements process, as described below:

1. High-level requirements comply with system requirements;
2. High-level requirements are accurate and consistent;
3. High-level requirements are compatible with target computer;
4. High-level requirements are verifiable;
5. High-level requirements conform to standards;
6. High-level requirements are traceable to system requirements;
7. Algorithms are accurate.

Table A-4 contains 13 objectives to achieve during the verification of outputs of software design process, as described below:

1. Low-level requirements comply with high-level requirements;

2. Low-level requirements are accurate and consistent;
3. Low-level requirements are compatible with target computer;
4. Low-level requirements are verifiable;
5. Low-level requirements conform to standards;
6. Low-level requirements are traceable to high-level requirements;
7. Algorithms are accurate;
8. Software architecture is compatible with high-level requirements;
9. Software architecture should be consistent;
10. Software architecture is compatible with target computer;
11. Software architecture is verifiable;
12. Software architecture conforms to standards;
13. Software partitioning integrity is confirmed.

Table A-5 contains 9 objectives to achieve during the verification of outputs of software coding and integration process, as described below:

1. Source code complies with low-level requirements;
2. Source code complies with software architecture;
3. Source code is verifiable;
4. Source code conforms to standards;
5. Source code is traceable to low-level requirements;
6. Source code is accurate and consistent;
7. Output of software integration process is correct and complete;
8. Parameter Data Item File is correct and complete;
9. Verification of Parameter Data Item File is achieved.

Table A-6 has already been described at the beginning of this appendix.

Table A-7 contains 9 objectives to achieve during the software verification process, as described below:

1. Test procedures are correct;
2. Test results are correct and discrepancies explained;
3. Test coverage of high-level requirements is achieved;
4. Test coverage of low-level requirements is achieved;

5. Test coverage of software structure (modified condition/decision coverage) is achieved;
6. Test coverage of software structure (decision coverage) is achieved;
7. Test coverage of software structure (statement coverage) is achieved;
8. Test coverage of software structure (data coupling and control coupling) is achieved;
9. Verification of additional code that cannot be traced to Source Code is achieved.

Table A-8 contains 6 objectives to achieve during the software configuration management process, as described below:

1. Configuration items are identified;
2. Baselines and traceability are established;
3. Problem reporting, change control, change review, and configuration status accounting are established;
4. Archive, retrieval, and release are established;
5. Software load control is established;
6. Software life cycle environment control is established.

Table A-9 contains 5 objectives to achieve during the software quality assurance process, as described below:

1. Assurance is obtained that software plans and standards are developed and reviewed for compliance with this document and for consistency;
2. Assurance is obtained that software life cycle processes comply with approved software plans;
3. Assurance is obtained that software life cycle processes comply with approved software standards;
4. Assurance is obtained that transition criteria for the software life cycle processes are satisfied;
5. Assurance is obtained that software conformity review is conducted.

Table A-10 contains 3 objectives to achieve during the Certification Liaison process, as described below:

1. Communication and understanding between the applicant and the certification authority is established;
2. The means of compliance is proposed and agreement with the Plan for Software Aspects of Certification is obtained;
3. Compliance substantiation is provided.

APPENDIX B: SOFTWARE SAFETY - A SYSTEMATIC COMPARISON BETWEEN AVIATION AND SPACE DOMAINS

B.1 - The process description

This appendix presents a systematic comparison between aviation and space domains in the software safety scope focusing on a representative set of standards from both domains. The purpose is to identify reuses of oversight activities from aviation best practices and adjustments due to specific necessities of the space oversight, rather than differences and similarities among standards. Considering the limitations identified in the 13 works on software safety comparison (see section 2.4.5), the *Systematic Comparison Process* must cover the following four concerns:

Concern-1: Ensure domains' comparison at adequate level, regardless of standards scope;

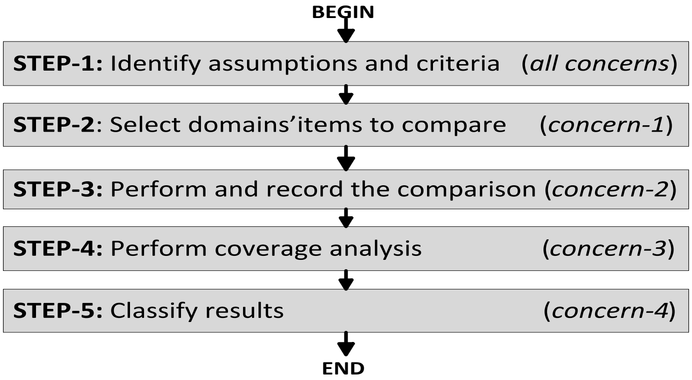
Concern-2: Clearly identify differences and similarities between both domains that impact the level of reuse of aviation best practices;

Concern-3: Ensure software safety coverage of the chosen scope from both domains;

Concern-4: Facilitate identifying reuses and adjustments from aviation.

The *Systematic Comparison Process* comprises 5 steps. Figure-B.1 shows the process and for every step the addressed concerns are indicated.

Figure-B.1: The Systematic Comparison Process


BEGIN
STEP-1: Identify assumptions and criteria (all concerns)
STEP-2: Select domains'items to compare (concern-1)
STEP-3: Perform and record the comparison (concern-2)
STEP-4: Perform coverage analysis (concern-3)
STEP-5: Classify results (concern-4)
END

STEP-1: Identify assumptions and comparison criteria (step related to all concerns)

As result of the evaluation of those 13 selected papers described in section 2.4.5, and also considering that the main goal of the *Systematic Comparison Process* is to investigate the possible reuse of aviation best practices, the following assumptions were identified:

a. There is a correlation between the recommendations provided by a given standard and the oversight activities necessary to assess the company process for the proper implementation of those recommendations. Consequently, if for example two standards have high commonalities between their recommendations, the related oversight activities necessary to assess compliance to those standards will also have high commonalities between them.

b. Aviation and space domains have similar basic approach, which is process-based and with activities commensurate to the assurance level. Moreover, their related standards are semi-formally specified, and the specification items are inter-domains comparable with respect to scope, granularity, level of detail and formalism. Although some aviation standards are objective-based approach while ECSS standards prescribe processes, they are comparable with respect to the oversight activities necessary to assess compliance to them. For example, DO-178C provides specific objectives/activities for software requirements verification, while ECSS specifies a process for the verification of requirements baseline. Some differences exist in form, granularity and even contents, but it is possible to conclude that for verification of software requirements, the *Aviation Oversight* activities used for DO-178C assessment can be reused with few adjustments to assess the related ECSS process.

Regarding to the criteria to be used, initially 184 comparison criteria were identified, grouped by similarities, and refined by removing repetitions, overlaps and subsets. Then, they were classified according to the subject, obtaining a final list with 32 criteria whose summary is provided in Table-B.1.

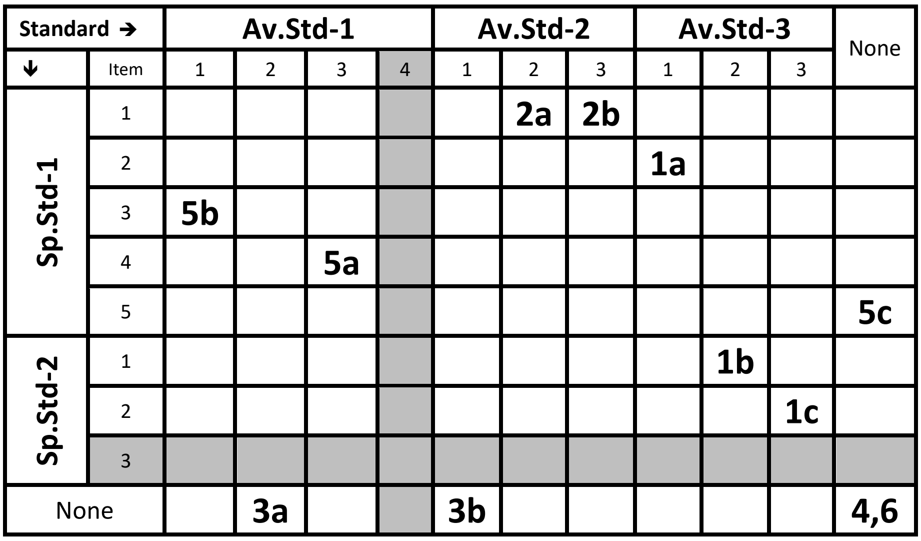
Table-B.1: Comparison criteria for software safety

Classification	Criteria
General characteristic	1- Domain and Standards' organization 2- Standard's level of update and use (e.g., periodically revised, standards widely used) 3- Harmonization of Terminology (e.g., use of common glossary) 4- Level of prescription and confirmation measures (e.g., requires specific technique) 5- Regulation regimes and certification (e.g., mandated by law, recommended)
Safety-Related	6- Level of safety evidence (e.g., use of Software Fault Tree Analysis – SFTA) 7- Assurance/safety level impact on software activities (e.g., defines levels of rigor)
Process-related	8- Lifecycle processes required (e.g., planning, development, verification) 9- Lifecycle data produced (e.g., plans, standards, detailed design, traceability matrix) 10- Level of independence required (e.g., separate organization for verification) 11- Test environment (e.g., real-target, simulated, emulated) 12- Level of Traceability required (e.g., unidirectional, bidirectional, vertical, horizontal)
Product-related	13- Software complexity (e.g., lines of code, cyclomatic complexity, function points) 14- Software portability (e.g., design for reusability) 15- Partitioning (e.g., time partitioning, memory segregation) 16- Use of configuration files and Databases 17- Concerns with unintended functions (i.e., not required but unintentionally implemented) 18- User-Modifiable Software (i.e., able to be modified in the operational environment)
Organization-related	19- Project management activities 20- Stakeholders involved (e.g., customer, developer, user) 21- Qualification and training of personnel 22- Safety benefit vs. Cost tradeoff 23- Lessons Learned
Methods and techniques	24- Overall testing techniques (e.g., black-box, white-box, fault injection) 25- Overall verification methods (e.g., review, analysis, inspection, testing) 26- Specific methods and techniques (e.g., Service history, Reverse engineering)
Integrity concerns	27- Dependability analysis (e.g., Reliability, Availability, Maintenability, Safety) 28- Fault Tolerance Techniques (e.g., detection, recovery, avoidance) 29- Software and Hardware Relationship
Additional concerns	30- Software reuse (e.g., Off-the-shelf – OTS) 31- Tool usage and qualification level 32- Notions of security

STEP-2: Select domains' items to compare (step related to concern-1)

Section B.2 presents the aviation standards related to software safety that were selected based on the *Aviation Oversight* scope. This step selected the set of ECSS standards that covers the scope of the aviation standards, and the set of items for comparison composed by each standard (e.g., ECSS are composed by requirements, and DO-178C by objectives and related activities), and organized them in a spreadsheet for items' association and domain coverage analysis. The organization is shown in Figure-B.2, where aviation domain comprises 3 standards (i.e., Av.Std-1, Av.Std-2, Av.Std-3) with 4, 3 and 3 items respectively (columns), while the space domain comprises 2 standards (i.e., Sp.Std-1, Sp.Std-2) with 5 and 3 items respectively (rows).

Figure-B.2: Simplified example of the spreadsheet for association and coverage analysis



Standard →		Av.Std-1				Av.Std-2			Av.Std-3			None
↓	Item	1	2	3	4	1	2	3	1	2	3	
Sp.Std-1	1						**2a**	**2b**				
	2								**1a**			
	3	**5b**										
	4			**5a**								
	5											**5c**
Sp.Std-2	1									**1b**		
	2										**1c**	
	3											
None			**3a**			**3b**						**4,6**

The pair number-letter (e.g., 5b) represents the items' association and will be explained in the next step. The way the standards and items are organized allows associations that use the same criterion to not be limited to a single pair of standards, i.e., the criterion can be applied to the whole set of standards from both domains, addressing the concern-1.

Note: In this step, the term *requirement* refers to the requirements that comprise the ECSS standards. In order to not confuse with product requirement, from now on *ECSS-requirement* will be used on those cases.

STEP-3: Perform and record the comparison (step related to concern-2)

This step applied the criteria from Table-B.1 to perform the comparison, but always taking as reference the impact on the activities of the *Aviation Oversight* (see assumptions of STEP-1). For every criterion, the items' association was identified in the spreadsheet, and for every association the comparison was recorded in a separate list of comparison description. In the example of Figure-B.2, numbers 1 to 6 in bold represent the criteria used for comparison and item's association. The criterion 1, for example, associates items 1, 2 and 3 (Av.Std-3), to the item 2 (Sp.Std-1), and items 1 and 2 (Sp.Std-2), respectively. If no association exists, the item is connected to the "none" from the other domain, and represents a domain specific characteristic (e.g., criterion 3, item 2, Av.Std-1, and item 1, Av.Std-2). Pairing with the bold numbers, the lowercase letters identify the index in the list of related comparison description illustrated in Figure-B.3.

Figure-B.3: Simplified example of the list of comparison description

Criteria	Index	Comparison description	Comparison Summary
1	a	description ...	summarized comparison for criteria 1
	b	description ...	
	c	description ...	
2	a	description ...	… for criteria 2
	b	description ...	
3	a	There are no equivalent items in Space standards	… for criteria 3
	b	There are no equivalent items in Space standards	
4		Criterion is applicable, but does not relate to any items from standards: description …	… for criteria 4
5	a	description ...	… for criteria 5
	b	description ...	
	c	There are no equivalent items in Aviation standards	
6		Criterion is not applicable to Aviation and Space	… for criteria 6

For every association from the spreadsheet there is one row of comparison description in the list. Cases of criteria 4 and 6 are related to the “none” to “none” association (see Figure-B.2). Examples of criterion 4 belong to the group *General Characteristic* which do not apply to any specific item, but to the whole set of standards. Cases like the criterion 6, though not expected, would just demand removing it from the Table-B.1. Once the list is completed, i.e., all spreadsheet’s associations were captured in the list, the comparison is summarized in the right column for each criterion.

STEP-4: Perform coverage analysis (step related to concern-3)

Using the spreadsheet results, a domain coverage analysis was performed in order to ensure software safety coverage of the chosen scope from both domains. In the example of Figure-B.2, the results of the coverage analysis are the gaps highlighted in grey, which identify the items without any related criteria (i.e. no association). Possible causes and related demanded actions are:

a. Item is not related to software safety; **action**: remove it from the spreadsheet.

b. There is association for at least one criterion, but the associated information is not in the expected format (e.g., the ECSS-requirement associates with DO-178C information other than objectives/activities); **action**: record the associated information in the spreadsheet with adequate identification, perform and record the comparison.

c. Missing criterion (Table-B.1 is incomplete); **action**: create criterion, perform and record the comparison.

STEP-5: Classify the systematic comparison results (step related to concern-4)

The starting point for constructing the Space oversight activities is the *Aviation Oversight* (see Section 3.2.4). The *Systematic Comparison Process* provides subsidies for identification of possible reuse of aviation best practices, as well as adjustments due to space oversight necessities. The following classification was adopted for the comparison results:

- **Type-A1, Aviation-only not reusable:** items that, though covered by the *Aviation Oversight*, do not have correspondence in space; for those cases, the aviation best practices are not reusable because are not applicable to the space oversight;

- **Type-A2, Aviation-only outside the *Aviation Oversight***: items that only exist in Aviation but are not covered by the *Aviation Oversight*; there are no aviation best practices to consider for reuse;

- **Type-AS1, partially reusable:** items covered by the *Aviation Oversight* but without clear correspondent items in space; they depend on adjustments to allow for reuse of aviation best practices;

- **Type-AS2, fully reusable:** items covered by the *Aviation Oversight,* and with correspondent items in space which should be covered by the space oversight; therefore, can allow for reuse of aviation best practices without adjustments;

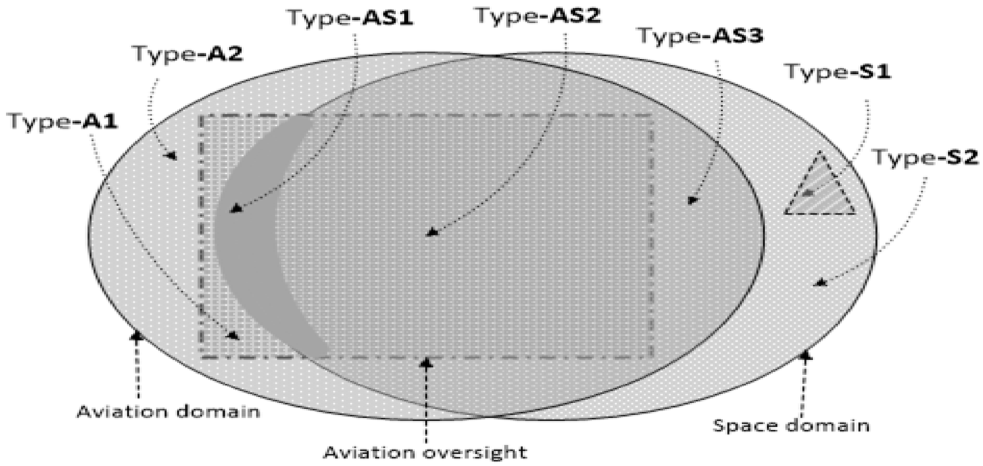
- **Type-AS3, similar but outside the *Aviation Oversight***: items that, though have correspondent items in space, are not covered by the *Aviation Oversight*; therefore, unlikely to be covered by the space oversight;

- **Type-S1, Space-only but in the intent of the *Aviation Oversight*:** Items that only exist in space, but they should be covered by the space oversight with punctual adjustments, preserving the basic intent of the *Aviation Oversight.*

- **Type-S2, Space-only beyond the intent of the *Aviation Oversight*:** Items that only exist in space, but a cost-benefit analysis should be performed to decide whether to extend the scope of oversight activities to cover them.

The figure-B.4 illustrates in a Venn diagram the adopted classification. The ellipses represent the set of software safety items from aviation and space domains, while the rectangle encompass the items considered relevant to the *Aviation Oversight* (see Section 3.2.4).

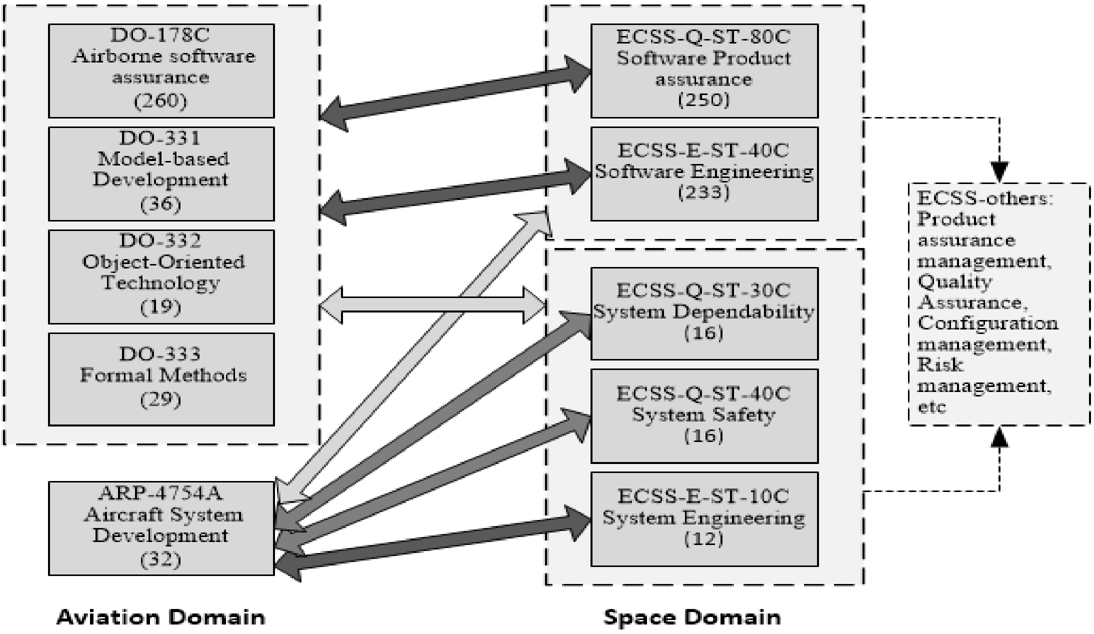
Figure-B.4: Venn diagram of the comparison results classification



B.2 - Standards selected for the comparison

Regarding to standards selected from the aviation domain, DO-178C was fully included and additional guidance from the supplements was also included (i.e., RTCA/DO-331, RTCA/DO-332 and RTCA/DO-333). For ARP4754A it was included the guidance that interfaces with the software. The other standards were also evaluated due to concerns on interface with software, but none were selected. The ARP4761 interfaces only with the ARP4754A (see Chapter-3, Figure-3.5) and the impact on software is indirect by classifying the function criticality which defines the DAL. In case of the DO-297, the IMA architecture that interfaces with software is already covered by the ARP4754A. As for the DO-254, per the Advisory Circular FAA-AC-20-152 (2005) the applicability is restricted to complex custom micro-coded components (e.g., application specific integrated circuits - ASIC, programmable logic devices – PLD, and field programmable gate array - FPGA), and in such case the interface with software is also covered by the ARP4754A. For the space domain, five standards were selected that together cover the scope of the civil aviation standards. The ECSS-E-ST-40C and ECSS-Q-ST-80C are dedicated to software, and comparison was made with all sections that specify ECSS-requirements. The ECSS-Q-ST-40C, ECSS-Q-ST-30C and ECSS-E-ST-10C are respectively for safety, dependability and system engineering (not specific for software), and it was included only the ECSS-requirements that interface with the software. Figure-B.5 depicts the standards and parenthesized number of items selected for comparison, indicating the level of equivalence between the compared standards.

Figure-B.5: Level of equivalence between standards selected for comparison

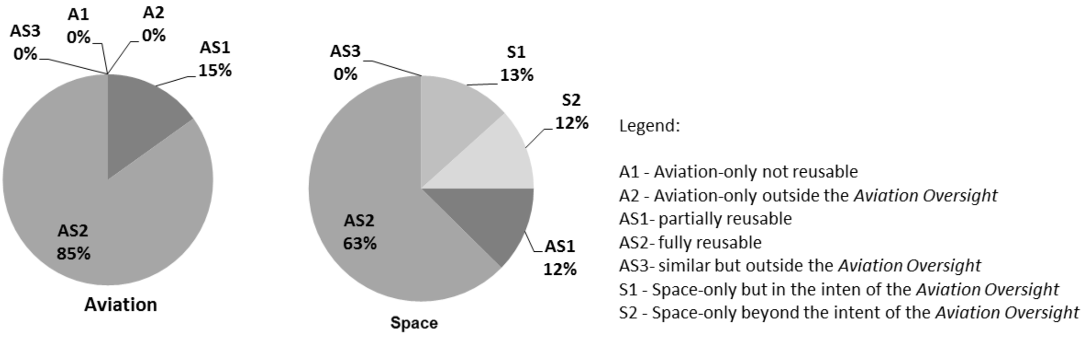


The arrows' grey tones represent the level of equivalence between a pair of standards (Aviation-Space), i.e., the pairs with higher percentage of associations has darker arrow connecting them (see appendix-C). The DO-178C and supplements are related to ECSS software engineering and software product assurance, and have some interface with system engineering, dependability and safety. Particularly, the MBD supplement can influence the system dependability, if used at system development scope. The ARP4754A is related to system engineering, but has some guidance on system safety and dependability, and only interfaces with the ECSS software engineering and assurance standards. Some ECSS items (e.g., ECSS-requirements for management, quality assurance, configuration) are more general, i.e., are applicable to many areas (e.g., electronic, software, mechanic) and have separated standards. For those cases, the ECSS standards selected for space domain provide only the specific items (e.g., software configuration ECSS-requirements), and makes reference to the standards that address the general items, as illustrated with the dashed arrows in Figure-B.5. For convenience of clarity, although some of those general items are applicable, they were not explicitly mapped for the comparison.

B.3 - Summary of the result in percentage

The figure-B.6 presents charts of aviation and space comparison results in percentage, according to the classification described in STEP-5. Both charts were obtained taking into account the scenario of using *Aviation Oversight,* described in section 3.2.6, as starting point for constructing space oversight activities. The aviation chart reflects the potential amount of reuse of, and exclusions from the *Aviation Oversight*. The space chart reflects the percentage of ECSS items that are already in the scope of the *Aviation Oversight,* or need to be added for constructing space oversight activities.

Figure-B.6: Comparison result charts



In the aviation chart the majority items can be fully reused (AS2=85%). The remaining (AS1=15%) can also be reused, but may demand some adjustments on the space oversight activities. It was not found non-reusable cases (A1=0%). The A2=0% and AS3=0% are due to the certification-oriented approach of aviation standards, i.e., only guidance considered relevant to certification are included; hence, there are no items irrelevant to the *Aviation Oversight*, as the oversight comprises only activities for certification purpose, as described in Section 3.2.6.

In the space domain chart, the majority are equivalent to the reusable items from aviation (AS1+AS2=75%), but a significant amount need to be added due to specific space necessities (S1=13%) or may be added depending on cost-benefit analysis (S2=12%). The table-B.2 provides the number of ECSS requirements per standards according to the comparison result classification, where “System interface” comprises the ECSS requirements that interface with software and belong to System Dependability, System Safety and System Engineering standards.

Table-B.2: ECSS requirements distribution per comparison result classification

Classification▶ ECSS Standards▼	**AS1**	**AS2**	**S1**	**S2**	**Total**
Software Engineering	10	74	30	16	130
Software Product Assurance	34	160	24	31	249
System interface	6	19	0	0	25
Total▶	50	253	54	47	404

The table-B.3 provides the same distribution, but in percentage.

Table-B.3: ECSS requirements distribution in percentage

Classification▶ ECSS Standards▼	AS1%	AS2%	S1%	S2%	Total%
Software Engineering	8	57	23	12	100
Software Product Assurance	14	64	10	12	100
System interface	24	76	0	0	100
Total%▶	12	63	13	12	100

B.4 - Summary based on the result classification

Type-AS1 - The partially reusable items are mainly due to some specific concepts in aviation. For those cases, some adjustments on the oversight activities may be necessary to cover the correspondent space items. Cases are as follow:

- The concept of Low-Level Requirement (LLR), from where the source code is directly produced. For space, the code is produced from the software units which are defined at detailed design phase. Nevertheless, the functionalities implemented by each unit can be seen as LLRs, although not explicitly named as such.
- The tests are all based on requirements (i.e., no white box testing). Differently, for space the software units can be tested based on the code structure.
- The concept of derived requirements, which are those that are not directly traceable to higher level requirements and or specify behavior beyond that specified by the system requirements or the higher-level software requirements; their existence must be justified and evaluated at system level for any adverse impact. For space, although that term does not exist, the concerns and related activities are also applicable to any new requirement or design decision made at software level.
- The concept of architecture as related to LLR. For the space, the architecture is related to software technical requirements (i.e., similar with HLR), and hierarchically below comes the detailed design (i.e., equivalent to architecture in Aviation domain) which is related to the software units.
- The traceability between HLRs and LLRs, where the architecture should be compatible with (but not traced to) the HLRs. For space, the traceability is between

the elements of the architecture (i.e., components) and elements of the detailed design (i.e., units).

- The criteria for code coverage. For space, it is not required 100% statement coverage for level C, and for some other cases the percentage can be agreed with the customer.
- The objectives and activities of the certification liaison process. For space, the customer-supplier relationship needs similar activities and can partially reuse from aviation.
- The planning process, mandatory at the beginning to plan all activities to be performed throughout the development. For space, it is not mandatory to plan all activities at the beginning. For example, development plan is required for SRR, but verification plan is required for PDR and maintenance plan for QR.

Type-AS2: The *Aviation Oversight* activities can be reused by the space oversight to assess through samplings the quality of, and adherence to the process of development and respective verification, covering from the space system requirements allocated to software until the executable code, including the requirement-based testing in the representative environment. The quality, configuration control and traceability of the generated life-cycle data, the nonconformity records and actions for solution, and the QA records, among others, are used as evidences. These oversight activities check for desirable properties that are common to most software engineering approaches; therefore, can be easily adapted to space domain. The additional objectives/activities of supplements DO-331, DO-332 and DO-333 can be useful for providing technology-specific guidance (e.g., MBD, OOT, formal methods).

Remark: It was not found any case where reuse would not be possible.

Type-AS3: no cases have been found;

Type-S1 - Although a high percentage of reuse was identified, a considerable amount of space specific necessities exists. The following cases demand additions in the space oversight activities, but preserving the intent of the *Aviation Oversight*:

- Space standards can be tailored based on technical, operational, managerial, conditional requirements, and customer-supplier agreement, which affect the mandatory set of ECSS-requirements, and should be captured by the space oversight process prior to starting the audit assessment.

- For space, the customer specifies the RB and provides them to the supplier. However, ECSS allows the supplier to specify the RB under support of the customer. Therefore, those activities that are typical of system scope are also addressed in the software scope.
- Due to the customer-supplier approach, space has the delivery and acceptance process, which delimitates the end border between supplier and customer.
- Due to some spacecraft operational characteristics, space software requires the possibility of maintenance inflight, high integrity communication with Ground, protection against single-event upset (SEU), and concerns on disposal phase.
- Space allows the customer to require an independent organization to perform V&V.
- Space requires the use of model to provide behavioral view in order to support the verification of requirements, architecture and detailed design.
- Space requires the use of computational models for the dynamic architecture design.
- Space requires mission and configuration dependent data to segregate from the software, e.g., a separate database.
- Space requires the specification of software quality requirements.

Type-S2 - Some additions go beyond the intent of the *Aviation Oversight*, and should be evaluated for cost-benefit to decide whether to extend the scope of the space oversight. Cases are as follow:

- Processes of procurement and retirement;
- Organization-related guidance including qualification and training program;
- Process assessment for capability and maturity level;
- Process and product metrics
- Ground software development assurance.
- A process for operation phase prior to launching.
- Space provides for a separate process for maintenance.

B.5 - Summary of the result based on comparison criteria

Taking into account the list of the comparison criteria (see Table-B.1), a summary of the differences is provided below by the criteria classification:

General characteristic: ECSS standards are well organized and harmonized, with terms-of-reference, top-level document, general glossary of terms, continuous revisions for improvements and updates, as result of cooperation among European space agencies and industries. In aviation, some standards are under responsibility of distinct and independent organizations (e.g., RTCA for software and SAE for systems), with distinct working procedures and non-synchronized schedules. Although there are concerns for harmonization, some differences exist in interface specification and terminologies. Aviation standards are driven by certification, and the standards are very clear about mandatory objectives to be accomplished. Space standards are driven by the customer-supplier relationship, allowing the customer-supplier contract to specify the mandatory set of requirements, i.e., solely by the standards it is not possible to identify which requirements are mandatory. ECSS addresses process assurance and product assurance separately and in distinct documents. Aviation addresses both together, but without clear separation.

Safety-related: Space requires safety analysis at software level with identification of hazard conditions caused by software, definition of hazard control performed by the software and related verification methods, followed by verification evidences of hazard control implemented by software. It recommends the use of Software Failure Mode and Effective Analysis - SFMEA, Software Fault Tree Analysis - SFTA, and Software Common Cause Analysis - SCCA. Aviation does not perform safety analysis at software level, and the software safety evidences focus on compliance with the assigned development assurance level. The actual hazard identification, control and related verification are performed at system level.

Process-related: ECSS lifecycle has additional processes (i.e., procurement, acceptance, operation and retirement), which are not addressed by aviation. Aviation has the concept of LLR from which the source code is directly produced, while for space the code is produced from the software units, which are defined at detailed design.

Product-related: After entry into service, the same aviation software product may change throughout the aircraft lifecycle for corrections, improvements or addition of new functionalities. In some cases, the architecture provides features that allow the user to modify the software, bringing more operational flexibility without compromising safety. Due to some spacecraft operational characteristics, space software requires the possibility of maintenance inflight.

Organization-related: ECSS main stakeholders comprise the customer, supplier, maintainer and operator. For the aviation, though not explicit, the main stakeholders are the certifier, the applicant for certification (usually the airplane integrator/manufacturer), the system supplier, and the software supplier. The ECSS customer-supplier approach requires organization-related guidance including definition of roles, responsibilities, hierarchy, qualification and training program, and procurement. For the aviation software, the DO-178C states that "*Matters concerning the structure of the applicant's organization, the commercial relationships between the applicant and its suppliers, and personnel qualification criteria are beyond the scope of this document*". The only organization-related guidance is for ensuring the independence and authority of the SQA.

Methods and techniques: ECSS requires computational models for behavioral analysis of real-time software. Aviation does not, but it provides a supplement with guidance for MBD, including the use of model simulation. Aviation recognizes only requirements-based tests for certification credits. ECSS provides guidance on white box testing as part of software development process (coding and unit test phase), where code structures are exercised.

Integrity concerns: Space demands high integrity (especially with Ground communication), use of fault tolerance techniques, degraded modes, protection against single-event upset (SEU), and explicitly requires dependability analysis at both, system and software level. Aviation emphasizes safety and may apply redundancy, safety monitoring, diversity, dissimilarity to prevent a single failure to leading to a catastrophic event. Aviation does not require dependability analysis, but considers the other dependability aspects (i.e., reliability, availability, maintenance) if they adversely impact safety.

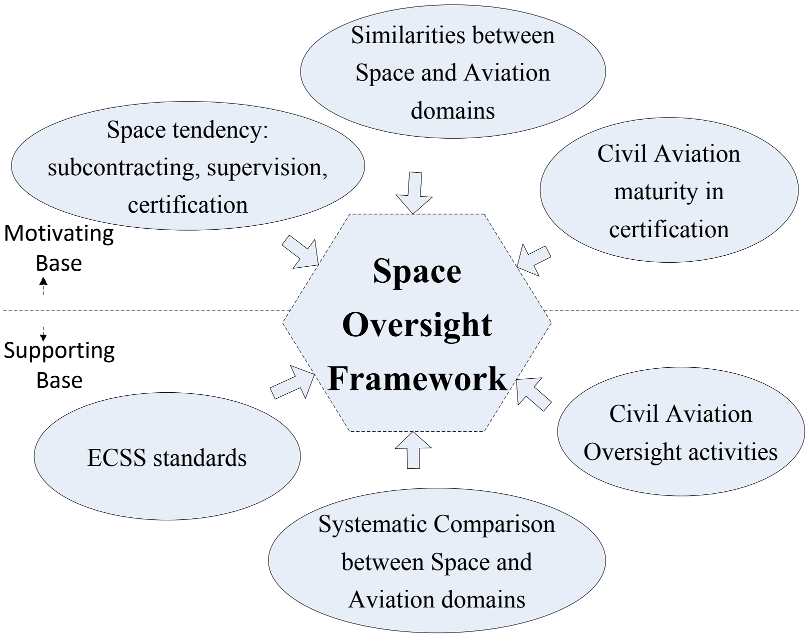
Additional concerns: ECSS guidance for determination of tool qualification level is based on the aviation guidance. However, for the related qualification activities the ECSS refers to the automotive guidance. Aviation has a dedicated document for tool qualification.

APPENDIX-C: AN OVERVIEW OF THE SPACE OVERSIGHT FRAMEWORK

C.1 – General context and scope

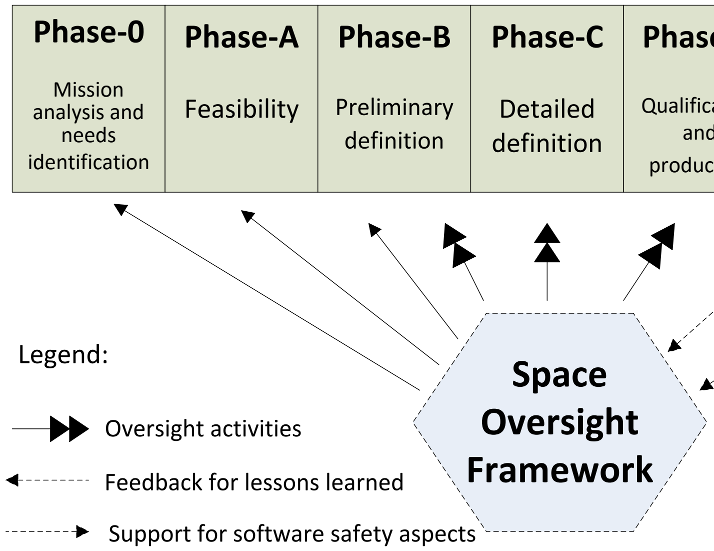
The motivation base to construct the Space Oversight Framework is the "*space tendency*", "*civil aviation maturity in certification*" and "*similarity between aviation and space*." The space domain tendency is for more oversight, either due to the outsource growth of increasingly complex parts or the need for regulation and consequent certification activity. In this scenario, the civil aviation high-level of maturity in certification comes as a potential source of contribution, because there are many similarities between these two domains, particularly regarding to software-intensive critical embedded systems (see section-6.2). The figure-C.1 shows the general context of the Space Oversight Framework.

Figure-C.1: General context of the Space Oversight Framework


Similarities between Space and Aviation domains
Space tendency: subcontracting, supervision, certification
Civil Aviation maturity in certification
Motivating Base
Space Oversight Framework
Supporting Base
ECSS standards
Civil Aviation Oversight activities
Systematic Comparison between Space and Aviation domains

The supporting base for the Space Oversight Framework construction comprises "*ECSS standards*", "*Civil Aviation Oversight activities*" and "*Systematic Comparison.*" The Systematic Comparison identifies similarities and differences between space and civil aviation in order to apply the civil aviation best practices customized for the space domain to build the Space Oversight Framework. The figure-C.2 shows the Space Oversight Framework scope in different phases of the space mission development. The Space Oversight Framework covers mainly the phases B, C and D.

Figure-C.2: The Space Oversight Framework general scope



The emphasis of the oversight application is when the software supplier is defined (phase B), the software is developed (phase C), verified and delivered (phase D). However, the Space Oversight Framework can also work in the earlier stages (stage 0 and A) providing support regarding software safety concerns, as well as in later phases (E and F) evaluating feedbacks from operational and disposal difficulties, and their impacts in the Space Oversight Framework as part of the lessons learned process.

C.2 – Main activities

The oversight activities begin with a risk assessment in the software supplier. The result of the risk assessment will define which subsequent oversight activities are necessary, starting from desktop review of key documents, e.g., the development plan and software delivery document (the lowest critical), up to a permanent staff on supplier's site (highest critical case), and may perform up to five formal reviews (intermediate cases) as follow:

Stage#1, Requirements Baseline and Planning: usually desktop review of the RB requirements defined by the customer at system level, and the initial supplier´s planning documents like development plan, verification plan, configuration management plan, quality assurance plan, and any standard documents to be adopted (e.g., requirements standard, coding standard), in order to ensure compliance to the software criticality level.
Remark: This stage set the transition from customer to supplier.

Stage#2, Software Requirements and architecture: usually on-site review of the processes implemented (tools, procedures, etc.) as well as the quality of the TS requirements, preliminary architecture and related life cycle data, in order to ensure compliance to the planning documents and adopted standards.

Stage#3, Detailed Design and implementation: usually on-site review of the processes implemented as well as the quality of the detailed design, source and object code, and related life cycle data, in order to ensure compliance to the planning documents and adopted standards.

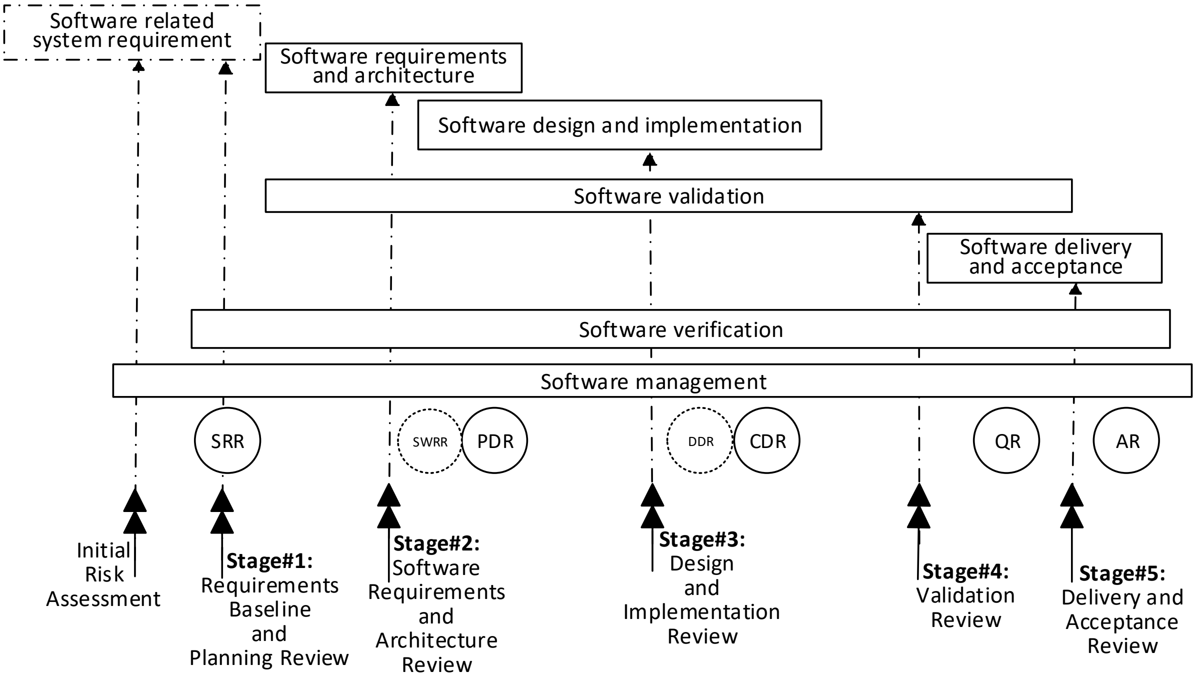
Stage#4, Validation: usually an on-site review of the processes implemented as well as the quality of the validation activities (e.g., testing against RB requirements and TS requirements) and related life cycle data, in order to ensure compliance to the planning documents and adopted standards.

Stage#5, Delivery and Acceptance: usually an on-site review of the processes implemented as well as the quality of the acceptance activities and related life cycle data, in order to ensure compliance to the planning documents and adopted standards.
Remark: This stage set the transition from supplier to customer.

Figure-C.3 shows at what periods of the software life cycle the initial risk assessment and formal reviews occur. The figure uses as reference the software life cycle process defined by the ECSS-E-ST-40C. The initial risk assessment should occur:

- after the software supplier selection and during the early stage of the "software management process";
- during the final stages of the software product definition (the second half of " software related system requirement process"), already with the supplier participation in the software definition finalization;
- After starting the planning of development and V&V.

Figure-C.3: Space Oversight Framework activities in the ECSS software life cycle


Software related system requirement
Software requirements and architecture
Software design and implementation
Software validation
Software delivery and acceptance
Software verification
Software management
SRR
SWRR
PDR
DDR
CDR
QR
AR
Initial Risk Assessment
Stage#1: Requirements Baseline and Planning Review
Stage#2: Software Requirements and Architecture Review
Stage#3: Design and Implementation Review
Stage#4: Validation Review
Stage#5: Delivery and Acceptance Review

Source*:* Adapted from ECSS-E-ST-40C (2009)

The Stage#1, Requirements baseline and planning review, should occur:

- Simultaneously with the System Requirements Review (SRR) or be part of it as complement;
- After finishing the planning of development and V&V;
- Before starting development and V&V activities.

The Stage#2, software requirement and architecture review, should occur:

- After more than 50% of the requirements and architecture have been defined, verified and validated;
- Before starting the software design and implementation;
- Before starting the Software Requirements Review (SWRR), as the focus is on the assessment of the processes and of a representative sample of requirements and architecture. It is an important mitigation for the Preliminary Design Review (PDR).

The Stage#3, design and implementation review, should occur:

- After more than 50% of the design and implementation have been completed, verified and validated;

- Before starting the Detailed Design Review (DDR), as the focus is on the assessment of the processes and of a representative sample of design and implementation.

The Stage#4, validation review, should occur:

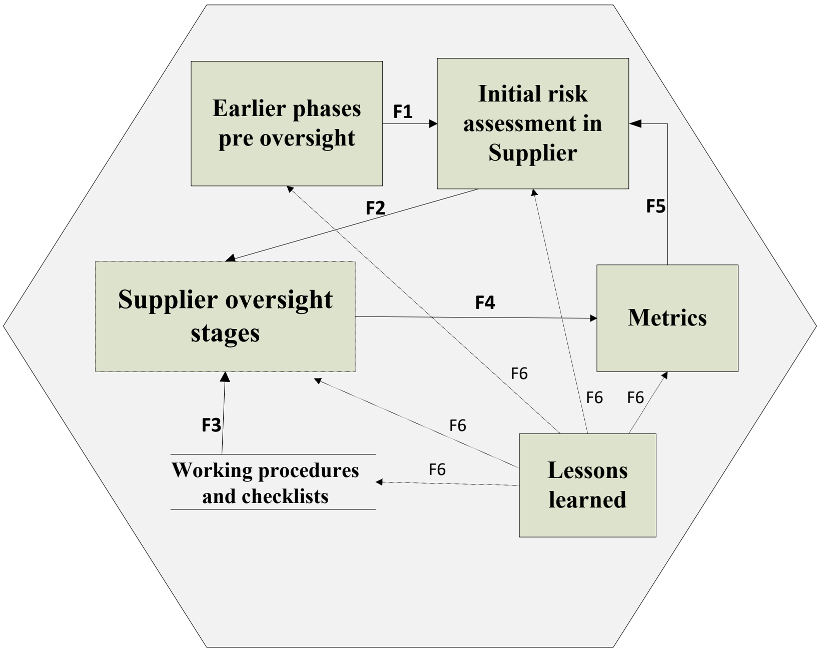
- After more than 50% of the requirements have been V&V through testing;
- Before starting the delivery/acceptance process.

The Stage#5, delivery and acceptance review can occur simultaneously with the Acceptance Review (AR) or be part of it as a complement.

C.3 – Main components

The Space Oversight Framework comprises the following components illustrated in the figure-C.4:

Figure-C.4: The Space Oversight Framework main components


Earlier phases pre oversight
F1
Initial risk assessment in Supplier
F2
F5
Supplier oversight stages
F4
Metrics
F6
F6
F6
F3
F6
Working procedures and checklists
F6
Lessons learned

The components are summarized below:

- **Earlier phases pre-oversight:** Support for software safety aspects in phase-0 and phase-A.

- **Initial risk assessment in supplier**: evaluation of company experience, use of subcontractors, level of reuse, new technologies, required safety level, system complexity, etc., to make the tailoring of the applicable ECSS standards and requirements and determine the oversight activities.
- **Supplier oversight stages**: planned activities commensurate to the risk assessment outcome (e.g., continuous supervision, periodic on-site reviews, periodic desktop reviews).
- **Working procedures and checklists:** for supporting the planned activities.
- **Metrics** for oversight evaluation and decision-making support (the focus of the thesis).
- **Lessons learned** from phase-E and phase-F to improve the Oversight Framework.

The flows among the components are summarized below:

F1: specific software safety concerns detected during phase-0 and phase-A;

F2: oversight activities to be performed as result of the risk assessment;

F3: set of working procedures and checklists to be used during oversight activities;

F4: oversight results for generation of measurement by applying the metric;

F5: measurement evaluation for continuous risk assessment;

F6: process improvement.

C.4 – Working procedures

The working procedures describe the steps for performing the on-site audits, and cover the document evaluation, assessment of the quality of the implemented process and adherence to the process. An overview is provided below:

Space Oversight Working Procedures

```
BEGIN
    IF document evaluation THEN refer to checklist for document evaluation
    ELSE
        Refer to the aviation working procedures that are fully applicable to verify
        compliance with ECSS requirements classified as AS2;
```

```
            Refer to the aviation working procedures that are fully applicable, but should check
            for specificities of the ECSS requirements classified as AS1;

            Refer to the specific space working procedure that should be used to verify
            compliance with ECSS requirements classified as S1;

            Refer to the specific space working procedure that should be used to verify
            compliance with ECSS requirements classified as S2; /* see consideration ‘a’ */

        END   /* ELSE */

    END /* Working Procedures */
```

Some considerations:

a. ECSS requirements classified as S2 are, a priori, excluded from the scope of the Space Framework described herein. However, if after a cost-benefit analysis some S2 requirements are included, then an impact analysis should be performed in order to identify adjustments in the existing working procedures and necessity of additional ones;

b. The Stage#1 covers activities performed mainly by the customer, and consequently the Stage#5 also includes customer activities for acceptance;

c. There is a specific oversight procedure for assessment of the validation due to Ground environment necessities;

d. If IVV is applicable, there is a separate oversight procedure focusing on a specific stakeholder responsible for the IVV, other than customer and supplier;

e. There is an additional process specific for maintenance after the acceptance process. However, for this thesis it was decided to keep the maintenance under responsibility of the supplier, in order to maintain similarity with the civil aviation approach.

C.5 - Software Compliance Checklist

Similar to the Civil Aviation, which performs oversight activities to verify compliance to DO-178C objectives and activities, space oversight activities verify compliance to the applicable ECSS-requirements. For a summary of aviation applicable objectives and activities refer to the appendix-A, and for space applicable ECSS requirements refer to the appendix-B, section B.2, where the ECSS standards were incorporated in the Space Framework as a result of the *Systematic Comparison Process*. As part of the Space Oversight Framework, it was developed a spreadsheet to work as a checklist, called *Software Compliance Checklist,* for recording and controlling the compliance verification with the applicable ECSS-requirements. The spreadsheet provides the possibility of filtering the ECSS-requirements according to the following:

a. *Stage*: the audit stage according to the Space Oversight Framework, where the ECSS-requirement is applicable and should be assessed for compliance verification;

b. *Stakeholder*: the stakeholder to which the ECSS-requirement is applicable, i.e., the one that should ensure compliance to the ECSS-requirement (i.e., Cst=customer, Spp=supplier, Mnt=maintainer, Opr=operator, C&C=customer and supplier, C&M=customer and maintainer, C&M&O=customer and maintainer and operator);

c. *ECSS vs Aviation*: the systematic comparison result classification (i.e., AS1, AS2, S1, S2);

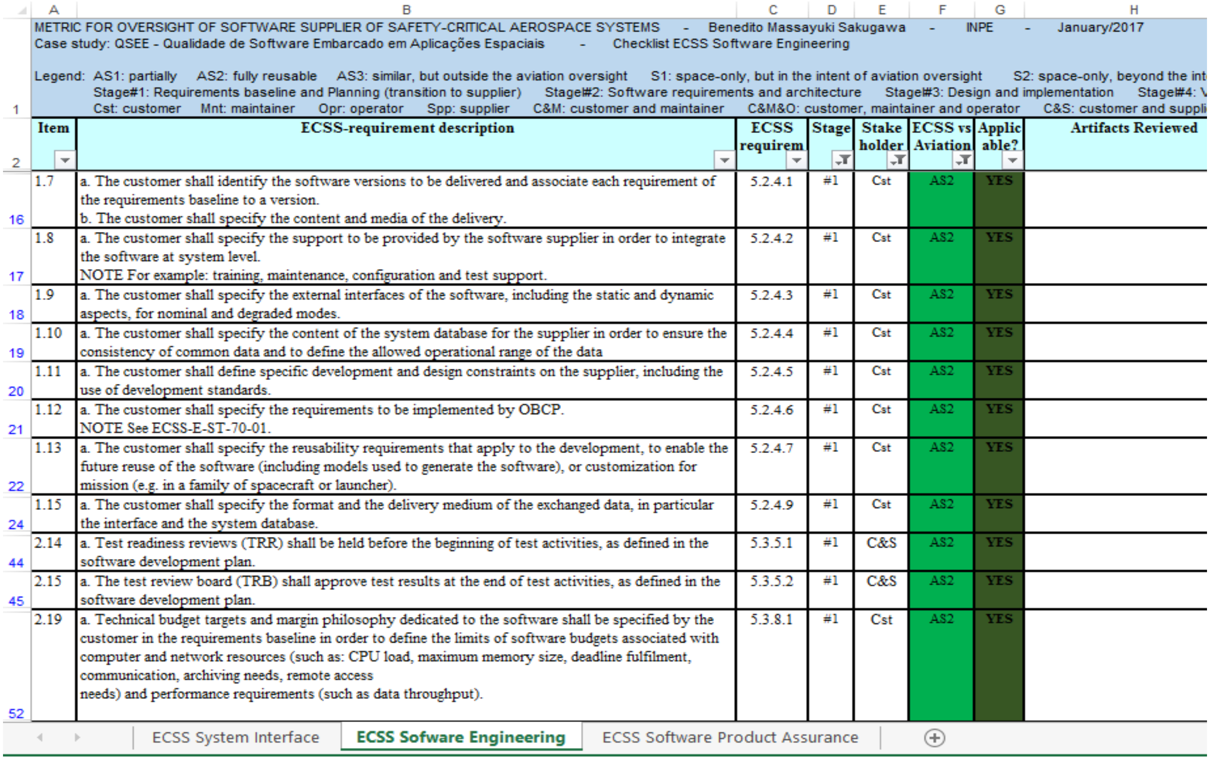
d. *Applicable?:* the ECSS-requirements applicable to the specific project. It is related to the type-S1, case ”a”, tailoring of the ECSS standards.

For example, during the Stage#1, assessment of the customer activities using the oversight procedure which is fully common between aviation and space, the filters should select the following:

- *Stage* = Stage#1
- *Stakeholder* = Cst (customer), C&S (customer, supplier), C&M (customer, maintainer) and C&M&O (customer, maintainer, operator)
- *ECSS vs Aviation* = AS2 (fully reused)
- *Applicable?* = YES

The figure-C.5 shows the compliance checklist after applying the above filter to the ECSS Software Engineering tab:

Figure-C.5: Example of applying filter in the *Software Compliance Checklist*



METRIC FOR OVERSIGHT OF SOFTWARE SUPPLIER OF SAFETY-CRITICAL AEROSPACE SYSTEMS - Benedito Massayuki Sakugawa - INPE - January/2017
Case study: QSEE - Qualidade de Software Embarcado em Aplicações Espaciais - Checklist ECSS Software Engineering

Legend: AS1: partially AS2: fully reusable AS3: similar, but outside the aviation oversight S1: space-only, but in the intent of aviation oversight S2: space-only, beyond the int
Stage#1: Requirements baseline and Planning (transition to supplier) Stage#2: Software requirements and architecture Stage#3: Design and implementation Stage#4: V
Cst: customer Mnt: maintainer Opr: operator Spp: supplier C&M: customer and maintainer C&M&O: customer, maintainer and operator C&S: customer and suppli

	A	B	C	D	E	F	G	H
2	Item	ECSS-requirement description	ECSS requirem	Stage	Stake holder	ECSS vs Aviation	Applic able?	Artifacts Reviewed
16	1.7	a. The customer shall identify the software versions to be delivered and associate each requirement of the requirements baseline to a version. b. The customer shall specify the content and media of the delivery.	5.2.4.1	#1	Cst	AS2	YES	
17	1.8	a. The customer shall specify the support to be provided by the software supplier in order to integrate the software at system level. NOTE For example: training, maintenance, configuration and test support.	5.2.4.2	#1	Cst	AS2	YES	
18	1.9	a. The customer shall specify the external interfaces of the software, including the static and dynamic aspects, for nominal and degraded modes.	5.2.4.3	#1	Cst	AS2	YES	
19	1.10	a. The customer shall specify the content of the system database for the supplier in order to ensure the consistency of common data and to define the allowed operational range of the data	5.2.4.4	#1	Cst	AS2	YES	
20	1.11	a. The customer shall define specific development and design constraints on the supplier, including the use of development standards.	5.2.4.5	#1	Cst	AS2	YES	
21	1.12	a. The customer shall specify the requirements to be implemented by OBCP. NOTE See ECSS-E-ST-70-01.	5.2.4.6	#1	Cst	AS2	YES	
22	1.13	a. The customer shall specify the reusability requirements that apply to the development, to enable the future reuse of the software (including models used to generate the software), or customization for mission (e.g. in a family of spacecraft or launcher).	5.2.4.7	#1	Cst	AS2	YES	
24	1.15	a. The customer shall specify the format and the delivery medium of the exchanged data, in particular the interface and the system database.	5.2.4.9	#1	Cst	AS2	YES	
44	2.14	a. Test readiness reviews (TRR) shall be held before the beginning of test activities, as defined in the software development plan.	5.3.5.1	#1	C&S	AS2	YES	
45	2.15	a. The test review board (TRB) shall approve test results at the end of test activities, as defined in the software development plan.	5.3.5.2	#1	C&S	AS2	YES	
52	2.19	a. Technical budget targets and margin philosophy dedicated to the software shall be specified by the customer in the requirements baseline in order to define the limits of software budgets associated with computer and network resources (such as: CPU load, maximum memory size, deadline fulfilment, communication, archiving needs, remote access needs) and performance requirements (such as data throughput).	5.3.8.1	#1	Cst	AS2	YES	

The table-C.1 provides the distribution of the ECSS Software Engineering requirements from the *Software Compliance Checklist* per audit stages and stakeholders, where the stages adopted by the Space Oversight Framework are as follow:

- Stage#1: Requirements Baseline and Planning (transition from customer to supplier)
- Stage#2: Software requirements and architecture
- Stage#3: Software design and implementation
- Stage#4: Software validation
- Stage#5: Software delivery and acceptance (transition from supplier to customer)

Table-C.1: Distribution of ECSS Software Engineering requirements

Stakeholder ▶ Stage ▼	Cst	Spp	Mnt	Opr	Total
#1	32	20	1	-	53
#2	-	16	-	-	16
#3	-	21	-	-	21
#4	1	12	-	-	13
#5	6	7	-	-	13
ALL	2	6	14	-	22
OUT	4	0	6	10	20
Total	45	82	21	10	158

In order to facilitate the analysis, those requirements that impacts more than one stakeholder or stages were counted in all impacted ones. Hence, the columns of stakeholders (and rows of stages) did not include combinations and the total number of ECSS-requirements accounted for more than the actual value (i.e., 158 instead of 130). Both Stage#1 and Stage#5 are shared between the customer and supplier, whereas Stage#2, Stage#3 and Stage#4 are in the supplier scope. The ECSS-requirements that are applicable to all stages (i.e., ALL) are mainly for configuration management, quality assurance and maintenance, and those that are out of the framework scope (i.e., OUT) are mainly cases related to the Operation process, and maintenance as separate group or organization.

C.6 – Closure comments

There are some considerations due to differences between aviation and space. In the aviation, suppliers are generally manufacturers of the equipment or system, and hold the know-how including the software. Therefore, the integrator does not have complete access to the software design data (i.e., requirements, architecture, and code). The contract with the aircraft integrator may consider the amount of equipment provided by the supplier for the aircraft manufacturing. In the case of space, the software supplier may not have the know-how of the equipment or system, and does not close the contract based on the amount of equipment provided by the supplier to the satellite integrator, because usually spacecraft is not serial production. The contract is based on the delivery of the software product that often includes the entire design, and a support for transferring of maintenance to the spacecraft integrator, or even to a third organization. Therefore, the oversight may have a mechanism somewhat different from that of the aviation, and is determined by contract.

APPENDIX D: AVIATION SURVEY PROCESS

D.1 – Introduction

The survey performed with aviation software safety specialists had the following objectives:

a. To obtain quantitative values for those metrics that, a priori, are not quantifiable;
b. To obtain quantitative relevance of each metric;
c. To identify new metrics;
d. To identify any dependency among the metrics;
e. To obtain scores for severity of a list of issues generated from ANAC past audits.

ANAC is among the major civil aviation certification agencies and attended the survey with 5 specialists. The aviation industry attended with 14 specialists and is among the world major industries for transport aircraft. The participants average experience with software safety is considerably high (16.8 years), and their participation in the survey can be considered representative of the international auditors' experience. The survey results can be divided in 4 types according to the 4 objectives previously mentioned, as follow:

a. Quantitative values for those metrics that, a priori, are not quantifiable;
b. Quantitative relevance of each metric;
c. New metrics identified;
d. Discussion on dependency among the metrics.

Remark: In this appendix, the term SOI has been used instead of "Stage", due to the familiarity of the survey participants.

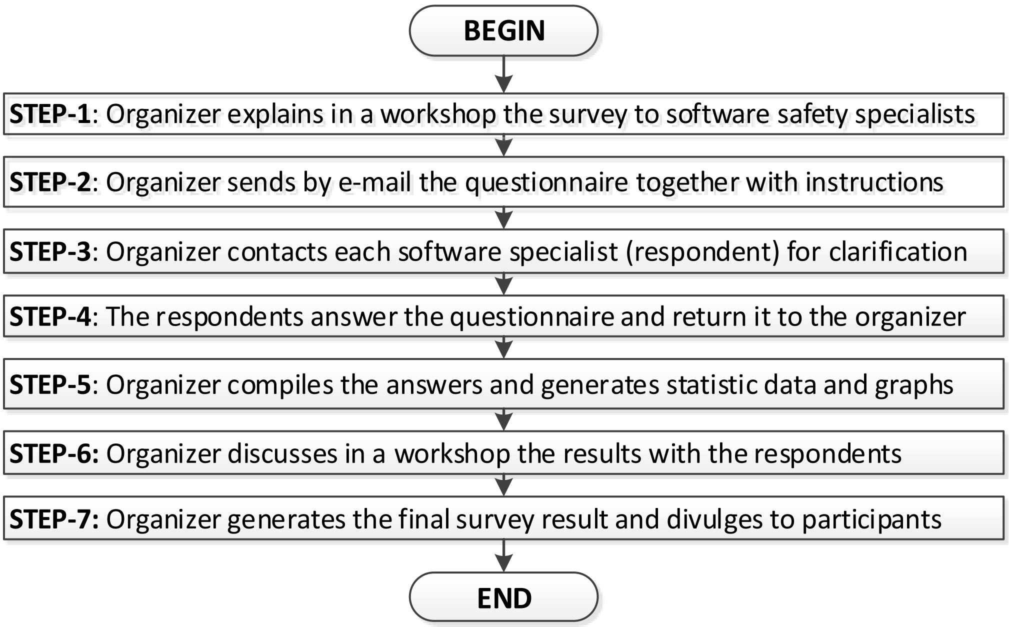
D.2 – The process description

The figure-D.1 illustrates the process used in the survey. The steps are described as follow:

STEP-1 - Organizer explains in a workshop the survey to software safety specialists:

A presentation of the topic "*Participation of the software specialists in the activities of analysis and improvement of ANAC internal processes*" took place during a workshop;

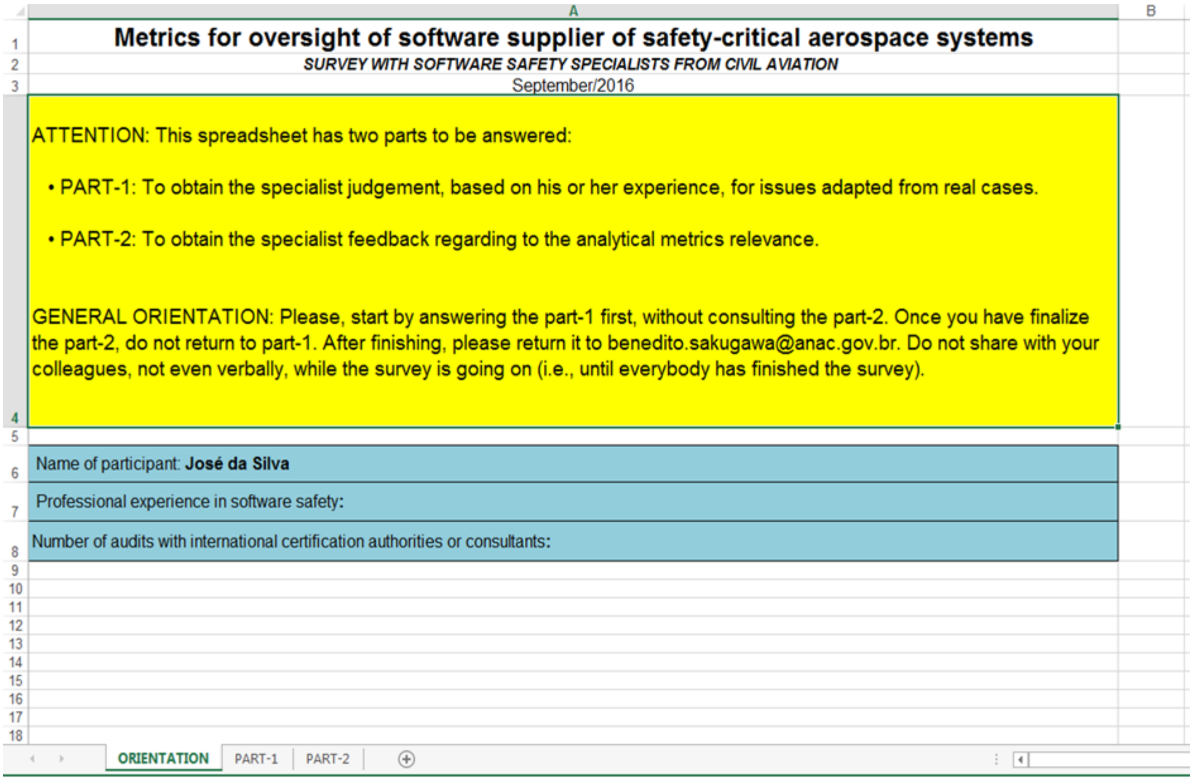
Figure-D.1: The process used for the survey


BEGIN
STEP-1: Organizer explains in a workshop the survey to software safety specialists
STEP-2: Organizer sends by e-mail the questionnaire together with instructions
STEP-3: Organizer contacts each software specialist (respondent) for clarification
STEP-4: The respondents answer the questionnaire and return it to the organizer
STEP-5: Organizer compiles the answers and generates statistic data and graphs
STEP-6: Organizer discusses in a workshop the results with the respondents
STEP-7: Organizer generates the final survey result and divulges to participants
END

STEP-2 - Organizer sends by e-mail the questionnaire together with instructions:

The figure-D.2 shows the spreadsheet sent to each participant by individual emails with guidelines as follow:

Figure-D.2: The spreadsheet provided to survey participants



Metrics for oversight of software supplier of safety-critical aerospace systems

SURVEY WITH SOFTWARE SAFETY SPECIALISTS FROM CIVIL AVIATION

September/2016

ATTENTION: This spreadsheet has two parts to be answered:

- PART-1: To obtain the specialist judgement, based on his or her experience, for issues adapted from real cases.
- PART-2: To obtain the specialist feedback regarding to the analytical metrics relevance.

GENERAL ORIENTATION: Please, start by answering the part-1 first, without consulting the part-2. Once you have finalize the part-2, do not return to part-1. After finishing, please return it to benedito.sakugawa@anac.gov.br. Do not share with your colleagues, not even verbally, while the survey is going on (i.e., until everybody has finished the survey).

Name of participant: **José da Silva**

Professional experience in software safety:

Number of audits with international certification authorities or consultants:

The Spreadsheet is composed of three tabs: ORIENTATION shown in the figure, PART-1 containing items based on actual cases of past audits, PART-2 containing the metrics that need quantitative values.

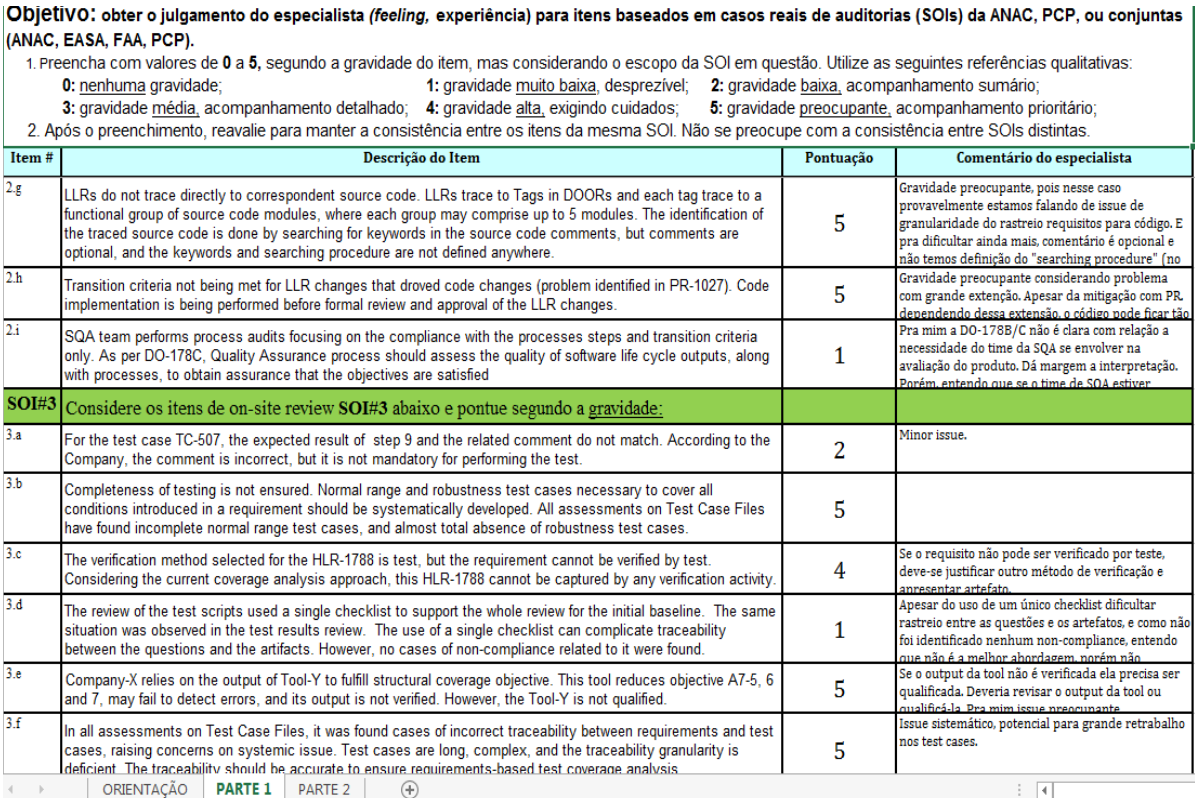
STEP-3 - Organizer contacts each software specialist (respondent) for clarification:

The contact was in person or by phone, or via e-mail when the previous means were not possible, to confirm understanding of the task to be performed and commitment with the schedule (i.e., one month for answering).

STEP-4 - The respondents answer the questionnaire and return it to the organizer:

The average time to fill-up the spreadsheet was 40 minutes. During this step, some questions were clarified and, when applicable, shared with all participants via e-mail. In some cases, additional guidance was required for clarification or reinterpretation of spreadsheet guidance texts. Figure-D.3 provides an example of a completed spreadsheet for PART-1 (instructions and participant´s comments were written in Portuguese).

Figure-D.3: Spreadsheet filled-up (PART-1) by a survey participant



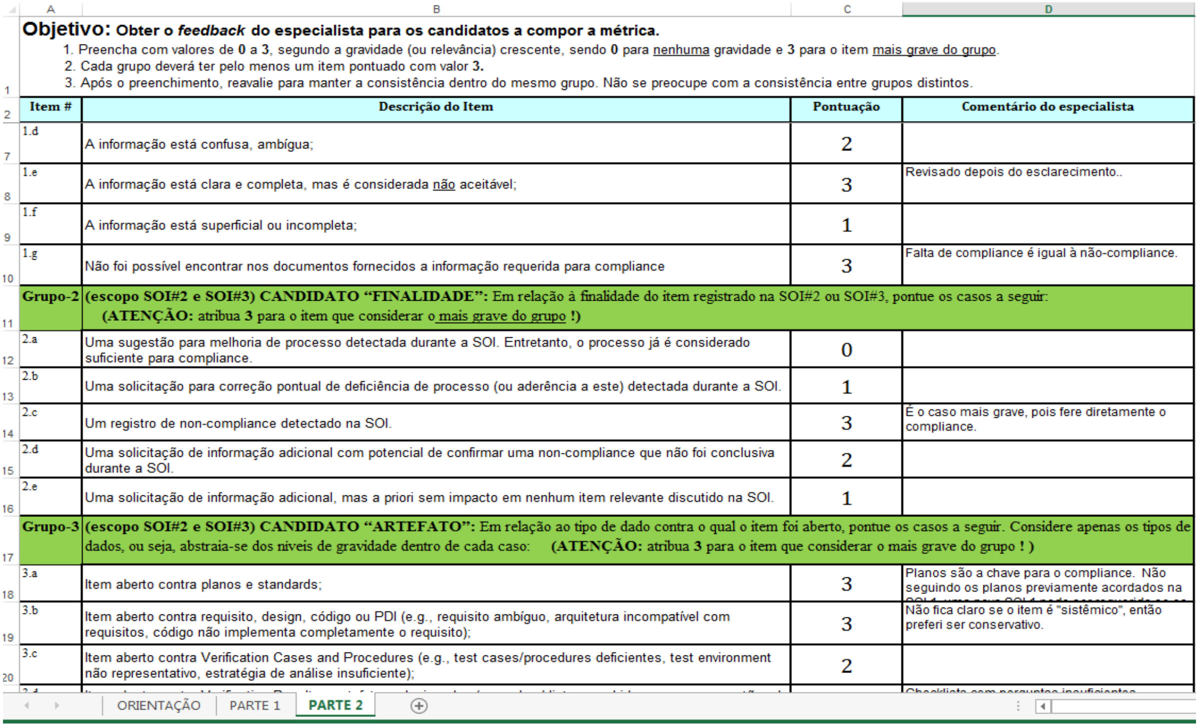
Objetivo: **obter o julgamento do especialista *(feeling,* experiência) para itens baseados em casos reais de auditorias (SOIs) da ANAC, PCP, ou conjuntas (ANAC, EASA, FAA, PCP).**

1. Preencha com valores de **0** a **5,** segundo a gravidade do item, mas considerando o escopo da SOI em questão. Utilize as seguintes referências qualitativas:
   0: nenhuma gravidade; **1:** gravidade muito baixa, desprezível; **2:** gravidade baixa, acompanhamento sumário;
   3: gravidade média, acompanhamento detalhado; **4:** gravidade alta, exigindo cuidados; **5:** gravidade preocupante, acompanhamento prioritário;
2. Após o preenchimento, reavalie para manter a consistência entre os itens da mesma SOI. Não se preocupe com a consistência entre SOIs distintas.

Item #	Descrição do Item	Pontuação	Comentário do especialista
2.g	LLRs do not trace directly to correspondent source code. LLRs trace to Tags in DOORs and each tag trace to a functional group of source code modules, where each group may comprise up to 5 modules. The identification of the traced source code is done by searching for keywords in the source code comments, but comments are optional, and the keywords and searching procedure are not defined anywhere.	5	Gravidade preocupante, pois nesse caso provavelmente estamos falando de issue de granularidade do rastreio requisitos para código. E pra dificultar ainda mais, comentário é opcional e não temos definição do "searching procedure" (no
2.h	Transition criteria not being met for LLR changes that droved code changes (problem identified in PR-1027). Code implementation is being performed before formal review and approval of the LLR changes.	5	Gravidade preocupante considerando problema com grande extenção. Apesar da mitigação com PR, dependendo dessa extensão, o código pode ficar tão
2.i	SQA team performs process audits focusing on the compliance with the processes steps and transition criteria only. As per DO-178C, Quality Assurance process should assess the quality of software life cycle outputs, along with processes, to obtain assurance that the objectives are satisfied	1	Pra mim a DO-178B/C não é clara com relação a necessidade do time da SQA se envolver na avaliação do produto. Dá margem a interpretação. Porém, entendo que se o time de SQA estiver
SOI#3	Considere os itens de on-site review **SOI#3** abaixo e pontue segundo a gravidade:		
3.a	For the test case TC-507, the expected result of step 9 and the related comment do not match. According to the Company, the comment is incorrect, but it is not mandatory for performing the test.	2	Minor issue.
3.b	Completeness of testing is not ensured. Normal range and robustness test cases necessary to cover all conditions introduced in a requirement should be systematically developed. All assessments on Test Case Files have found incomplete normal range test cases, and almost total absence of robustness test cases.	5	
3.c	The verification method selected for the HLR-1788 is test, but the requirement cannot be verified by test. Considering the current coverage analysis approach, this HLR-1788 cannot be captured by any verification activity.	4	Se o requisito não pode ser verificado por teste, deve-se justificar outro método de verificação e apresentar artefato.
3.d	The review of the test scripts used a single checklist to support the whole review for the initial baseline. The same situation was observed in the test results review. The use of a single checklist can complicate traceability between the questions and the artifacts. However, no cases of non-compliance related to it were found.	1	Apesar do uso de um único checklist dificultar rastreio entre as questões e os artefatos, e como não foi identificado nenhum non-compliance, entendo que não é a melhor abordagem, porém não
3.e	Company-X relies on the output of Tool-Y to fulfill structural coverage objective. This tool reduces objective A7-5, 6 and 7, may fail to detect errors, and its output is not verified. However, the Tool-Y is not qualified.	5	Se o output da tool não é verificada ela precisa ser qualificada. Deveria revisar o output da tool ou qualificá-la. Pra mim issue preocupante.
3.f	In all assessments on Test Case Files, it was found cases of incorrect traceability between requirements and test cases, raising concerns on systemic issue. Test cases are long, complex, and the traceability granularity is deficient. The traceability should be accurate to ensure requirements-based test coverage analysis	5	Issue sistemático, potencial para grande retrabalho nos test cases.

Figure-D.4 provides an example of a spreadsheet filled-up for PART-2:

Figure-D.4: Spreadsheet filled-up (PART-2) by a survey participant



Objetivo: **Obter o *feedback* do especialista para os candidatos a compor a métrica.**

1. Preencha com valores de **0** a **3**, segundo a gravidade (ou relevância) crescente, sendo **0** para nenhuma gravidade e **3** para o item mais grave do grupo.
2. Cada grupo deverá ter pelo menos um item pontuado com valor **3.**
3. Após o preenchimento, reavalie para manter a consistência dentro do mesmo grupo. Não se preocupe com a consistência entre grupos distintos.

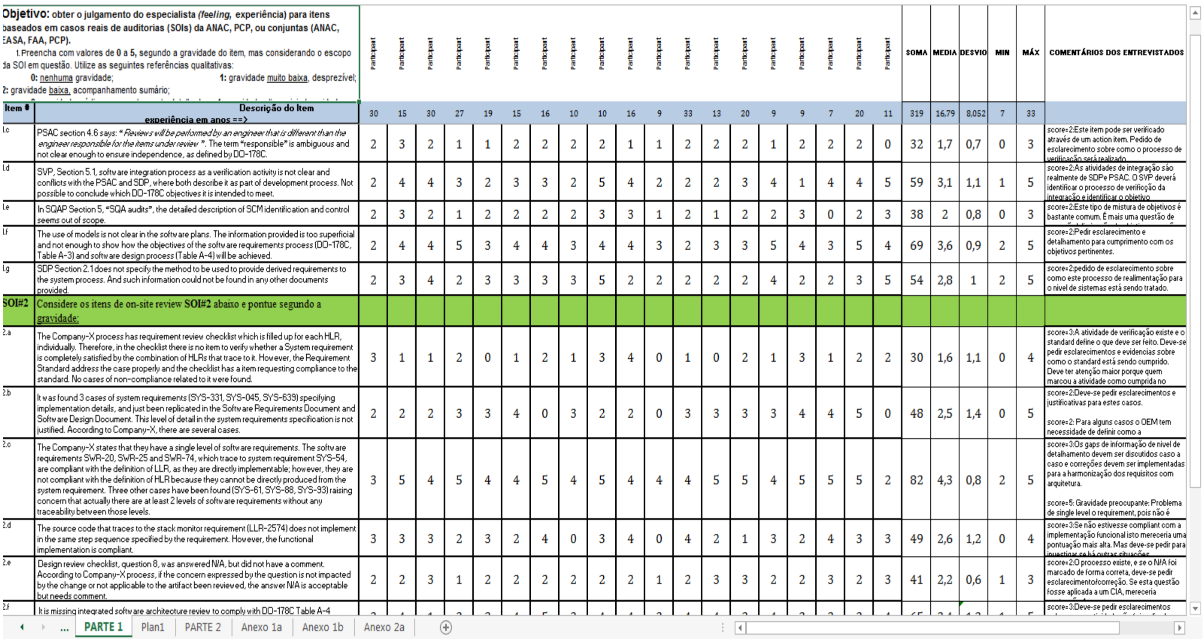
Item #	Descrição do Item	Pontuação	Comentário do especialista
1.d	A informação está confusa, ambígua;	2	
1.e	A informação está clara e completa, mas é considerada não aceitável;	3	Revisado depois do esclarecimento..
1.f	A informação está superficial ou incompleta;	1	
1.g	Não foi possível encontrar nos documentos fornecidos a informação requerida para compliance	3	Falta de compliance é igual à não-compliance.
Grupo-2	**(escopo SOI#2 e SOI#3) CANDIDATO "FINALIDADE":** Em relação à finalidade do item registrado na SOI#2 ou SOI#3, pontue os casos a seguir: **(ATENÇÃO:** atribua **3** para o item que considerar o mais grave do grupo **!)**		
2.a	Uma sugestão para melhoria de processo detectada durante a SOI. Entretanto, o processo já é considerado suficiente para compliance.	0	
2.b	Uma solicitação para correção pontual de deficiência de processo (ou aderência a este) detectada durante a SOI.	1	
2.c	Um registro de non-compliance detectado na SOI.	3	É o caso mais grave, pois fere diretamente o compliance.
2.d	Uma solicitação de informação adicional com potencial de confirmar uma non-compliance que não foi conclusiva durante a SOI.	2	
2.e	Uma solicitação de informação adicional, mas a priori sem impacto em nenhum item relevante discutido na SOI.	1	
Grupo-3	**(escopo SOI#2 e SOI#3) CANDIDATO "ARTEFATO":** Em relação ao tipo de dado contra o qual o item foi aberto, pontue os casos a seguir. Considere apenas os tipos de dados, ou seja, abstraia-se dos niveis de gravidade dentro de cada caso: **(ATENÇÃO:** atribua **3** para o item que considerar o mais grave do grupo **!)**		
3.a	Item aberto contra planos e standards;	3	Planos são a chave para o compliance. Não seguindo os planos previamente acordados na
3.b	Item aberto contra requisito, design, código ou PDI (e.g., requisito ambíguo, arquitetura incompatível com requisitos, código não implementa completamente o requisito);	3	Não fica claro se o item é "sistêmico", então preferi ser conservativo.
3.c	Item aberto contra Verification Cases and Procedures (e.g., test cases/procedures deficientes, test environment não representativo, estratégia de análise insuficiente);	2	

ORIENTAÇÃO | PARTE 1 | **PARTE 2**

STEP-5 Organizer compiles the answers and generates statistic data and graphs:

All responses were grouped in a single worksheet, the mean and deviation were calculated for each item, as well as the deviation of each participant from the group average, and all comments were captured in that worksheet. Figure-D.5 partially shows the PART-1 tab of the worksheet. Charts have been generated and will be provided in the next sections.

Figure-D.5: Spreadsheet consolidated by the survey organizer



Objetivo: **obter o julgamento do especialista *(feeling,* experiência) para itens baseados em casos reais de auditorias (SOIs) da ANAC, PCP, ou conjuntas (ANAC, EASA, FAA, PCP).**

1. Preencha com valores de **0** a **5,** segundo a gravidade do item, mas considerando o escopo da SOI em questão. Utilize as seguintes referências qualitativas:
0: nenhuma gravidade; **1:** gravidade muito baixa, desprezível; **2:** gravidade baixa, acompanhamento sumário;

Item #	Descrição do Item	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	Participant	SOMA	MEDIA	DESVIO	MIN	MÁX	COMENTÁRIOS DOS ENTREVISTADOS
	experiência em anos ==>	30	15	30	27	19	15	16	10	10	16	9	33	13	20	9	9	7	20	11	319	16,79	8,052	7	33	
1.c	PSAC section 4.6 says: "*Reviews will be performed by an engineer that is different than the engineer responsible for the items under review* ". The term "responsible" is ambiguous and not clear enough to ensure independence, as defined by DO-178C.	2	3	2	1	1	2	2	2	2	1	1	2	2	2	1	2	2	2	0	32	1,7	0,7	0	3	score=2:Este item pode ser verificado através de um action item. Pedido de esclarecimento sobre como o processo de verificação será realizado
1.d	SVP, Section 5.1, software integration process as a verification activity is not clear and conflicts with the PSAC and SDP, where both describe it as part of development process. Not possible to conclude which DO-178C objectives it is intended to meet.	2	4	4	3	2	3	3	2	5	4	2	2	2	3	4	1	4	4	5	59	3,1	1,1	1	5	score=2:As atividades de integração são realmente de SDPe PSAC. O SVP deverá identificar o processo de verificção da integração e identificar o objetivo
1.e	In SQAP Section 5, "SQA audits", the detailed description of SCM identification and control seems out of scope.	2	3	2	1	2	2	2	2	3	3	1	2	1	2	2	3	0	2	3	38	2	0,8	0	3	score=2:Este tipo de mistura de objetivos é bastante comum. É mais uma questão de
1.f	The use of models is not clear in the software plans. The information provided is too superficial and not enough to show how the objectives of the software requirements process (DO-178C, Table A-3) and software design process (Table A-4) will be achieved.	2	4	4	5	3	4	4	3	4	4	3	2	3	3	5	4	3	5	4	69	3,6	0,9	2	5	score=2:Pedir esclarecimento e detalhamento para cumprimento com os objetivos pertinentes.
1.g	SDP Section 2.1 does not specify the method to be used to provide derived requirements to the system process. And such information could not be found in any other documents provided.	2	3	4	2	3	3	3	3	5	2	2	2	2	2	4	2	2	3	5	54	2,8	1	2	5	score=2:pedido de esclarecimento sobre como este processo de realimentação para o nivel de sistemas está sendo tratado.
SOI#2	Considere os itens de on-site review **SOI#2** abaixo e pontue segundo a gravidade:																									
2.a	The Company-X process has requirement review checklist which is filled up for each HLR, individually. Therefore, in the checklist there is no item to verify whether a System requirement is completely satisfied by the combination of HLRs that trace to it. However, the Requirement Standard address the case properly and the checklist has a item requesting compliance to the standard. No cases of non-compliance related to it were found.	3	1	1	2	0	1	2	1	3	4	0	1	0	2	1	3	1	2	2	30	1,6	1,1	0	4	score=3:A atividade de verificação existe e o standard define o que deve ser feito. Deve-se pedir esclarecimentos e evidencias sobre como o standard está sendo cumprido. Deve ter atenção maior porque quem marcou a atividade como cumprida no
2.b	It was found 3 cases of system requirements (SYS-331, SYS-045, SYS-639) specifying implementation details, and just been replicated in the Software Requirements Document and Software Design Document. This level of detail in the system requirements specification is not justified. According to Company-X, there are several cases.	2	2	2	3	3	4	0	3	2	2	0	3	3	3	3	4	4	5	0	48	2,5	1,4	0	5	score=2:Deve-se pedir esclarecimentos e justificativas para estes casos. score=2: Para alguns casos o OEM tem necessidade de definir como a
2.c	The Company-X states that they have a single level of software requirements. The software requirements SWR-20, SWR-25 and SWR-74, which trace to system requirement SYS-54, are compliant with the definition of LLR, as they are directly implementable; however, they are not compliant with the definition of HLR because they cannot be directly produced from the system requirement. Three other cases have been found (SYS-61, SYS-88, SYS-93) raising concern that actually there are at least 2 levels of software requirements without any traceability between those levels.	3	5	4	5	4	4	5	4	5	4	4	4	5	5	4	5	5	5	2	82	4,3	0,8	2	5	score=3:Os gaps de informação de nivel de detalhamento devem ser discutidos caso a caso e correções devem ser implementadas para a harmonização dos requisitos com arquitetura. score=5: Gravidade preocupante: Problema de single level o requirement, pois não é
2.d	The source code that traces to the stack monitor requirement (LLR-2574) does not implement in the same step sequence specified by the requirement. However, the functional implementation is compliant.	3	3	3	2	3	2	4	0	3	4	0	4	2	1	3	2	4	3	3	49	2,6	1,2	0	4	score=3:Se não estivesse compliant com a implementação funcional isto mereceria uma pontuação mais alta. Mas deve-se pedir para
2.e	Design review checklist, question 8, was answered N/A, but did not have a comment. According to Company-X process, if the concern expressed by the question is not impacted by the change or not applicable to the artifact been reviewed, the answer N/A is acceptable but needs comment.	2	2	3	1	2	2	2	2	2	2	1	2	3	3	2	2	3	2	3	41	2,2	0,6	1	3	score=2:O processo existe, e se o N/A foi marcado de forma correta, deve-se pedir esclarecimento/correção. Se esta questão fosse aplicada a um CIA, mereceria
2.f	It is missing integrated software architecture review to comply with DO-178C Table A-4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	65	3,4	[illegible]	[illegible]	5	score=3:Deve-se pedir esclarecimentos

... | **PARTE 1** | Plan1 | PARTE 2 | Anexo 1a | Anexo 1b | Anexo 2a

STEP-6 Organizer discusses in a workshop the results with the participants:

A workshop has been performed for the following objectives:

a. Present the survey results (see sections D.3 and D.4);

b. Provide clarification for some cases whose result have revealed potential unclear or ambiguous instructions that may have led to misunderstandings (see section 5.3);

c. Discuss the acceptance of the suggested additional metrics (see section 4.4.3);

d. Discuss possible dependency among the metrics (see section 4.4.4);

STEP-7 Organizer generates the final survey result and divulges to participants:

Each participant received a dedicated spreadsheet containing their responses, mean and deviation of the group, their deviation from the group, indication of the items outside the standard deviation, and collection of all participants' comments. Relevant notes captured during the workshop were also provided. The result of each participant was not disclosed to the group.

D.3 – The survey results for PART-1

Objective of PART-1: to obtain the software safety specialist's judgment (feeling, experience) for issues based on real cases of audits (SOIs) of ANAC or joint (ANAC, EASA, FAA). Answered with values from 0 to 5, according to the severity of the issue, but considering the scope of the SOI, and using the following qualitative references:

0- No severity

1- Severity very low, negligible

2- Low severity, summarized follow-up is enough

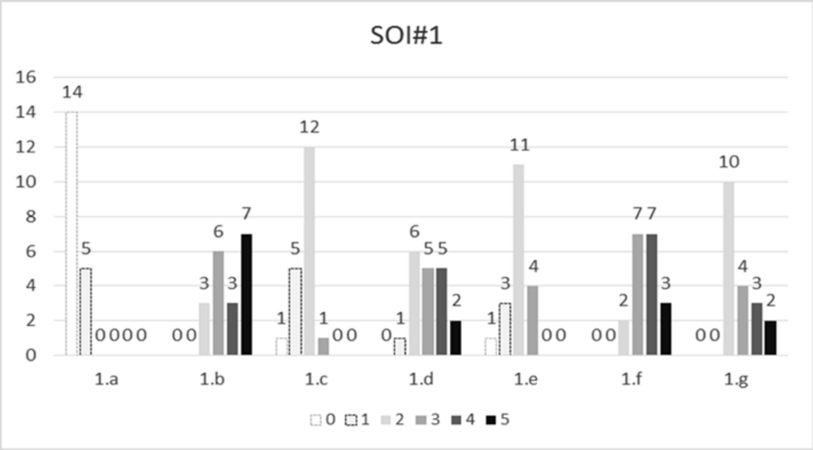
3- Medium severity, detailed follow-up required

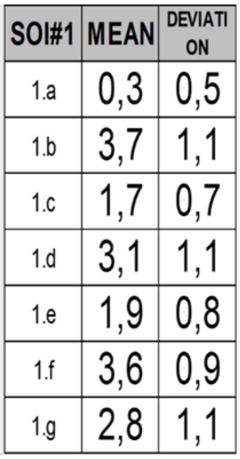
4- High severity, requiring careful follow-up

5. Severity of serious concern, requiring priority in the follow-up

Figure-D.6 shows the chart of the results from PART-1 for the SOI # 1 issues severity:

Figure-D.6: Chart for SOI#1 issues severity

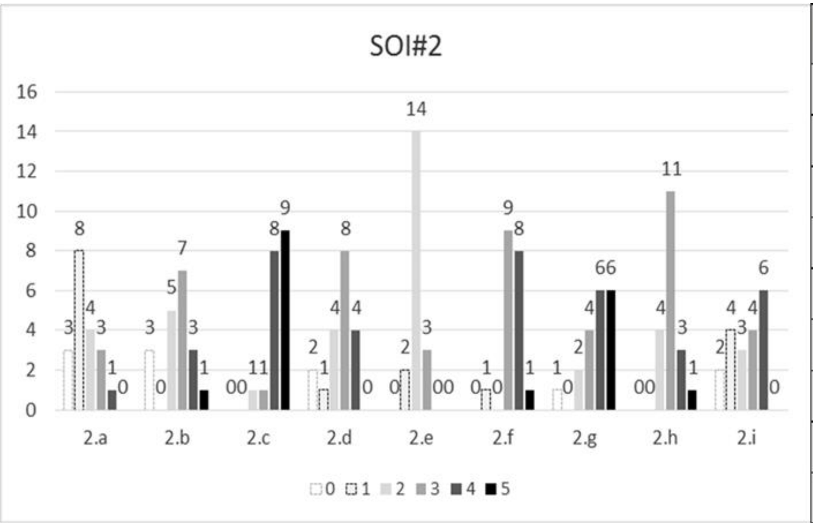


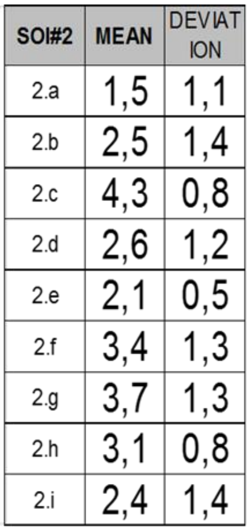


SOI#1	MEAN	DEVIATION
1.a	0,3	0,5
1.b	3,7	1,1
1.c	1,7	0,7
1.d	3,1	1,1
1.e	1,9	0,8
1.f	3,6	0,9
1.g	2,8	1,1

Figure-D.7 shows the chart of the results from PART-1 for the SOI # 2 issues severity:

Figure-D.7: Chart for SOI#2 issues severity

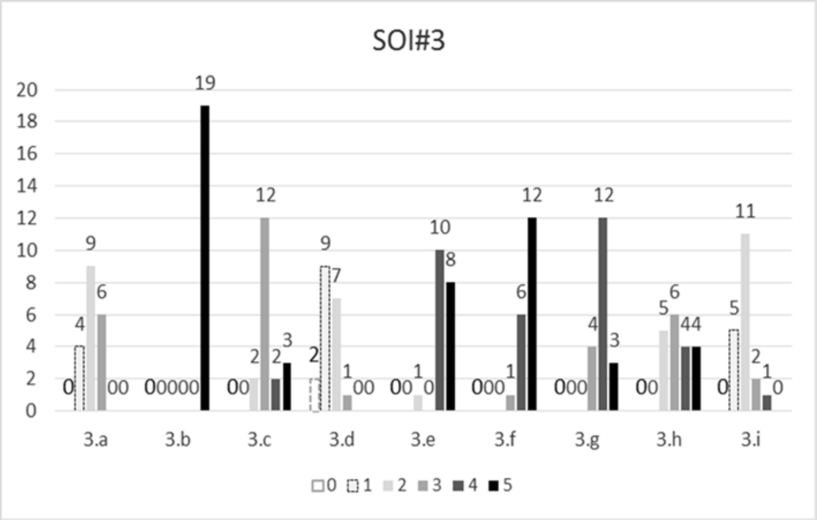


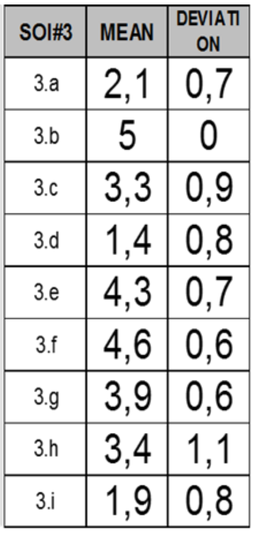


SOI#2	MEAN	DEVIATION
2.a	1,5	1,1
2.b	2,5	1,4
2.c	4,3	0,8
2.d	2,6	1,2
2.e	2,1	0,5
2.f	3,4	1,3
2.g	3,7	1,3
2.h	3,1	0,8
2.i	2,4	1,4

Figure-D.8 shows the chart of the results from PART-1 for the SOI # 3 issues severity. It is noticed that item 3.b obtained unanimity in the score (5 = severity of serious concern, priority follow-up). Item 3.c, despite the high deviation, still shows a tendency, but the same does not occur with item 3.h that does not indicate any tendency. This last one is a strong candidate to explore in the workshop (Ambiguous text? Misaligned concepts? Controversial points?). It was noted that in general the deviation is high, and assuming the issues were clearly described, this suggests that there is a certain subjectivity in the evaluation, and that a metric would have the potential to reduce it.

Figure-D.8: Chart for SOI#3 issues severity

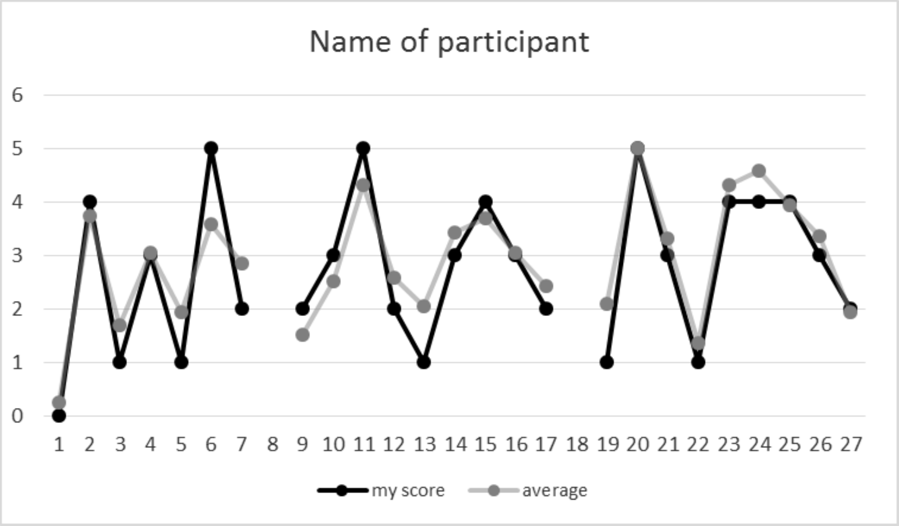




SOI#3	MEAN	DEVIATI ON
3.a	2,1	0,7
3.b	5	0
3.c	3,3	0,9
3.d	1,4	0,8
3.e	4,3	0,7
3.f	4,6	0,6
3.g	3,9	0,6
3.h	3,4	1,1
3.i	1,9	0,8

The following figures (D.9 to D.12) illustrate examples of participant scores compared to the group average for the spreadsheet PART-1. Figure D.9 shows a case of participant scores close to the average:

Figure-D.9: Scores close to the average (PART-1)

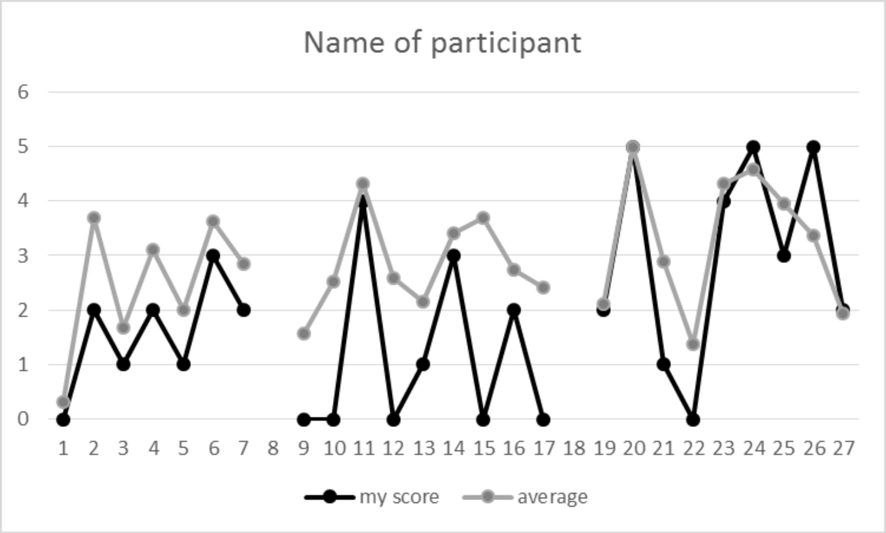


The specific participant scores are represented in black, while the group average scores are in grey. The three groups of segments represent the three audit stages, i.e., SOI#1 (score numbers 1 to 7), SOI#2 (score numbers 9 to 17) and SOI#3 (score numbers 19 to 27). The SOI#4 has not been included because ANAC has not performed any SOI#4 up to this time due to the scope of that type of stage. Considering that the score is always an integer value (i.e., 0 to 5), while the average can be fractional, almost all scores are inside the average, except the scores number

9 and 17, which are both one unit below. Moreover, one can notice that the scores of the specific participant follow the group tendency, i.e., both lines are synchronized in ascending and descending sequence.

The figure-D.10 shows a case where the participant scores are far from the average, and with tendency to less rigor.

Figure-D.10: Scores showing tendency to less rigor (PART-1)



The scores that are far from the average are all below it (i.e., scores number 2, 9, 10, 12, 15, 17, 21, 22, except 26), which show a tendency to less rigor than the average of participants. Nevertheless, the scores follow the average tendency (i.e., both lines are synchronized) with few exceptions.

Figure-D.11 shows a case where the points that were far from the average are all above (4, 7, 9, 16, 22), which suggests a tendency to more rigor. The scores that are far from the average are all above it (i.e., scores number 4, 7, 9, 22), which show a tendency to more rigor than the average of participants. Nevertheless, the scores follow the average tendency (i.e., both lines are synchronized) with few exceptions.

Figure-D.11: Scores showing tendency to higher rigor (PART-1)

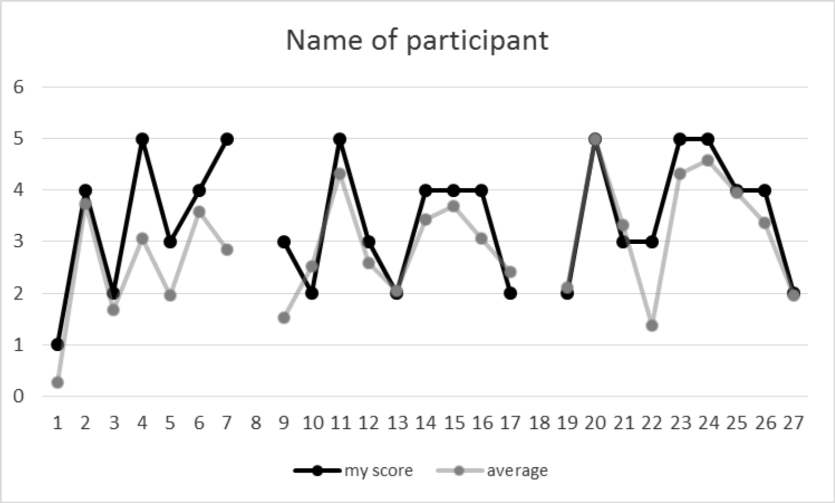
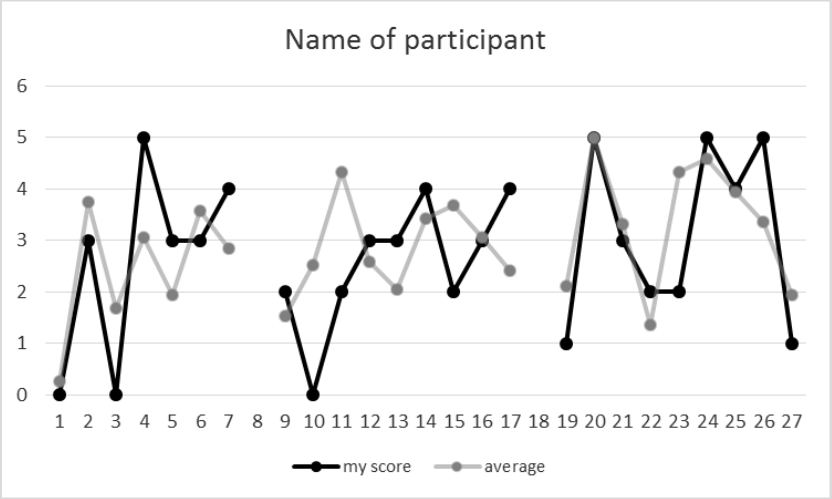


Figure-D.12 shows a case where several points were far from average, but do not suggest any tendency.

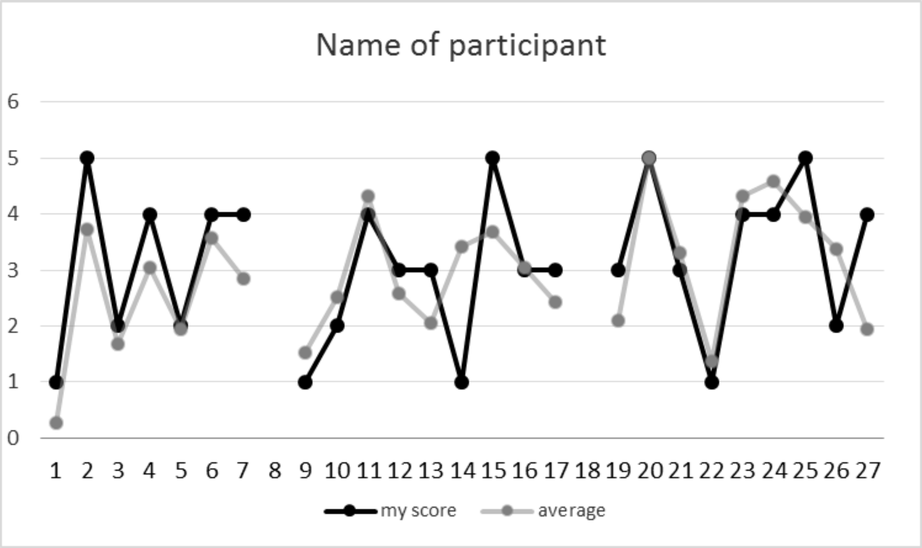
Figure-D.12: Scores far from the average, but without any tendency (PART-1)



For the scores that are far from the average, some are below it (i.e., 3, 10, 11, 15, 23) while others are above it (i.e., 4, 17, 26), which does not suggest any tendency and does not follow the average.

The four cases presented so far, i.e., close to average, tendency to less rigor, to more rigor, and without any tendency are all normal cases expected in any survey, and do not invalidate the survey result. The next cases are examples that have questioned the survey result and have demanded some analysis and adjustments during the workshop. The figure-D.13 shows a case of participant scores very close to the group average, but with a specific score very distant from the average.

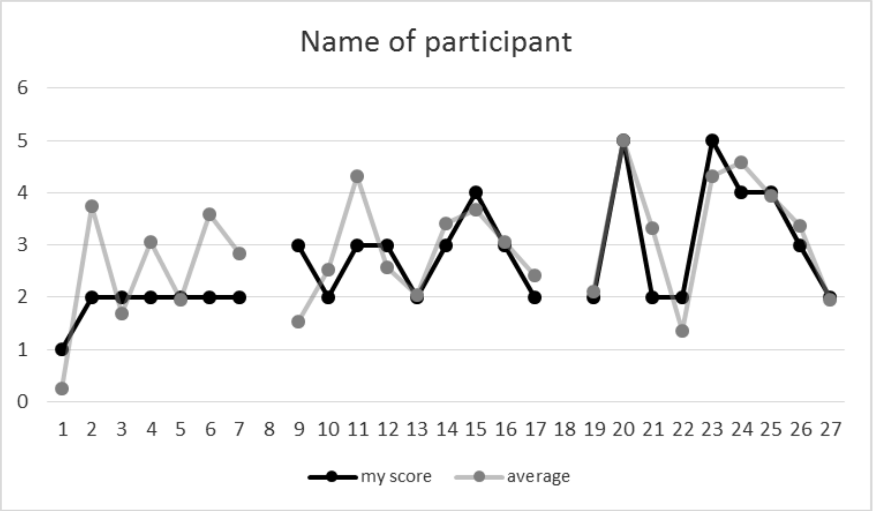
Figure-D.13: Scores close to the average, but with one case very distant (PART-1)



Almost all scores are inside the average or very close to it, and both lines are synchronized. However, one specific score (i.e., number 14) is two units below the average and in opposition to the average tendency, i.e., the first is descending while the latter is ascending. Is it a case where the participant misunderstood the issue? Or does the participant have a peculiar interpretation of this issue severity? This case has been selected for workshop discussion.

The figure-D.14 shows a case of a participant assigning scores with fixed values during an interval, without following the average tendency.

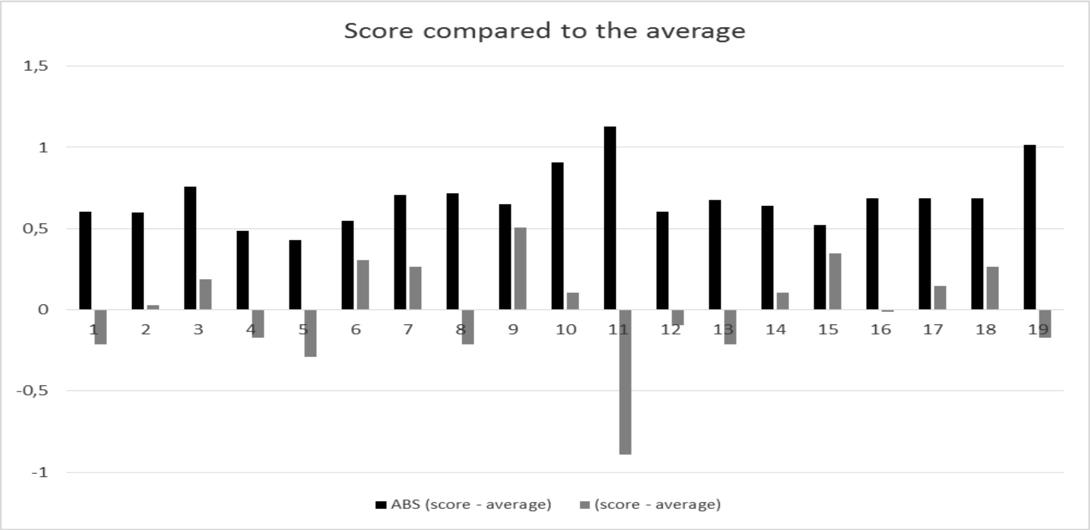
Figure-D.14: Scores with fixed value, not following the average tendency



Similar with the previous case, the exception being the interval with fixed values, almost all scores are inside the average or very close to it and both lines are synchronized. In the specific interval between score number 2 and 7, there are 6 sequential scores with fixed value equal 2, not following the average tendency at all. Is it a case of misunderstanding the instructions for the SOI#1 group? Or does the participant have a fixed criterion for this interval? A mind set? This case has also been selected for workshop discussion.

Figure-D.15 shows an overall participant performance comparing to the average:

Figure-D.15: Participants performance comparing to the average



The vertical bars in black represent the absolute mean difference between the participant's score and the group average. In other words, it measures the average distance between the participant scores and the group scores average. The vertical grey bars take into account the signal, i.e., the average difference between the participant’s score and the group score average can be either positive or negative, which indicates the tendency to more rigor (i.e., positive grey bar) or less rigor (i.e., negative grey bar). The participant 5 is the closest to the average (see figure-D.9), the participant 9 is the one with the most rigorous tendency (see figure-D.11), the participant 11 is the one with the least rigor tendency (see figure-D.10), and the participant 19 is one with high distance from the average, but without any tendency (see figure-D.12).

D.4 – The survey results for PART-2

The objective of PART-2 was to obtain from the software safety specialists the quantitative values for some metrics, as well as quantitative relevance of the metrics. The quantitative values were obtained for four metrics: “*document evaluation*”, “*purpose of the issue*”, “*artifacts impacted*” and “*root cause*”. Values from 0 to 3 were chosen according to the severity, being 0 for no severity and 3 for the most severe item in the metric. Each metric must have at least one item scored with 3. The survey results were very positive, as in the analysis of the scores provided by the participants there was always a tendency to converge the values. Exceptions (e.g., high deviation) were discussed in a dedicated workshop to identify possible ambiguities and unclearness that might have generated the problem. The following tables and charts present the quantitative values obtained by the survey, and the comments were mainly captured during the workshop where those data were discussed with the participants. The table-D.1 presents the result for the metric "*document evaluation*".

Table-D.1: Quantitative values for metric M1 "*document evaluation*"

Item#	Metric "document evaluation"	MEAN	DEVIATION
1.a	The information contains editorial errors (typos);	0.1	0.3
1.b	The information is out of context, e.g., recorded in an inappropriate section or document;	0.8	0.5
1.c	The information is inconsistent between sections or documents;	1.8	0.5
1.d	The information is confused, ambiguous;	2	0.5
1.e	The information is clear and complete, but is considered unacceptable;	2.9	0.3
1.f	The information is superficial or incomplete;	1.9	0.7
1.g	Could not find in the provided documents the required information for compliance.	2.5	0.7

By analyzing the results, it is possible to divide them in three severity levels, as follow:

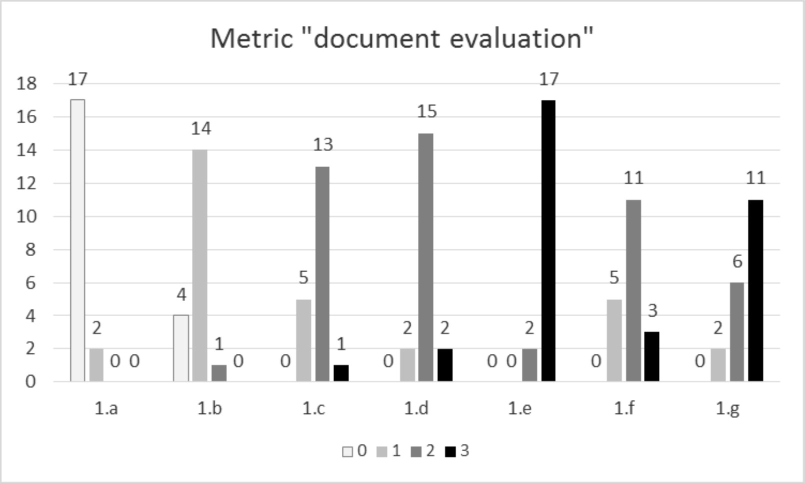
d. **Low** severity: score below 1; items 1.a and 1.b; related to editorial issues without impact in the required information;

e. **Medium** severity: score close to 2; items 1.c, 1.d and 1.f; related to the quality of the information, but without clear impact in compliance;

f. **High** severity: score close to 3; items 1.e and 1.g; clearly related to non-compliance.

The figure-D.16 provides the frequency of the scores for each item of the metric.

Figure-D.16: Chart for metric M1 "*document evaluation*"



The extreme cases almost reached consensus among the participants, i.e., for editorial issues (item 1.a) almost all participants scored zero (two exceptions), and for clear information that does not comply (item 1.e) almost all participants scored 3 (also with two exceptions). The item 1.g, though considered of high severity (average = 2.5), has high deviation, which can be explained because many auditors consider the document evaluation as sampling-based, i.e., not exhaustive. As such, it is usually opened an issue requesting the company to indicate where in the documents provided the information can be found. In that case, only if confirmed the absence of the information the issue would be related to a non-compliance (score 3), otherwise it could also be scored as 2. The table-D.2 presents the result for the metric "*purpose of the issue*".

Table-D.2: Quantitative values for metric M2 "*purpose of the issue*"

Item#	Metric "purpose of the issue"	MEAN	DEVIA TION
2.a	A sugestion for process improvement detected during the audit. However, the process is considered sufficient for compliance;	0.2	0.4
2.b	An issue to correct a punctual process deficiency (or adherence to the process) detected during the audit;	1.3	0.5
2.c	An issue to record a non-compliance and request a closure approach;	2.8	0.4
2.d	An issue to request additional information, which may drive to a non-compliance that was not conclusive during the audit;	2.1	0.5
2.e	An issue to request additional information, but a priori without any impact in items discussed during the audit;	0.7	0.5

It was observed that for this metric the results were very clear, with low deviation and well-define values. It is also possible to divide them in three severity levels, as follow:

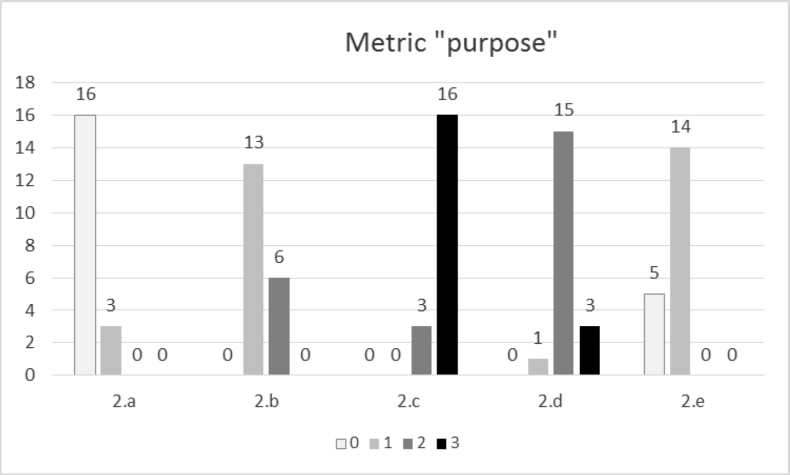
a. **Low** severity: score below 1; items 2.a and 2.e; issues without any impact in compliance;

b. **Medium** severity: score above 1; item 2.b; corrections with known and controlled impact;

c. **High** severity: score above 2; items 2.c and 2.d; clear (or potential) non-compliance with impact not yet known.

The figure-D.17 provides the frequency of the scores for each item of the metric.

Figure-D.17: Chart for metric M2 "*purpose of the issue*"



Almost all items had only two values selected for score (exception is 2.d), with high predominance of one value. The distribution is very gradual, e.g., the least severe item has predominance of the value zero, for the next less severe the predominance was 1 (and zero as the second value), the next also has 1 as predominant (but 2 as the second), the next has predominance of 2, and the most severe has predominance of 3. The table-D.3 presents the result for the metric "*type of artifact impacted*". Although the metric has many items to consider, it is also possible to divide them in three severity levels, as follow:

a. **Low** severity: score close to zero; item 3.j; related to informal artifacts;

b. **Medium** severity: score close to 2; items 3.a, 3.e, 3.f, 3.g, 3.h and 3.i; artifacts used as support for the development and verification activities;

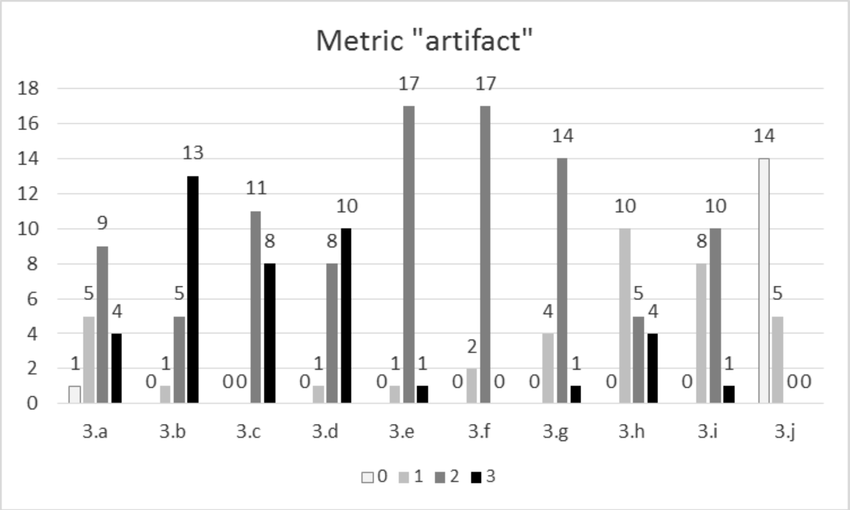
c. **High** severity: score close to 2.5; items 3.b, 3.c, and 3.d; artifacts directly related to development and verification activities.

Table-D.3: Quantitative values for metric M3 "*type of artifact impacted*"

Item#	Metric "type of artifact impacted"	MEAN	DEVIATION
3.a	Issue opened against plans and standards;	1.8	0.8
3.b	Issue opened against requirement, design, code or configuration data (e.g., ambiguous requirement, architecture incompatible with requirements, code does not fully implement the requirement);	2.6	0.6
3.c	Issue opened against verification cases and procedures (e.g., defective test cases/procedures, non-representative test environment, insufficient analysis strategy);	2.4	0.5
3.d	Issue opened against verification results and related artifacts (e.g., checklist filled with errors, checklist questions insufficient for revision needs, incorrect test result not detected by the review);	2.5	0.6
3.e	Issue opened against traceability (e.g., requirement points to wrong parent requirement, insufficient granularity);	2	0.3
3.f	Issue opened against tools (e.g., poor qualification report, justification for non-qualification is unacceptable);	1.9	0.3
3.g	Issue opened against Problem Reports (PR);	1.8	0.5
3.h	Issue opened against Software Configuration Management Records;	1.7	0.8
3.i	Issue opened against Software Quality Assurance (SQA) Records;	1.6	0.6
3.j	Issue opened against informal data (e.g., an SQA spreadsheet for informal control not planned for use by the process).	0.3	0.5

The figure-D.18 provides the frequency of the scores for each item of the metric.

Figure-D.18: Chart for metric M3 "type of artifact impacted"



This metric also shows tendency to converge the values, but it is the least one among the four cases (i.e., M_1, M_2, M_3, M_4). Only three cases of items with two values selected (i.e., 3.c, 3.f and 3.j), and only two cases of clear predominance of one value (i.e., 3.e and 3.f). There are three cases where two values were competing for the top score (i.e., 3.c, 3.d and 3.i), and one case where all possible values were selected, though with a converging value (i.e., 3.a). The item 3.h has an unusual distribution, which is more common for cases where the converging value is at the edge (i.e., 0 or 3), for example 3.b and 3.d. The table-D.4 presents the result for the metric "*root cause*".

Table-D.4: Quantitative values for metric M4 "*root cause*"

Item#	Metric "root cause"	MEAN	DEVIA TION
4.a	It was only a reviewer slip, an isolated case;	0.6	0.5
4.b	The amount and complexity of the information needed for the review may have contributed to the reviewer mistake ;	1.9	0.7
4.c	Similar cases have been found involving the same reviewer, raising suspicion of insufficient training;	1.7	0.5
4.d	The training material was deficient, raising suspicion that the reviewer did not understand enough the activity to perform ;	1.8	0.6
4.e	The process followed was clear but incorrect, leading the reviewer to the mistake;	2.9	0.3
4.f	The process followed was not clear, which may have contributed to the reviewer mistake;	2	0.5

By analyzing the results, it is possible to divide them in three severity levels, as follow:

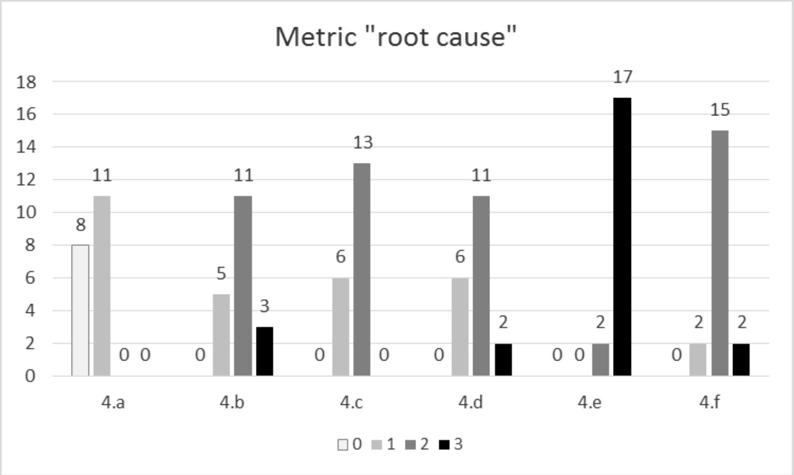
a. **Low** severity: score below 1; item 4.a; related to non-systemic error;

b. **Medium** severity: score close to 2; items 4.b, 4.c, 4.d and 4.f; a potential systemic error, but not evident;

c. **High** severity: score close to 3; item 4.e; clearly a systemic error.

The figure-D.19 provides the frequency of the scores for each item of the metric.

Figure-D.19: Chart for metric M4 "root cause"



All items show clear convergence to a value, with the exception of the item 4.a where two values were competing for the top score. Items 4.e and 4.f have clear predominance of one score. Half of the items have only two values selected for score (i.e., 4.a, 4.c and 4.e).

Quantitative relevance for each metric:

Regarding the quantitative relevance of each metric, values from 0 to 3 were also chosen according to the relevance, being 0 for no relevance and 3 for the most relevant metric. At least one metric had to be scored with 3. The table-D.5 presents the result of quantitative relevance of each metric.

Table-D.5: Relevance of each metric in quantitative values

Item#	Relevance of each metric	Mean	Deviation
5.a	Metric M2: purpose of the issue	2.2	0.8
5.b	Metric M3: type of artifact impacted by the issue	1.8	0.8
5.c	Metric M4: root cause of the issue	2.2	0.7
5.d	Metric M5: distance from the issue to the final product	1.6	0.8
5.e	Metric M6: amount of artifacts impacted by the issue	2.3	0.7
5.f	Metric M7: adequacy of the issue regard to the stage of the audit	1.6	0.9

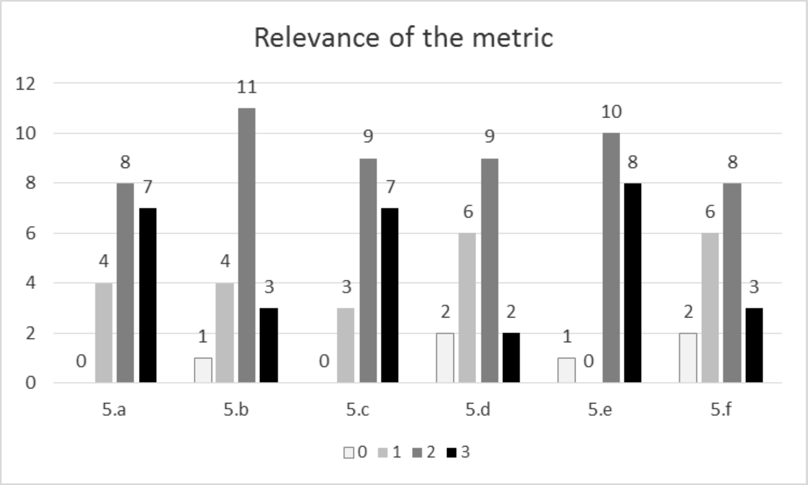
There is not much difference among the metrics for the quantitative relevance, and all of them had values close to 2. Nevertheless, it is possible to classify them in two levels of relevance:

a. **Medium-high** relevance: score above 2; items 5.a, 5.c and 5.e; related to the essence of the issue (What for? Why it happened? How much damage it caused?);

b. **Medium** relevance: score below 2; items 5.b, 5.d and 5.f; related to process and life cycle data (How far from the mainstream data? How far from the final data? How delayed from the current process?).

The deviation is higher than those cases of quantitative values for each metric. The figure-D.20 provides more details of the scores distributions. Although the scores are more spread (high deviation), all cases have shown a distribution close to a normal, except the item 5.e. There were no cases of consensus or cases with two values selected. They all have selected 3 or all values for score. However, all cases had the top score as 2, which is coherent with the average of each metric (all close to 2). Only the item 5.b had predominance of one score, while all others had two values competing for the top score.

Figure-D.20: Chart for relevance of each metric



Remark: for the workshop results refer to section 4.4.3 "*Identification of new metrics*" and 4.4.4 "*Discussion on dependency among metrics*".

D.5 –Closure comments

A major contribution of the survey that was not originally planned was to serve as a tool for the self-assessment of the software safety specialist and for the alignment of concepts and rigor among the specialists (auditors). It has been also studied the possibility of applying the survey within aviation industry. In this case, the survey would also be used as a tool to aid in the training of future software auditors.

APPENDIX E: GLOSSARY

Accident – An unintentional event or sequence of events that causes death, injury, environmental or material damage (STOREY, 1996); Undesirable and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss (LEVESON, 1995); Undesirable event arising from operation of any project-specific item that results in (a) human death or injury, (b) loss of, or damage to, project hardware, software or facilities that can then affect the accomplishment of the mission, (c) loss of, or damage to, public or private property, or (d) detrimental effects on the environment (ECSS-S-ST-00-01C, 2012).

Assurance – The planned and systematic actions necessary to provide adequate confidence and evidence that a product or process satisfies given requirements (RTCA/DO-178C, 2011); Planned and systematic activities implemented, and demonstrated as needed, to provide adequate confidence that an entity fulfils its requirements (ECSS-S-ST-00-01C, 2012).

Audit – An independent examination of the software life cycle processes and their outputs to confirm required attributes (RTCA/DO-178C, 2011); Systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which audit criteria are fulfilled (ECSS-S-ST-00-01C, 2012).

Baseline – The approved, recorded configuration of one or more configuration items, that thereafter serves as the basis for further development, and that is changed only through change control procedures (RTCA/DO-178C, 2011); Set of information which describes exhaustively a situation at a given instant of time or over a given time interval (ECSS-S-ST-00-01C, 2012).

Certification – For the civil aviation, the legal recognition that a product, service, organization, or person complies with the applicable requirements. Such certification comprises the activity of technically checking the product, service, organization, or person and the formal recognition of compliance with the applicable requirements by issue of a certificate, license, approval, or other documents as required by national laws and procedures (ARP 4754A, 2010); Procedure by which a party gives formal assurance that a person or an organization acts, or a product is, in compliance with specified requirements (ECSS-S-ST-00-01C, 2012).

Dead code – Executable Object Code (or data) which exists as a result of a software development error but cannot be executed or used in any operational configuration of the target computer environment (RTCA/DO-178C, 2011).

Derived requirements – Requirements produced by the software development processes which (a) are not directly traceable to higher level requirements, and/or (b) specify behavior beyond that specified by the system requirements or the higher-level software requirements (RTCA/DO-178C, 2011).

Development assurance - all of those planned and systematic actions used to substantiate, at an adequate level of confidence, that development errors have been identified and corrected such that the system satisfies the applicable certification basis (ARP 4754A, 2010).

Error – With respect to software, a mistake in requirements, design, or code (RTCA/DO-178C, 2011); A design flaw or deviation from a desired or intended state (LEVESON, 1995).

Failure - The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered (RTCA/DO-178C, 2011); The event resulting in an item being no longer able to perform its required function (ECSS-S-ST-00-01C, 2012); The nonperformance or inability of the system or component to perform its intended function for a specified time under specified environmental conditions (LEVESON, 1995).

Fault – A manifestation of an error in software through the executable code. A fault, if it occurs, may cause a failure (RTCA/DO-178C, 2011); State of an item characterized by inability to perform as required. A fault can generate a failure (ECSS-S-ST-00-01C, 2012); A defect within the system (STOREY, 1996).

Hazard – A condition resulting from failures, external events, errors, or combinations thereof where safety is affected. A situation that can lead to an accident (ARP 4754A, 2010); Existing or potential condition that can result in a mishap (ECSS-S-ST-00-01C, 2012); A situation in which there is actual or potential danger to people or to the environment (STOREY, 1996); A state or set of conditions of a system (or an object) that together with other conditions in the environment of the system (or object), will lead inevitably to an accident (loss event) (LEVESON, 1995).

High-level requirements – Software requirements developed from analysis of system requirements, safety-related requirements, and system architecture (RTCA/DO-178C, 2011).

Low-level requirements – Software requirements developed from high-level requirements, derived requirements, and design constraints from which Source Code can be directly implemented without further information (RTCA/DO-178C, 2011).

Measure – Provides a quantitative indication of the extent, amount, dimension, capacity, or size of some attribute of a product or process (PRESSMAN, 2015).

Measurement – The act of determining a measure (PRESSMAN, 2015).

Metric - A quantitative measure of the degree to which a system, component, or process possesses a given attribute (PRESSMAN, 2015).

Process – A collection of activities performed in the software life cycle to produce a definable output or product (RTCA/DO-178C, 2011); Set of interrelated or interacting activities which transform inputs into outputs (ECSS-S-ST-00-01C, 2012).

Reliability – The probability that a piece of equipment or component will perform its intended function satisfactorily for a prescribed time and under stipulated environmental conditions (LEVESON, 1995); The probability of a component, or system, functioning correctly over a given period of time under a given set of operating conditions (STOREY, 1996); The ability of an item to perform a required function under given conditions for a given time interval (ECSS-S-ST-00-01C, 2012).

Risk – A combination of the frequency or probability of a specified hazardous event, and its consequence (STOREY, 1996); The hazard level combined with (1) the likelihood of the hazard leading to an accident (sometimes called danger) and (2) hazard exposure or duration (sometimes called latency) (LEVESON, 1995); The product of the probability of existence of the hazard by the magnitude of its consequences. The combination of the frequency (probability) of an occurrence and its associated level of severity (ARP 4754A, 2010); Undesirable situation or circumstance that has both a likelihood of occurring and a potential negative consequence on a project (ECSS-S-ST-00-01C, 2012).

Safety – The state in which risk is acceptable (ARP 4754A, 2010); A property of the system that this will not endanger human lives or the environment (STOREY, 1996); Freedom from accidents or losses (LEVESON, 1995); State where an acceptable level of risk is not exceeded (ECSS-S-ST-00-01C, 2012).

Validation – The process of determining that a system is appropriate for its purpose (STOREY, 1996); The process of determining that the requirements are the correct requirements and that they are complete (RTCA/DO-178C, 2011); Process which demonstrates that the product is able to accomplish its intended use in the intended operational environment (ECSS-S-ST-00-01C, 2012); Process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product (ECSS-E-ST-40C, 2009).

Verification – The process of determining that a system, or module, meets its specification (STOREY, 1996); The evaluation of the outputs of a process to ensure correctness and consistency with respect to the inputs and standards provided to that process (RTCA/DO-178C, 2011); Process which demonstrates through the provision of objective evidence that the product is designed and produced according to its specifications and the agreed deviations and waivers, and is free of defects (ECSS-S-ST-00-01C, 2012); Process to confirm that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input (ECSS-E-ST-40C, 2009).