

Customization Process of the Process for the Development of Embedded Components for the Aerospace Industry

Magda A.S. Miyashiro¹(✉), Maurício G.V. Ferreirao¹, Mauro M. Spínola²,
Marcelo S.P. Pessoa², and Rodrigo Franco Gonçalves²

¹ National Institute for Space Research, São José dos Campos, Brazil
magda.silverio@ globo.com

² University of São Paulo, São Paulo, Brazil

Abstract. In the aerospace industry, the number of products using embedded components is increasing. Nevertheless, there are still processes to ensure that the embedded components can be constructed without failure. Therefore it is necessary to identify and establish process of adaptation to achieve the product quality through the quality of its production process, preventing flaws the process rather than fixing them in the final product actions to be considered in the customization process. From a set of factors obtained from the literature, they were listed CMMI elements and embedded system characteristics for definition of the criteria used for Aerospace Industry. It was obtained a sequence of actions to be taken to adapt any process to meet the particularities of its component products. The use of a static process could not meet the characteristics of each component, but you can set an adaptive process from a default and thus meet their individualities.

Keywords: Adaptation · Aerospace · CMMI · Embedded components · Customization · Quality factors · Software process

1 Introduction

An embedded component is an installed device in a product, it is generally dedicated to performing a set of predefined tasks with specific requirements. They are formed by hardware and software that are closely related, where the hardware is the physical part and the software component is the computing part, namely, the logic of which is shown in Fig. 1.

With the development and evolution of technology, embedded systems have also become increasingly sophisticated and complex, which directly influences their development process, making it a costly and prone to errors activity [2].

Several accidents show that embedded components designed to perform critical systems often fail, as in the cases of rocket Ariane 5 [2], which exploded 37 seconds after launching, and Therac-25 [3], radiation therapy machine, which made six fatalities, among others.

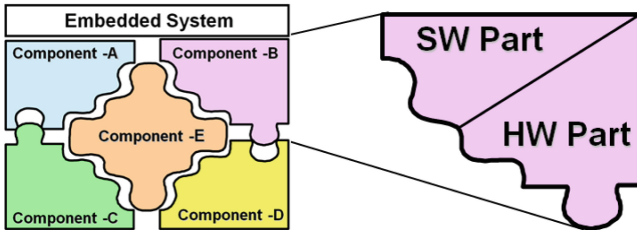


Fig. 1. Representation of parts of a system (Source: adapted [1])

A sad conclusion presented by researcher Lim Joanne on one of the most significant examples about software failure in embedded components, demonstrates the emergency of continuous in-depth studies to ensure the quality of these products: *“In many senses, a computer had killed their mother, father, son, daughter, or whoever was closest to their heart. It is hard to believe that through simple human error, an unnecessary loss of lives could take place. It has happened, and may happen again in the future. Therefore, it must be our goal to never let such a tragedy as Therac-25 take place again”*.

The aerospace industry demands highly specialized and customized components. The increased use of these systems, their diversity and the number of functions that are being incorporated into a single embedded system multiplies the degree of criticality [4], especially on devices that perform critical functions in systems that directly affect people’s lives. A software product is considered critical when, in case of failure, takes a system into a dangerous state, that is, when the software can cause disasters or fatalities [5].

The software quality theme, whether in academia or in companies, reports the various models and methodologies to be applied in the development process. However, when it comes to embedded component, which means developing software and hardware together, the question is: how to integrate these two realities ensuring quality? [6].

A study developed by Carnegie Mellon University’s Software Engineering Institute (CMU/SEI) on industry trends in the use of software components pointed out, among other weaknesses, that software engineering lacks methods to produce quality systems [7]. Current researches show great need of processes, methods, techniques and tools to assess the quality of software components, even more specifically for embedded software [1,2].

CMMI V1.3 (Capability Maturity Model Integration) was developed by the Software Engineering Institute - SEI using total quality concepts introduced by Humphrey (2000) and his team. CMMI is based on the concept of maturity of software processes and is compatible with ISO/IEC 15505e ISO/IEC 12207 standards. It has five levels of maturity in a growing range of control and visibility that allows the classification of processes, technical results and software project managements [1].

For this study, we considered a number of factors obtained from the literature, elements of CMMI, ISO standards and quality factors, and characteristics of embedded systems in general were listed, in order to set a definition for the criteria used for aerospace industry based on a specific case, as Fig. 2. The following activities were carried out: problem identification to be searched, literature revision, definition of the scope of the search, study of quality models and proposal in line with the research model presented by Creswell [8].

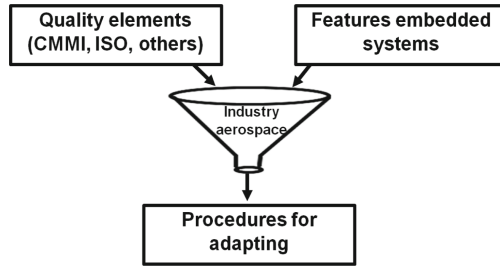


Fig. 2. Search methodology

This paper presents criteria for adaptation of a process for the development of embedded components following the CMMI (Capability Maturity Model Integrated), being part of a study being carried out to provide the adaptation criteria with different arguments. It is being held at the Postgraduate Course in Space Engineering and Te - National Institute for Space Research in continuing the research work in doctoral technology, specialization in Engineering and Space Systems Management at INPEId at the same institution [1].

2 Current Situation of Embedded Components

There are considerable references to embedded systems in literature. However, few of them direct their researches or activities considering only the software components independently. Therefore, it is necessary to use a software development approach that takes into account the characteristics and peculiarities of this context.

A survey conducted by Accenture shows that this increase in the embedded component usage proportionally increases the demand for solutions and services that meet such systems [9]. Most of the effort involved in producing components for embedded systems is directed to the development of software part of the component, being 62% of the budget for research and development and 67% of the cost [9]. It was also observed in this research that 33% of the produced devices do not meet the requirements of functionality or product performance, and that 80% of the development effort is spent on correcting unidentified errors during the earlier stages of production [9]. In addition, 80% of the reasons for failure in embedded systems come from problems in the software, not the hardware [10].

For being older and having more consolidated processes, the hardware industry (manufacturing) has had for a long time a higher level of maturity for the development of its products, while software production, despite its progress, is still at the beginning of development, and needs more studies and applications in order to accomplish this maturity [1]. This means that the stability of the hardware is not enough to ensure the quality of the system as a whole, since the development of software components of the product may present weaknesses that can cause execution failures.

The problems caused by the development of complex software show us that even with all the efforts made, the software can also be built with a few flaws that compromise their performance. This can be seen by the high number of defects presented by such programs [7].

3 Cyclic Process: A Model

The development of design of an embedded component system must follow a specialized life cycle, since these components perform specialized activities. For this, quality models were examined, especially the CMMI-DEV, and it was found out that the application of its recommendations in the preparation of a development process that suits the characteristics of the embedded system component can contribute to increase the quality of the product to be designed.

The “Cyclic Process” is being used in this study, and it was organized in phases where its activities are designed to be applied in an independent form and shared between the software and hardware parts of the embedded component. The process is called cyclic because each embedded system component to be developed according to their specifications must perform all phases and process activities, thus forming a complete cycle of development of components for embedded system. This operation is shown in Fig. 3.

Development cycle is the linear execution of all phases and activities of the life cycle, where each development cycle will result in a component and all documents are produced during its development. In general, an embedded system consists of several components, then for each component individually, it should perform different development cycles.

The phases of the “Cyclic Process” consist of activities and are implemented through actions directed by procedures and documents (models) that result in shared artifacts, which make up the embedded component (programs and their documentation), as follows: (a) Common activities, (b) Engineering, (c) Analysis of requirements, (d) Analysis and design, (e) Implementation and integration, (f) verification and validation of the system and (g) cycle assessment [1]. As an example, Fig. 4 shows the Analysis and Design phase.

Even the “Cyclic process” particularly designed for the development of embedded components, it should allow the adaptation of activities to be undertaken in the development of each developed component. The “Cyclic Process” induces the adjustment of its activities for each component to be produced, but does not provide criteria or guidelines to be followed while carrying out such adaptations.

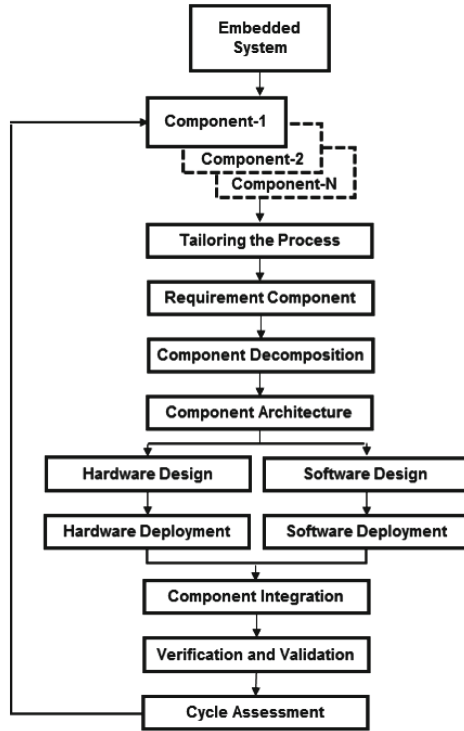


Fig. 3. Operation of the “Process Cyclic” (Source: [1])

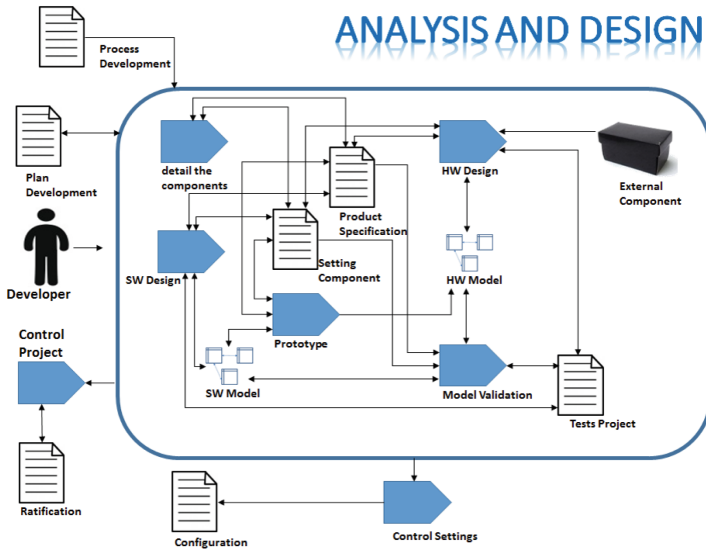


Fig. 4. Phase analysis and design (Source: [1])

4 Criteria to Be Considered in the Adaptation by CMMI-DEV

In general, the whole development process that meets the CMMI model, follows standards, procedures, tools and methods that should be employed in their preparation, and their adjustment is given as a set of standard processes of the organization, according to its guidelines for adaptation [1]. The area in the Organization Processes Focus (OPF) directs the development of the standard process following the organizational needs, while the process area Definition of the Organization’s Processes (OPD) has fundamentals that must be followed during the process of adaptation, so that it maintains the basic requirements of CMMI model. Figure 5 shows the structure of adaptation where the standard process was drawn from the OPF process area and the requirements for each product to be produced it is defined an adapted process, following the process area of requirements OPD.

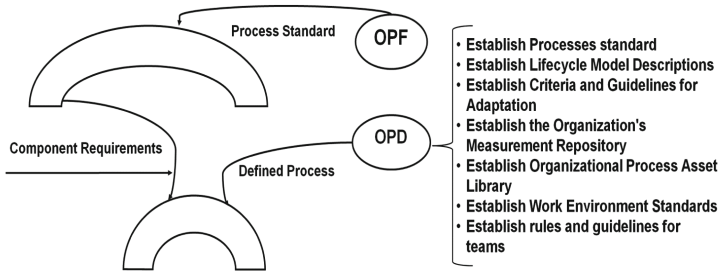


Fig. 5. Adaptation of “Cyclic process according to CMMI”

Every software component, especially embedded ones, require mechanisms to provide the monitoring service of their requirements during production. Therefore, it is necessary to know the component requirements and parameters that define its full service by performing identification and adjustments through inspection of the aspects of the process that might not meet the acceptable limits of the required result for the component to be produced. These mechanisms should be the result of an elaborate process, carefully adapted and documented. Generically, these activities are shown in Fig. 6. As a result of the implementation of the activities to the “Cyclic Process”, the following adjustments were made:

- Abstraction of requirements - This activity requires the understanding and details of the component requirements, and must be registered in the “Development Plan” document.
- Identification of the factors of the requirements that must be observed - This activity should be performed based on the records held in the “Development

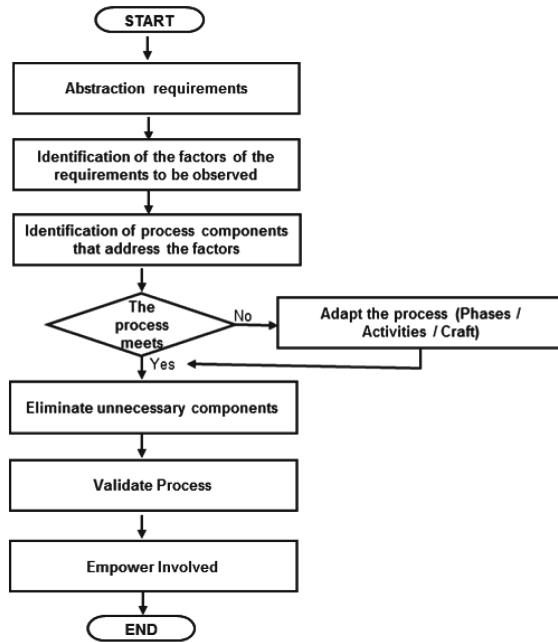


Fig. 6. Activities adaptation

Plan”. Based on the characteristics and component requirements and on quality standards or models, it should specify which factors and care are the appropriate criteria for the component needs, and it must be registered in the “Development Plan”.

- Identification of process components that address the factors - for this activity, it should identify items which should describe the actions that meet the component requirements in the activities of the lifecycle process and its artifacts.
- Adapt the process (Phases/Activities/Artifacts) - This activity should carry out the adjustments in the documents and artifacts of the process: “Development Plan”, “Product Specification”, “Component Specification”, “Test Project” “Settings” and “Approval”.
- Eliminate unnecessary components - this activity should be carried out to evaluate the life cycle and process artifacts, eliminating unnecessary items from the development of the component. As a result, it may cause adjustments in the documents and artifacts of the process, “Development Plan”, “Product Specification”, “Component Specification”, “Test Project”, “Settings” and “Approval”.
- Validate process - this activity is necessary to evaluate the adapted process and to validate the compliance regarding the component to be produced. The result of this action must be registered in the artifact “approval”.
- Training the involved ones - Every process must be known by those who use it. Then, any adjustment that might occur should be informed and the involved

ones, be trained to use. The training plan, when necessary, should be recorded in the artifact “Development Plan”.

The documents “Development Plan”, “Product Specification”, “Component Specification”, “Test Project”, “Settings” and “Approval are part of the “Cyclic process” and are published in the thesis “An approach to the process of embedded system development that meets the level 2 CMMI-DEV maturity” [1].

For the adaptation process, it is necessary to take into account the environment in which it is inserted (the system and its requirements) as part of requirement, in addition to the component requirements themselves.

As a result, this paper presents the basic activities and criteria to be considered for adaptation of a process for the development of embedded components with CMMI-DEV’s vision.

5 Conclusions

It is clear that the use of a development process that meets the characteristics of each embedded component, in particular when performing the treatment of software and hardware parts differently and separately, contributes significantly to improve the quality of the produced component, as in the “Cyclic Process”. However, the same process, in a static way, is not enough to meet the particularities of different components to be produced. This requires adaptations that meet their development and operation constraints and should also take into account the special features of the component to be developed.

References

1. Marwedel, P.: Uma Abordagem para o Processo de Desenvolvimento de Sistema Embarcado que Atende ao Nível 2 de Maturidade do CMMI-DEV. Ph.D. thesis, Instituto Nacional de Pesquisas Espaciais (2015)
2. Sommerville, I., Melnikoff, S.S.S., Arakaki, R., de Andrade Barbosa, E.: Engenharia de Software, vol. 6. Addison Wesley, São Paulo (2003)
3. Lim, J.: An Engineering Disaster: Therac-25. Technical report (1998)
4. Wehrmeister, M.A., Becker, L.B., Pereira, C.E.: Metodologia de Projeto Orientada a Objetos Baseada em Plataformas para Sistemas Tempo-Real Embarcados. In: 25° Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Belém do Pará (2006)
5. Páscoa, J.: Fatores e Subfatores para Avaliação da Segurança em Software de Sistemas Críticos. Ph.D. thesis, Universidade de São Paulo (2002)
6. Barroso, M.: Um Processo de Desenvolvimento para Sistemas Computacionais Aderente ao MPS. Ph.D. thesis, Universidade de Fortaleza (2010)
7. Spinola, M.M.: Diretrizes para o Desenvolvimento de Software de Sistemas Embarcados. Ph.D. thesis, Universidade de São Paulo (1999)
8. Creswell, J.W.: Educational Research: Planning, Conducting, and Evaluating Quantitative. Prentice-Hall, Upper Saddle River (2002)
9. Accenture. www.accenture.com
10. Gomes, H.: Metodologia de Projeto de Software Embarcado Voltada ao Teste. Ph.D. thesis, Mestrado Universidade Federal do Rio Grande do Sul (2010)