One approach to the use of the practices of CMMI-DEV V1.3 level 2 in a process of development of Embedded Systems

Magda A. Silvério Miyashiro
IMT – Instituto Mauá de Tecnologia
São Caetano do Sul, Brasil
magda.miyashiro@maua.br
INPE – Instituto Nacional de Pesquisa Espaciais
São José dos Campos, Brasil

Maurício G. V. Ferreira
CCS – Centro de Controle de Satélite
INPE – Instituto Nacional de Pesquisa Espaciais
São José dos Campos, Brasil
mauricio@ccs.inpe.br

***Abstract* — Currently the major concern of software engineering is the quality of Embedded Computing Systems, as they are a classification of software that are strongly present in our daily lives so that often we do not realize. Are composed of hardware and software encapsulated and generally dedicated to a single activity a major characteristic of such systems is the ability to perform very important tasks with little computational resource which represents a high degree of constraints to it is development and operation. It is importance is intensified when they are embedded in aircraft, medical equipment, automobiles, space systems among others that are considered critical, since if it fails, can cause major disasters. The dependence of society, different applications and technological developments are delegating all flexibility of these systems for their part formed by software, which makes it even more critical thus requiring appropriate and specific procedures in their production. The difficulty of software Engineering in meeting this demand is the specificity of each software component to be developed. The use of CMMI-DEV/V1.3 (Process Improvement of Brazilian Software) a model of international reference that directs the execution of practices necessary for the maturity of a software development process may represent a breakthrough in developing such systems one since it is development, we need to use processes with defined procedures to identify and assist in the development of products with these features and thus allow errors can be eliminated or minimized through the activities that must be performed in all stages of development This paper presents part of a study being carried out to develop a process framework that addresses all phases of a development process of embedded systems, organized into phases, activities and document templates that induce developers carrying out the good practices suggested by the level of quality CMMI-DEV/V1.3 model.**

Keywords— Embedded System; Hardware; Software; Software Quality; CMMI; Maturity; Critical Software; Process; Generic Practices; Specific Practices.

I. INTRODUCTION

The "Software Crisis" suffered by the entire community of development and use of software products, which occurred in the 60s, was due to the growing number of computer-based systems Fifty years after the crisis, there was great progress in development software, such as the emergence of tools, methods and techniques, however, the picture has not changed as expected. A category of software that has undergone a major evolution was embedded systems, software products that are characterized by a strong interaction between hardware and software dedicated to a specific application. In general, are parts of a larger system and react to it is stimuli [5]. Originally electronics were structures of an electronic circuit, with the evolution of the software and to decrease the cost of production is passed using software as part of it is components which may be responsible for product differentiation, i.e., the same component e of an Embedded System can take different behaviors depending on the actions taken by software inserted.

Studies on the characterization and development of these systems have evolved since the early 80s, starting with military equipment, integrated with the supervisory and process control systems and generalizing in recent years for various applications, [2] as: appliances, PBX systems, automobiles, airplanes, toys, satellites, among others. The increased use of such systems was mainly equipment that perform critical functions in systems that directly affect people's lives, such as software used in aviation, medical equipment and other equipment on which lives depend. A software product is considered critical if it is able to take a critical system to a dangerous condition. [4]

Currently most electronic products has some computational component, where many of these are critical because they are embedded in aircraft, radiation therapy machine, hemodialysis machine, etc., Where upon failure, can cause major disasters. The increased use of these systems, their diversity and the number of functions being incorporated into a single Embedded System, makes it even more critical. [6]

In general, the operation of Embedded System have restrictions because to work must meet certain requirements such as energy consumption, amount of memory, time, size, weight, safety, cost and besides often being exposed to external events. In an Embedded System, compliance with the non-functional requirements is critical to the performance of it is functions (functional requirements) because their effectiveness also depends on the time taken to achieve a result, it is performance, energy consumption, robustness, reliability among other characteristics of an Embedded System, plus the cost for the Embedded System is feasible. Presently with the growing use and complexity of embedded applications, the new features of it is components, in general, are added by software, which means that the variation of the functions performed by an electronic component happen through software inserted.

The software component of an embedded, is gaining more space and importance which increases the demand for solutions and Embedded System [8] services. Studies showed that most of the efforts used in the production of components for Embedded System is directing the development of the software, according to VDC-Accenture [8], 62% of research and development budget and 67% of the cost of a component embedded are used for the development of the software, which indicates their importance in a component. Another significant finding shows that 33% of the produced devices do not meet the requirements of functionality nor performance of the product and 80% of the development effort is spent on correcting errors not identified during earlier phases of their production [8] and 80% of the reasons of failure in embedded systems were caused by problems in the software, not the hardware. [10]

Software Quality is a complex combination of factors that vary with different applications and clients that request is applied throughout the software engineering process, including methods and revisions to assist in constructing a final product.

The development of software to Embedded System components requires the Software Engineering controls include in it is mechanisms allowing to optimize the final product beyond it is development process, to observe in their procedures mechanisms: a) better distribution of functionality between the hardware component and software component depending on their restrictions. The Embedded System in general need to be efficient in their operation where their functions should be carried out precisely and in exact time with the available resources. [7]

The Embedded System in particular, by it is characteristics, the criticality and the high degree of dependence requires special attention. In it is development, should employ activities with defined procedures involving since it is request, understanding, design (selection or development of components), installation, maintenance and use in perfect conditions. The lack of processes and methods of software development with specific guidelines for Embedded Systems software, can derail the prevention of it is defects, especially since the criticality of such systems, we must prevent failures rather than clean up after them, distributing this responsibility for the entire development process

"The defects do not go free. Someone produces them, and get paid to do them. Assuming that costs just as much to correct a defect as costs to do so, it is concluded that 42% of payroll and taxes were being spent to produce and repair defective products". [20]

In general the problems are not in the software itself, but in how they are made, then it is necessary to apply more effectively in the Embedded System industry, the concepts of quality with the same commitment from that applied in other engineering with disciplines, procedures, models and independent life cycles.

II. THE CMMI MODEL

The CMMI model - (Capability Maturity Model Integration) for Development has 5 levels of maturity and enables a growing range of control and visibility over processes and technical and managerial project outcomes software, acts as a reference for obtaining proper standards enabling a common and standardized language is structured with the best (specific and general) practices related to development and system maintenance activities spanning the entire lifecycle of products from conception to delivery and maintenance.

The maturity levels provide a way to control and structure the organization's performance within a given discipline or set of disciplines, are well-defined evolutionary stages where each level provides an important part of the process.

Level 1 - Initial, where processes are informal and success depends on the competence and heroics of the people and not the use of proven processes; Level 2 - Managed - ensure project management, requirements and all it is planning and execution; Level 3 - defined - their processes are well characterized and understood, and are described; Level 4 - Quantitatively Managed - quantitative own for quality and process performance objectives; Level 5 - Optimized - where processes are continuously improved based on an understanding.

Each level consists of the CMMI specific and generic goals and practices organized by process areas. Where a process area is a set of best practices related to a field. Together, these actions meet significant goals.

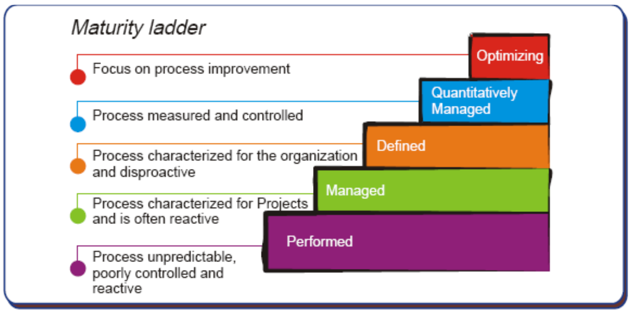

Figure 01 - CMMI – Level [1]

The generic goals are required model components used in general to determine whether a process area has the characteristics necessary to institutionalize the processes that implement are generic in that they apply to various process areas and generic practices are the components to be undertaken and expected to result to the satisfaction of the generic goal.

Specific goals describe the characteristics that must be present for the implementation of a particular process area, is a model required to identify whether a process area is implemented component. The specific practice is the description of an activity considered performed to the satisfaction of the specific goal associated with expected components meet the specific goals of a process area. [12]

III. PROPOSED WORK

An Embedded System is composed of software and hardware component cohabiting component to develop certain activities. Their needs and circumstances are different and independent, so should be designed independently, but by proximity, or even uniqueness of their functions, to achieve the end result, it is development should be done simultaneously and shared way.

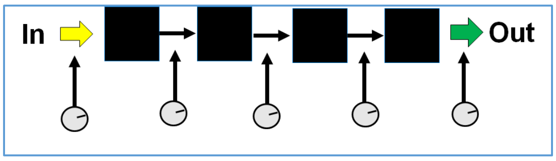

Figure 02 - CMMI Level 2 - Unmanaged [9]

Studying quality models was observed that the application of it is practices in developing a process that meets the product feature of Embedded System, can contribute to the quality of the product to be available on the market, however if not applied properly can derail or impede the progress of their development.

In this work the general practices and specific practices of CMMI Level 2, which has directed it is activities for the project management structuring a process for the development of Embedded System which can ensure the management of the requirements and the project is being implemented, causing development processes can be planned, executed, measured and controlled, and the practices used are likely to be maintained even in times of stress.

Level 2 - Managed CMMI consists of a set of black boxes that represent the stages of their life cycle and their inner workings, must provide adequate infrastructure to support the process, should be planned and implemented in accordance with a policy, and provide adequate resources to produce controlled outputs; should be monitored, controlled and reviewed, and assessed it is compliance in relation to the description of the process. Level 2 can be identified as the scope of a standard to be used.

Figure 02 represents the CMMI Level - 2 model, where do landmarks of work, results and other controls, but does not require the specification of activities for each phase, with the objective planning and management are defined

The Performance Level 3 is characterized as a "defined process" because from the set of processes determined at Level 2, they are described accurately stating clearly the objectives, input, criteria, activities, roles, measures, verification steps and outputs.

The use of a development process Embedded System rated at level 3 maturity generally hinders it is use for different component to be developed since the differences in requirements must be considered.

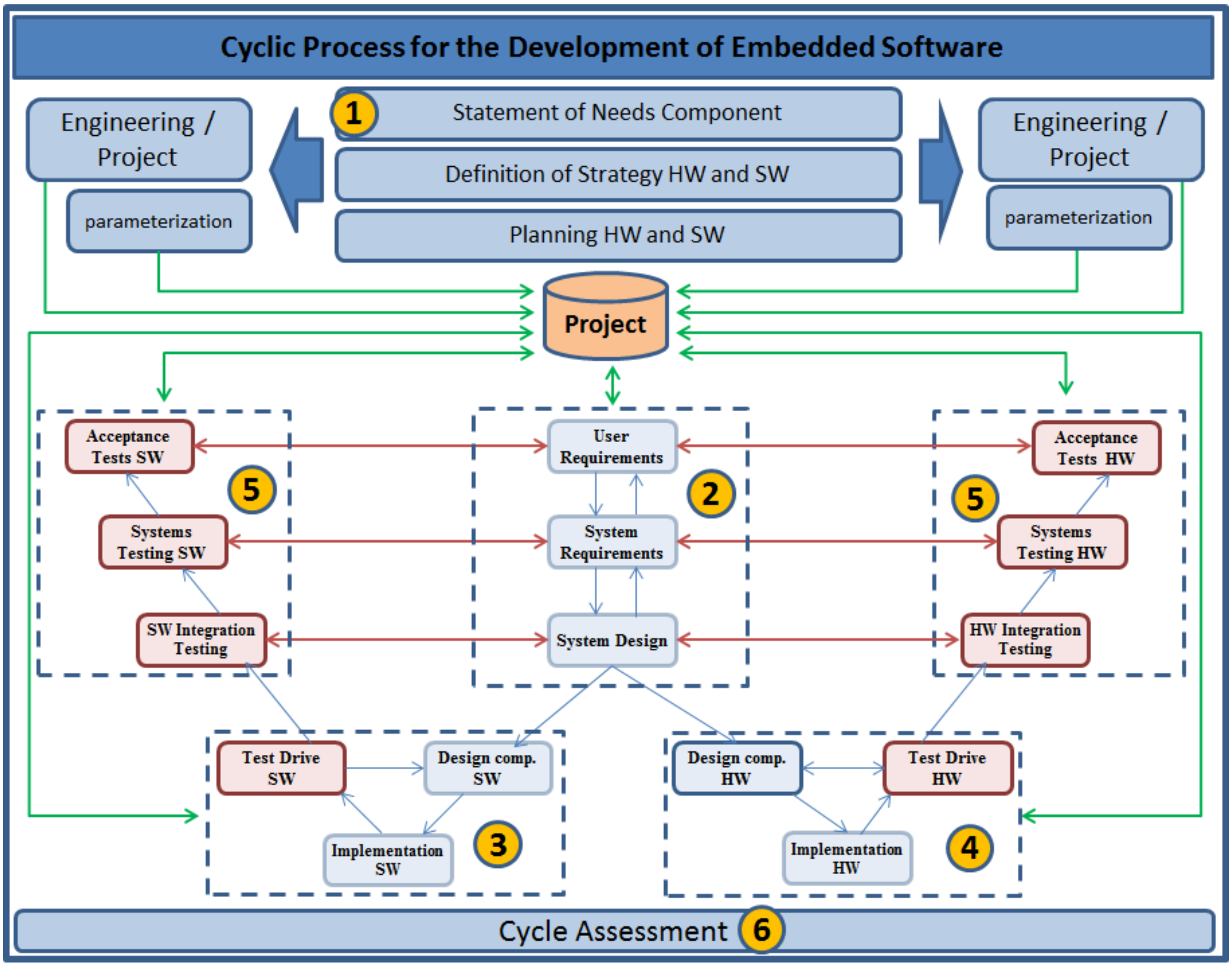

Figure 03 - Process for the proposed Embedded System.

Level 2 of CMMI presents set of practices applied in a process leading developer meet the development standards in stages, however releases details of the development of the standard to be applied.

The generic practices of CMMI model has a fundamental importance in the process, because when applied provides conditions so that it can be established since their activities are focused on the organizational aspects of planning, control, personal commitment, among other activities enabling it is realization. The specific practices of CMMI - DEV V1.3, consist of activities related to each of the process areas that implement it is characteristics and peculiarities.

For the development of Embedded System using a process classified as Level 2 of CMMI, enables the prerequisite for the development of such products as flexible processes classified in the highest levels of CMMI can not be applied to projects by Embedded System particularities involved. The use of such practices, should help in the identification, development and implementation of models for validation of the Embedded System behaviors before implementation, in the early stages, where to find errors are easier and cheaper to be corrected, i.e., eliminate errors design thus minimizing potential problems, losses or catastrophes.

For this study a process for developing Embedded System, organized in stages, allowing for more flexible management of the project and shared between software and hardware using activities that meet best practice models of CMMI Level 2 and Level C MPSBr (Process Improvement of Brazilian Software) in all phases was prepared process.

As presented in figure 03 a process for development of Embedded System must follow a specialized life cycle since it is components develop specialized activities.

The full study is being developed for structuring a development process that performs the Embedded System product designs Embedded System so that at each stage of the process, each activity and each action is the hardware and software components independently performed and shared entity.

3.1. PROCESSO CÍCLICO

The process is divided into phases, and consists of activities carried out through actions governed by procedures and documents (templates) that result in common artifacts that make up the product Embedded System (software and documentation) as shown in figure 03.

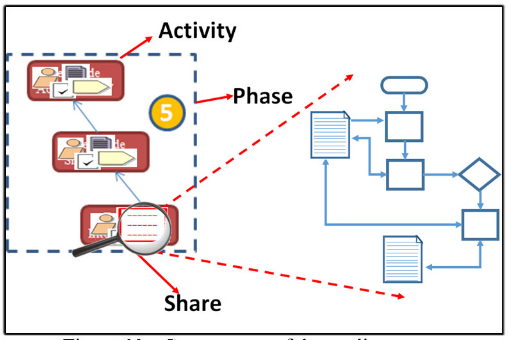

Figure 03 - Components of the cyclic process

The cycle must establish working relationships, define and distribute roles to team members as well as goal setting, strategy and work plan for all phases of the process, and when appropriate modifications of the method according to the evaluation result cycle, as figure 04.

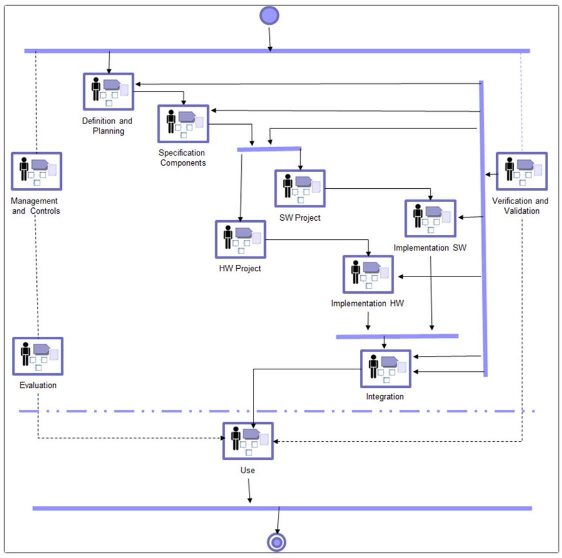

Figure 04 - Process flow of activities

3.1.1. PHASE ENGINEERING - Shall perform general understanding of Embedded System activities as well as it is components, hardware and software, defining functional and non functional requirements, definition of control items, configuration and traceability, test planning, establishing metrics and monitoring and approval (item 1 of figure 02).

3.1.2. PHASE: PRODUCT DETAILS – Implementation of joint activities for hardware and software, detailing requirements and restrictions of product identification and detailing of (functional and non-functional) requirements of the components of hardware and software components independent, detailing the items control and monitoring, identification of items and components for reuse, project design (coprojeto) Embedded System, an integration mechanism, organization of the test model and homologation requirements (item 2 of figure 02).

3.1.3. PHASE: MODELING AND DESIGN – Study design and specification of component software and hardware components. This stage should be elaborate models, prototypes, designs architectures to identify the best solution for Embedded System as well as identification and distribution of tasks to be performed by the hardware and software in finding the best solution (ideal architecture).

3.1.3.1. PHASE: DESIGN SOFTWARE – Study and detailing the software component of the power control items, configuration, tracking and metrics, construction and implementation of software testing, monitoring and approval of the draft software. (Item 3 in figure 02).

3.1.3.2. PHASE: HARDWARE DESIGN – Study and detailing the hardware component, study and approval of reuse items, feed items control, configuration, tracking and metrics, construction and implementation of hardware testing, monitoring and approval of the draft hardware. (Item 4 in figure 02).

3.1.4. PHASE: VERIFICATION AND VALIDATION – Integration activities of the components of software and hardware, realization of integration of component tests, system tests of hardware, software part test systems, test Embedded System, acceptance testing of the product, supply of items of control, configuration, and traceability metrics, monitoring and approval draft Embedded System. (Item 5 of figure 02).

3.1.5. PHASE: CYCLE ASSESSMENT - Analysis of activities in the cycle, adjustments and adaptation for the next cycle (item 6 in figure 02).

4. EXPECTED RESULTS AND CONTRIBUTIONS.

Historically the activities of software engineering are directed to obtaining mechanisms for management and control processes for major developments, and thus improve the quality, productivity and performance of their processes get their current needs are also demanding a look at software with special features such as Embedded Systems. This work seeks to contribute to the community by presenting an organization's software development process, and a commitment to quality of the process, also allows using the resources of software engineering, assist in performing activities that are specific to embedded systems and thus contribute to increasing the quality of such systems, using consistent procedures, activities and document templates that induce developers to use good software engineering practices.

The dissociation of the activities of it is flow enables targeted to the characteristics of it is various components (hardware and software) simultaneously and shared way procedures.

The present study and the detailed specifications of the process may stimulate further research.

REFERENCES

[1] http://www.qaiglobalservices.com/Minisites/High_Maturity_Leadership/index.asp em 30/03/2014.

[2] SPINOLA, M. M.; PESSÔA, M. S. P. Produtos que embutem software: aspectos relevantes para a melhoria do processo de desenvolvimento. In: Simpósio Brasileiro de Engenharia de Software, 11. Wqs´97 – Workshop Qualidade de Software, Fortaleza, 14/10/1997. Anais. Fortaleza, Universidade Federal do Ceará, 1997. p.7-14.

[3] DEMING, William Edwards. Qualidade: a revolução da administração. Trad. Clave

[4] Comunicações. Rio de Janeiro: Marques-Saraiva, 1990.

[5] PÁSCOA, João, E, P., Fatores e Subfatores para Avaliação da Segurança em Software de Sistemas Críticos Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia - 2002 São Paulo, SP.

[6] Barroso, M. A; Um Processo de Desenvolvimento para Sistemas Computacionais Aderente ao MPS.BR Nível G - Universidade de Fortaleza - Centro de Ciências Tecnológicas - Mestrado em Informática Aplicada – Fortaleza – 2010

[7] Wehrmeister.M.A., Leandro Buss Becker L.B. Pereira C.E, Metodologia de Projeto Orientada a Objetos Baseada em Plataformas para Sistemas Tempo-Real Embarcados – SBRC2007 – Belem do Pará – Brasil - 2007

[8] F. Rech Wagner, “Minicurso Engenharia de Software para Sistemas Embarcados” - III Simpósio Brasileiro de Engenharia de Sistemas Computacionais - SBESC/2013 – UFRJ.

[9] Accenture - Embedded Software for the Electronics and High-Tech Industry - http://www.accenture.com/SiteCollectionDocuments/PDF/EmbeddedSoftware2.pdf em 13/02/2014

[10] MIYASHIRO, M. A. S., Identificação e melhoria do nível de maturidade de uma organização explorando técnicas de inteligência computacional. 2007. 100 f. Dissertação (Mestrado) - Instituto Nacional de Pesquisas Espaciais, São José Dos Campos, 2007

[11] MARWEDEL, P. Metodologia de Projeto de Software Embarcado Voltada ao Teste - Dissertação de Mestrado Universidade Federal do Rio Grande do Sul – 2010.

[12] SOFTWARE ENGINEERING INSTITUTE. - CMMI for Development, Version 1.3 CMMI-DEV, V1.3 - CMMI Product Team – Improving processes for developing better products and services - Technical Report - Pittsburgh: Carnegie Mellon University, November 2010.