The Practices of Level 2 (Managed) CMMI-DEV V1.3 in Development of Embedded Systems

Magda A. Silvério Miyashiro[1] and Maurício G. V. Ferreira[2]

[1]642 Tokuchika Miki Street, Zip Code: 03211-010 - São Paulo - SP, Brasil

[2]1758 Astronautas Street, Zip Code: 12227-010 - São José dos Campos - SP, Brasil

Keywords: Embedded System, Hardware, Software, Software Quality, CMMI, Maturity, Critical Software, Process, Generic Practices, Specific Practices**.**

Abstract. The constant evolution of the electrical and electronics industry, use of its components in products from different areas of the industry has contributed provision of smart products and automated eat and which help us to conduct our activities of the day, from household appliances, toys, medical equipment, traffic among others, are embedded in our society so strongly that often without notice its existence. Most of these devices make use of computer embedded systems, which are components of the systems which are produced with a combination of hardware and software capability to perform extremely important tasks using little computational resources, which most often possess restrictions for their use and for their development because they require specific requirements in accordance with the functionality to be realized. It is use generally adds certain values that are currently indispensable to products that use these components. The use of such systems allows greater flexibility and intelligence to products where they are inserted. The importance of embedded systems in general , it intensifies as embedded in aircraft, radiation therapy machine, space systems among others are considered as critical, where upon failure, can cause major disasters. The dependence of society, different applications and technological developments are delegating all flexibility of these systems for your software component, which makes it even more critical thus requiring appropriate and specific procedures in their production. To ensure its functioning in it is development, should employ processes with defined procedures that meet their particularities, where mistakes can be eliminated or minimized through the activities in all stages of development and not leave to correct them when found. The use of specific and generic practices of CMMI Level 2 - DEV V1.3 model assists in developing a process for managing development that meets the characteristics of embedded system, where its phases and activities carry the good practice suggested by the model and as a result produce templates and documents for seamless integration and operation of software and hardware components of embedded system.

INTRODUCTION

In the 60s, with the growth of computer-based systems, also increased the complexity of software development, which gave rise to the so-called "Software Crisis". Fifty years after this crisis was large developments in the area of software development, 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 (HW) and software (SW) dedicated to a specific application. In general, are parts of a larger system and react to it is stimuli [5].

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, making then part of our daily lives [2] as: appliances, PBX systems, automobiles, airplanes, toys, satellites, among others, which implies

increasing and differentiated demands, expanding, generally, difficulties in developing systems embedded both in quantity and importance [2].
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.
In general, the operation of SE 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. The increased use of these systems, their diversity and the number of functions being incorporated into a single SE, makes it even more critical [8].
Originally electronics were structures of an electronic circuit, with the evolution of the SW and to decrease the cost of production is passed using SW as part of its components which may be responsible for product differentiation , ie, the same component e of an SE can take different behaviors depending on the actions taken by SW inserted. In an SE, compliance with the non-functional requirements is critical to the performance of its functions (functional requirements) because their effectiveness also depends on the time taken to achieve a result, its performance, energy consumption, robustness, reliability and other characteristics of an SE, plus the cost for the SE is feasible. Presently with the growing use and complexity of embedded applications, the new features of its components, in general, are added by SW, which means that the variation of the functions performed by an electronic component happen through SW inserted.
The SW component of an embedded, is gaining more space and importance which increases the demand for solutions and SE [11]. Studies showed that most of the efforts used in the production of components for SE is directing the development of the SW, according to VDC- Accenture [11], 62% of research and development budget and 67% of the cost of a component embedded are used for the development of the SW, 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 [11] and 80% of the reasons of failure in embedded systems were caused by problems in the SW and HW not in [16].

QUALITY SOFTWARE EMBEDDED SYSTEMS

The SE 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 [10].
The development of SW to SE components requires the Software Engineering controls include in its mechanisms allowing to optimize the final product beyond its development process, to observe in their procedures mechanisms: a) better distribution of functionality between the HW component and SW component according to its constraints, b) identify errors in projects prior to implementation, c) facilitate the reuse of components, d) use of metrics product metrics beyond the project, e) use of behavioral models, f) increased productivity SE, among other practices that can directly influence the quality of SE.
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 SE in particular, by its characteristics, the criticality and the high degree of dependence requires special attention. In its development, should employ activities with defined procedures involving since its 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 SW and SE, 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.
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 SE industry, the concepts of quality with the same commitment from that applied in other engineering with disciplines, procedures, models and independent life cycles.

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 SW, 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. Their efforts are focused in three dimensions deemed critical in a organization identified by the SEI (Software Engineering Institute) who are people, procedures and methods, and tools and equipment [18], as shown in figure 1.

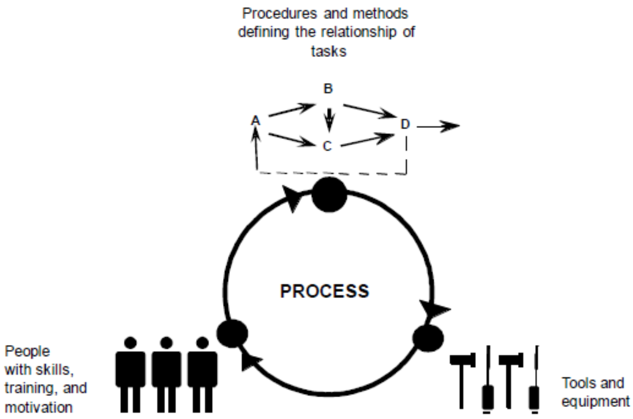

Figure 1[18]

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 its 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.
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.

PROPOSED WORK

One can understand that a minor component embedded software still is a system, since it is composed of SW and HW cohabiting to develop certain activities. Their needs and circumstances should be designed independently, since their characteristics are different conceptually being the HW and SW is physical is logical, but by proximity, or even uniqueness of their functions to achieve the final result its development should be done simultaneously and seamlessly.

Studying quality models was observed that the application of its practices in developing a process that meets the product feature of SE, can contribute to the quality of the product to be available on the market.

In this work the general practices and specific practices of CMMI Level 2, which has directed its activities for the project management structuring a process for the development of SE which can ensure the management of requirements , processes are planned, executed is being implemented measured and controlled , and that existing practices are retained during times of stress.

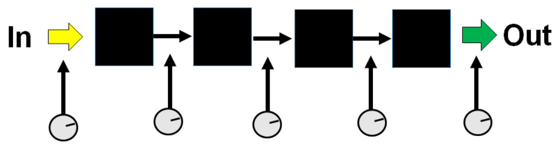

Figure 2 – Level 2 CMMI – Managed [13]

Figure 2 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 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 practices are focused on the organizational aspects of planning, control, personal commitment, among other activities enabling its realization.

Table 1 - Process Areas of CMMI Level 2.

Cod	Generic Practices
GP 2.1	Establish and maintain an organizational policy for planning and performing the process.
GP 2.2	Establish and maintain the plan for performing the process.
GP 2.3	Provide adequate resources for performing the process, developing the work products, and providing the services of the process.
GP 2.4	Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process.
GP 2.5	Train the people performing or supporting the process as needed.
GP 2.6	Place selected work products of the process under appropriate levels of control.
GP 2.7	Identify and involve the relevant stakeholders of the process as planned.
GP 2.8	Monitor and control the process against the plan for performing the process and take appropriate corrective action.
GP 2.9	Objectively evaluate adherence of the process and selected work products against the process description, standards, and procedures, and address noncompliance.
GP 2.10	Review the activities, status, and results of the process with higher level management and resolve issues.

The specific practices of CMMI-DEV V1.3, consist of activities related to each of the process areas to implement its features and peculiarities.

Table 2 - Process Areas of CMMI Level 2.

Area Process	Target Area
Requirements Management	Provide subsidies to manage product requirements and product components of the project and ensure alignment between those requirements and the plans and work products of the project.
Project Planning	Establish and maintain plans aiming to define project activities.
Monitoring and Control Project	Provide subsidies to provide visibility of project progress so that appropriate corrective actions can be implemented when the project's performance deviates significantly from the plan.
Management Supplier Agreement	Provide subsidies to manage the acquisition of products and services from suppliers.
Measurement and analysis	Provide grants to develop and maintain a measurement capability used to support the information needs of management.
Quality Assurance Process and Product	Provide visibility to the staff and management of processes and associated work products.
Configuration Management	Provide grants to establish and maintain the integrity of work products using configuration identification, configuration control, configuration balance of activities and configuration audits.

For the development of SE 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 SE particularities involved . The use of such practices, should help in the identification, development and implementation of models for validation of the SE behaviors before implementation, in the early stages, where to find errors are easier and cheaper to be corrected, ie, eliminate errors design thus minimizing potential problems, losses or catastrophes.

For this study a process for developing SE, organized in stages, allowing for more flexible management of the project and shared between SW and HW using activities that meet best practice models of CMMI Level 2 and Level C MPSBr in all phases was prepared process.

As presented in figure 3 a process for development of SE must follow a specialized life cycle since its components develop specialized activities.

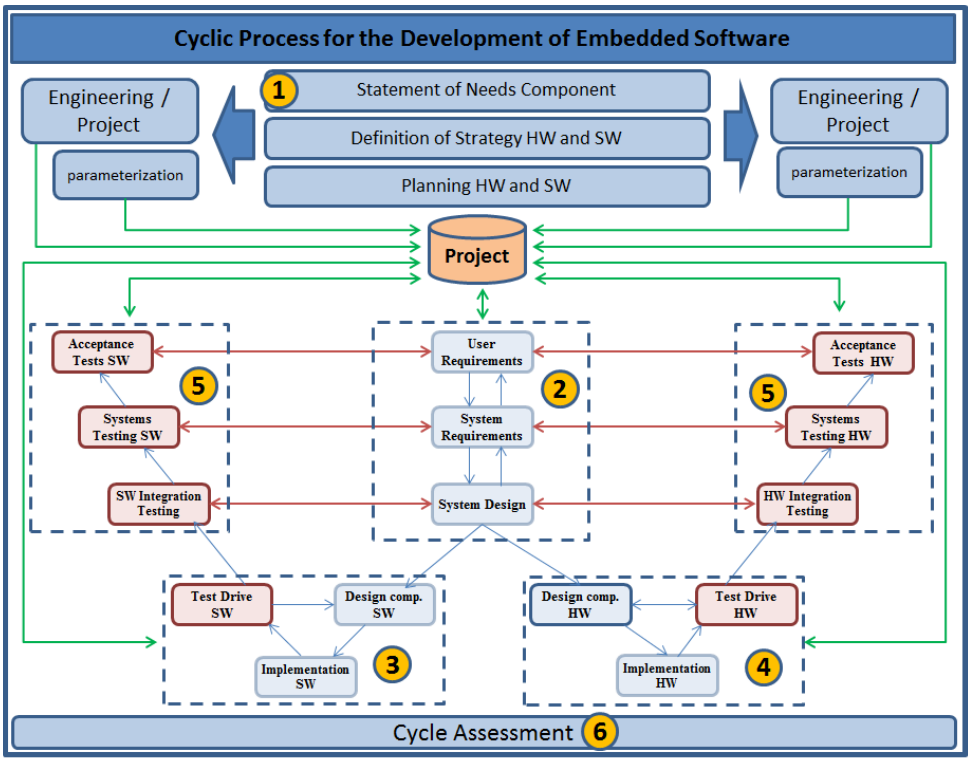

Figure 3 – Processo proposed for SE

The full study is being developed for structuring a development process that performs the SE product designs SE so that at each stage of the process, each activity and each action is the HW and SW components independently performed and shared one.

- CYCLIC PROCESS

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 SE (software and documentation) as shown in figure 4.

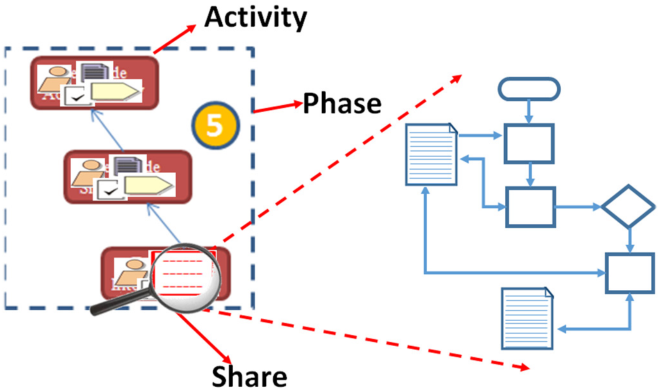

Figure 4 – Components of the cyclic process

The concept of cycle was reasoned from the adaptation of the concept proposed by Humphrey in TSP (Team Software Process) [3]. Should 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.

- PHASE ENGINEERING - should perform general understanding of SE 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 mechanisms monitoring and approval (Item 1 in Figure 3).
- PHASE PRODUCT DETAILS - Conducting joint activities for HW and SW, detailing requirements and restrictions of product identification and detailing of (functional and non-functional) requirements of the components of HW and SW components independent, detailing the control items and monitoring, identification of items and components for reuse, project design (co-design) SE, an integration mechanism, organization of the test model and homologation requirements (Item 2 in Figure 3).
This stage should be elaborate models, prototypes, designs architectures to identify the best solution for SE as well as identification and distribution of tasks to be performed by the HW and SW in finding the best solution (ideal architecture).
- PHASE DESIGN SOFTWARE - Drafting of the study and detailing the SW component of the power control items, configuration, tracking and metrics, construction and implementation of SW testing, monitoring and approval of the draft SW. (Item 3 in Figure 3).
- PHASE HARDWARE DESIGN - Preparation of the study and detailing the HW component, study and approval of reuse items, feed items control, configuration, tracking and metrics, construction and implementation of HW testing, monitoring and approval of the draft HW. (Item 4 in Figure 3).
- PHASE VERIFICATION AND VALIDATION - Activities integration of SW components and HE , realization of integration of component tests , system tests of HW , SW part test systems , test SE, acceptance testing of the product nutrition items control, configuration, tracking and metrics , monitoring and project approval SE. (Item 5 in Figure 3).
- PHASE CYCLE ASSESSMENT - Analysis of Activities in the cycle, adjustments and adaptation for the next cycle. (Item 6 in Figure 3).

EXPECTED RESULTS AND CONTRIBUTIONS

This paper seeks to contribute to increasing the quality of SE with consistent procedures, activities and document templates. The breakdown of your activity stream enables targeted to the characteristics of its various components (HW and SW) simultaneously and shared way procedures. The present study and the detailed specifications of the process may stimulate new research.

REFERENCE

[1] T.Y. Yen, W.Wolf, Hardware-software co-synthesis of distributed embedded systems. Kluwer, 1996.

[2] M.M. Spinola, M.S.P. Pessôa, 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] M. Barr, Programming Embedded Systems in C and C++. O'Reilly & Associates, Inc., 1999.

[4] J.E.P. Páscoa, 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.

[5] M.A. Barroso, 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.

[6] I. Crnkovic, Component-based approach for embedded systems, Mälardalen University, Department of Computer Science and Engineering Sweden. Apude [D].

[7] A Comparative Study of Software Component Models for Embedded Real Time Systems. www.idt.mdh.se/utbildning/exjobb/files/TR0582.doc acessado em 13/02/2014.

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

[9] M.N.S. Kondo, J.R. Silva, A.Y. Hira, M.K. Zuffo - Estudo de Requisitos do Software Embarcado no Segmento da Telemedicina – Congresso Brasileiro de Informática em Saúde – Brasil.

[10] ES p/ Sistemas embarcados Flavio – EFRJ

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

[13] M.A.S. Miyashiro, 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.

[14] J.P.M. Reginato, Uma Proposta de Aperfeiçoamento de um Processo de Gerenciamento de Requisitos de Sistema e de Software e Sua Aplicação a Sistemas Espaciais e Aeronáuticos Embarcados - Dissertação de Mestrado – Instituto Nacional de Pesquisas Espaciais, São José Dos Campos, Brasil – 2012.

[15] GU2009 A.M. Guerra, R.M.T. Colombo, Tecnologia da informação-Qualidade de Produto de software. PBQP Software- Jun-2009.

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

[17] J.C. Ossada; L.E. Martins GERSE: Guia para Elicitação de Requisitos de Sistemas Embarcados I Workshop de Sistemas Embarcados Gramado, RS, Brasil – 2010.

[18] 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.