# Process for the development of Embedded System following the practices of CMMI Level 2

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

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 there is a great dependence on industry and society to the use of embedded computer systems, they are present in our daily lives and often we do not realize it. The greatest feature of such systems is the ability to perform very important tasks with little computational resource. Many of these products are considered critical, since in case of failure, may cause catastrophes or large losses when embedded in aircraft, radiation therapy machine, space systems among others. In most cases, embedded systems have restrictions on their development and operation, they require specific requirements, in addition to not having models of the systems development process that meets these needs. The high demand and different applications are making it even more critical. To ensure it is functioning in it is development, should employ processes with defined procedures for their particularities. This article presents a study that is being conducted to define the structure of the development process that meets the characteristics of embedded system, where it is phases and activities induce the good practice suggested by current models of quality, and produce documents that meet these models and allow control points for the seamless integration of the functions of the software and hardware of the embedded system.**

***Keywords— Embedded System; Hardware; Software; Software Quality; CMMI; Maturity; MPS.Br; Critical Software; Process***

## I. INTRODUCTION

Embedded Systems are electronic artifacts embedded in most products we use targeting it is operation and control. Currently most electronic products has some computational component, where many of these are part of products that are considered critical when loaded on planes, radiotherapy machine, hemodialysis machine, etc. where upon failure, can cause large disasters.

In the current literature has not found a consensus for a formal definition of Embedded Systems [1] [2] however, in general, Embedded Systems is a combination of hardware and software closely related, designed to perform a function specific [3], [4], [5], i.e. the Embedded Systems are an integrated part of a larger system and performs some of the requirements of this system computer eg., a computer system used in a aircraft or rapid transit system. [7] apud [6] [5].A strong feature of Embedded Systems is having restrictions on their development and operation, to operate it 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 Embedded Systems is able to perform very important tasks featuring little computational resource. The design of such systems starts with a specification that generally implies establishing components of hardware and software as well as your specific requirements. Based on system specifications, the features required are mapped into concurrent tasks to be implemented in their areas of hardware or software implementation. In developing if there is a high degree of difficulty, which starts from understanding and estimation of need, development of the project to the specification and implementation of the product [9]. 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; [8].

## II. EVOLUTION EMBEDDED SYSTEMS

The Embedded Systems adds a significant amount of value to the product, allowing for greater flexibility and intelligence to it is components then making their use a great solution trends in different market.

Initially the electronics were structures of an electronic circuit, that is, components that are part of any electrical or electronic circuit interconnected to transmit electrical current. With the evolution of the software and to decrease the cost of production is passed using software as part of it is components being responsible for this product differentiation, i.e., the same electronic component of an Embedded Systems can take different behaviors depending on the shares held by software inserted.

Functional requirements (features) of the Embedded Systems are essential for it is operation and should be treated with caution, however, in it is development should be given great importance also to non-functional requirements that can call requirements are met. Compliance with such requirements is critical to the performance of it is functions because it is effectiveness also depends on the time taken to achieve a result, it is performance, energy consumption, robustness, reliability and other characteristics, plus the cost for Embedded Systems is feasible. Presently with the growing use and complexity of embedded applications, their new features are usually added by software, which means that the variation of the functions performed by an electronic component happen through software inserted. The dependence of the use of these

technologies by society began to demand increasingly sophisticated products and an increasingly low [10], thus increasing productivity and quality of an Embedded Systems has become an essential task.

The software component of an embedded, is gaining greater importance where space and a greater proportion increases demand for solutions and Embedded Systems [11]. Studies showed that most of the efforts used in the production of components for Embedded Systems are directing the development of the software, according to Accenture [11], 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.[11]

## III. Engineering Software Embedded Systems

The conventional software engineering in general aims to improve the productivity of the process of projects considering the reuse of components and techniques for the abstraction of their features and their functional constraints beyond the control and management of their development, i.e. directs it is efforts to the way in which the product must be built to achieve higher productivity, however neglects certain requirements of the final product. The Embedded Systems in general need to be efficient and effective, the first because it is functionality must be performed precisely and second because these features must happen at the exact moment with the resources available. [10]

The development of software to Embedded Systems components requires the Software Engineering controls include in it is mechanisms allowing to optimize the final product beyond it is development process, since the components for embedded systems, in general, require for their functioning, smaller memory size, better execution time and response, lower power consumption among other restrictions (non-functional).[10]

Currently able to meet the characteristics of Embedded Systems, developers often fail to observe certain activities of software engineering since such activities do not take into account the specific needs of development and the final product of a component of Embedded Systems. The use of Software Engineering for the development of Embedded Systems should observe procedures in their mechanisms: a) better distribution of functionality between the hardware component and software depending on their 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 Embedded Systems, among other practices that can directly influence the quality of Embedded Systems.

## IV. Restictions and Limitations Embedded Systems

The Embedded Systems in general have in common some limitations, like owning a single specialty, i.e. perform a single function repeatedly, little information storage, among others, such limitations are given by the existence of design constraints that must be met, such as size, power consumption, cost, performance, etc..

Many of these restrictions depends on the intended use Embedded Systems, they become more rigid proportional to the criticality of their use, such as a Embedded Systems in a pacemaker, need to have an acceptable size to be able to be inserted into the human body in a simple and convenient way, requires immediate response time to meet your needs, requires low power consumption (battery) to minimize substitutions, must have cost so that it is use is accessible to people of all incomes and ultimately requires that it is operation is exact, because life depends on it is accuracy.

## V. Quality Process for Embedded Systems

A product of software quality is the goal to be achieved by all segments that use software products; While the quality of the software process can be translated into what is done to achieve this quality. 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. Progressively increase this quality has become a mission of software engineering. The Embedded Systems in particular, by it is characteristics, the criticality and the high degree of dependence requires more attention since we cannot transfer this concern only the final product, we should prevent failures rather than clean up after them, thus spreading this responsibility for all development. 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 condition.

To increase the product quality and quality of software development, designs and patterns, including international process, were created and are being used, however, a disadvantage of these standards is a generalist, thus making quality models difficult could be applied to products with restrictions. The lack of processes and development methods with specific guidelines for software of Embedded Systems, can derail the prevention of it is defects. 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 Systems industry, the concepts of quality with the same commitment from that applied in other engineering with disciplines, procedures, models and independent life cycles. Projects such software are becoming more complex, increasing it is use the amount of defects are increasing, and become even more visible by their criticality, exemplified by the number of aircraft accidents that cannot be evidence of human failure as the cause.[9]

The space missions are gaining a considerable increase in the amount of available memory and the capacity of the processors used in space [12] missions, so you can advance in the use and autonomy of embedded systems products as well as products for airlines, medical products, products for the automotive industry and other.

## VI. THE CMMI MODEL

The CMMI model (Capability Maturity Model Integration) is based on the concept of process maturity software, inspired by the techniques of TQM (Total Quality Management) and is certainly the most widespread in the software industry, has 5 levels of maturity of the model and allow an increasing scale control and visibility into processes and technical and managerial project outcomes software, acts as a reference to obtain adequate levels of quality, provides a common language, standardize goods and services and serve as legal support. The maturity levels provide a way to control or structure the performance of the organization 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.

## VII. PROPOSED WORK

It is understood that the embedded system has specific characteristics and thus need to be developed using specific processes well established and consolidated pattern. Studying quality models was observed that the application of it is practices in developing a process that meets the product feature of Embedded Systems, can contribute to the quality of the product to be available on the market. In this work, the practices of CMMI Level 2, which has directed it is activities for the project management structuring a process for the development of Embedded Systems which can ensure the management of requirements, processes are planned, executed, measured and controlled in use, and that existing practices are retained during times of stress.

Figure 01 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.

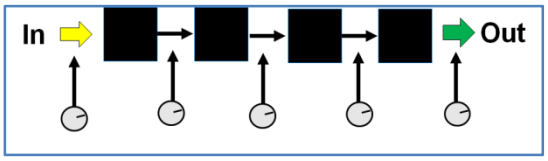


Fig. 1. CMMI Level 2 - Managed

For the development of Embedded Systems using a process classified as Level 2 of CMMI, enables the flexibility to develop products such as cases classified at higher levels of CMMI cannot be made possible to be applied to projects in Embedded Systems. The use of such practices, if implemented, should help in the identification, development and implementation of models for validation of the Embedded Systems behaviors before implementation, in the early stages, where to find errors are easier and cheaper to be fixed, i.e. eliminate design errors thus minimizing potential problems, losses or disasters.

For this study a process for developing Embedded Systems, 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 02 a process for development of Embedded Systems must follow a specialized life cycle since it is components develop specialized activities.

A set of related practices such as CMMI practices represent a process area, which, when implemented jointly and fully satisfy important goals for making improvements in that area.

The process areas of CMMI level 2 shown in Table 1, have goals that are extremely important and must be observed in addition to the conventional systems also in embedded systems, since their practices are directed to organize the completion of a management activity project activities that need to be applied more rigorously in embedded systems projects, which justifies it is use in processes for the development of such systems.

TABLE I. 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. |

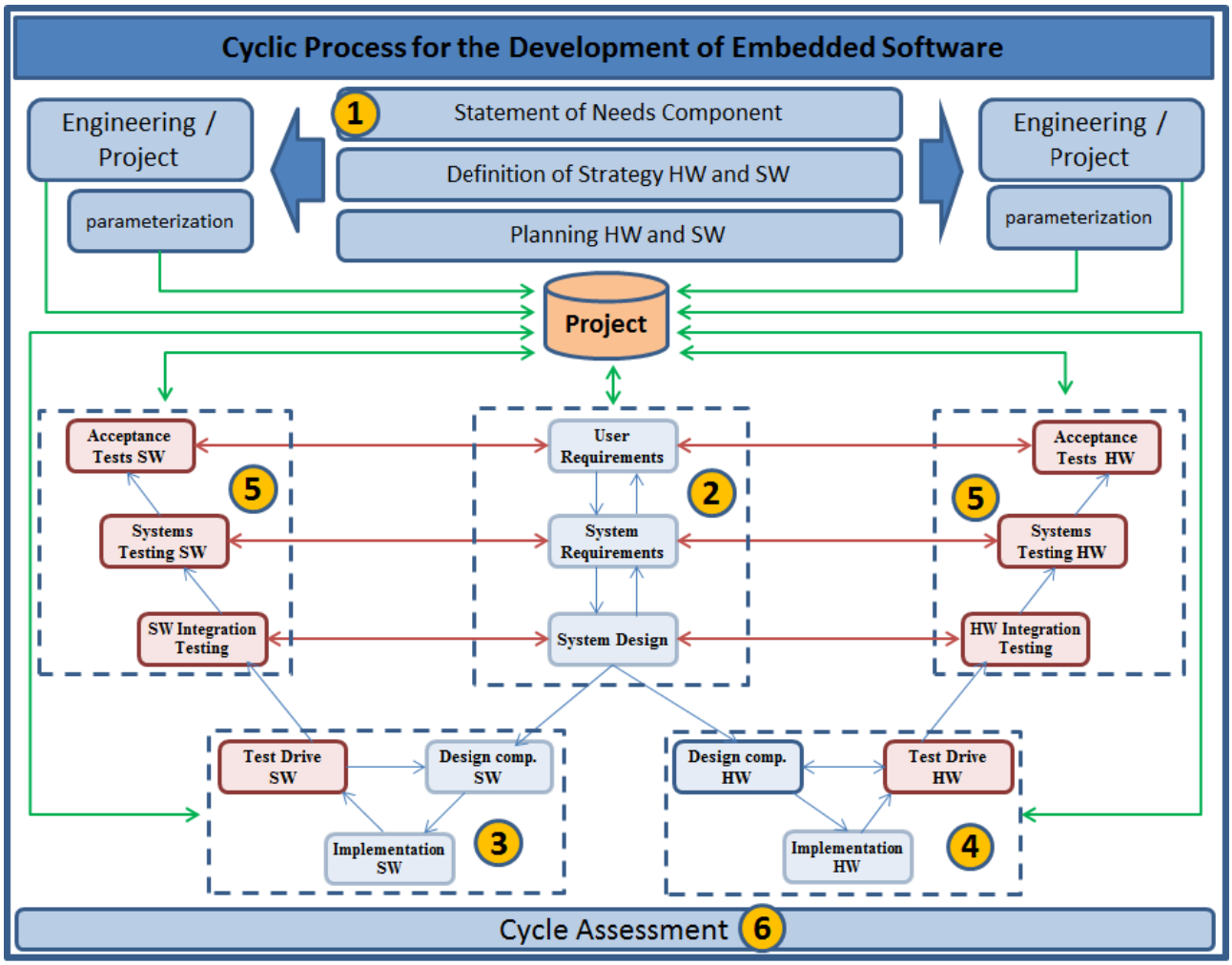


Fig. 2. Process for the Proposed Embedded Systems

### A. 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 Embedded Systems (software and documentation) as shown in figure 03.

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

At this stage one 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 result evaluation cycle.

### B. Phase Engineering

Each development cycle will result in one or more components of a Embedded Systems (hardware and software) performing the following activities: (a) Release (Statement of Requirement), (b) strategy, (c) Planning and (d) Preparation of Setting 1 shown in section in Figure 02.

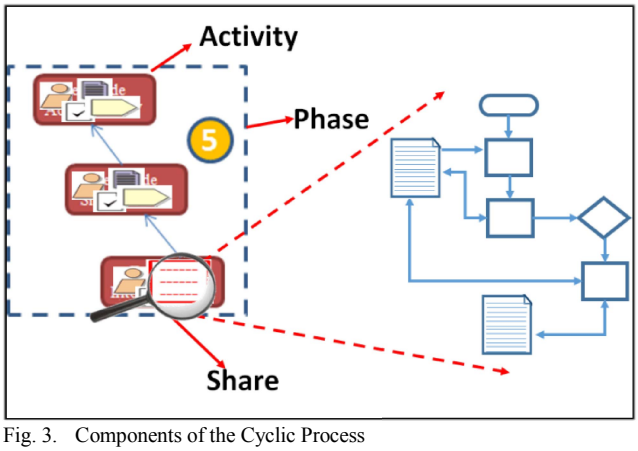


Fig. 3. Components of the Cyclic Process

### C. Phase: Product Details

Have common activities for hardware and software which are identified and defined in detail as presented in item 2 of figure 02. (a) User Requirements: Defining the needs and characteristics of Embedded Systems in full context, (b) System Requirements Specification Embedded Systems and

restrictions, and area features, (c) System Design: Embedded Systems Modeling identification and definition of functions to be performed by the hardware and software, and identify reuse or development of software and hardware.

These activities are for the independent components of hardware and software, producing shared artifacts.

### *D. Phase: Design Software*

At this stage we undertake the study of the software, as shown in item 3 of figure 02 with the following activities: (a) Project (study reuse) of the software component (b) Implementation (reuse) software (c) software verification.

### *E. Phase: Hardware Design*

At this stage we undertake the study of hardware, as presented in item 4 of figure 02 with the following activities: (a) Project (study reuse) of the hardware component, (b) Implementation (reuse) of hardware, (c) Verification of hardware.

### *F. Phase: Verification And Validation System*

Shown in Item 5 of figure 02, where the phase of verification and validation activities of the components are made:

(a) Hardware Test Plans (b) of the Software Test Plan, (c) Summary of Hardware testing, (d) Summary of Software Testing (e) Integration of Software X Hardware tests, (f) Integrated testing, (g) Evaluation of Integrated tests.

### *G. Phase: Cycle Assessment*

Analysis of activities in the cycle to adapt to the next cycle is shown in item 6 of figure 02.

## VIII. Expected Results and Contributions

As presented throughout this work, currently, the software engineering activities are directed to obtaining mechanisms for management and control processes for major developments, and thus improve the quality, productivity and performance of their processes. 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 its flow enables targeted to the characteristics of its various components (hardware and software) simultaneously and shared way procedures.

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

## References

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

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

[4] A. Medeiros Avelino, Processamento Embarcado aplicado a detecção de vazamento. Dissertação de mestrado Universidade Federal do Rio Grande do Norte - Rio Grande do Norte, dezembro de 2009.

[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 S w eden.

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

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

[9] M. Narumi Kondo, J. Reinaldo Silva, A. Yuuji Hira, M. Knörich Zuffo, "Estudo de Requisitos do Software Embarcado no Segmento da Telemedicina" - Congresso Brasileiro de Informática em Saúde - Brasil.

[10] F. Rech Wagner, "Minicurso Engenharia de Software para Sistemas Embarcados" - III Simpósio Brasileiro de Engenharia de Sistemas Computacionais - SBESC/2013 - UFRJ

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

[12] F. Novaes Kucinskis - Considerações sobre o Aumento da Autonomia Operacional no Segmento Espacial das Missões do INPE - Trabalho para o Exame de Qualificação de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espacial, área de concentração em Engenharia e Gerenciamento de Sistemas Espaciais - 2011.

[13] M. Aparecida 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. Paulo Marques 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] A. Cervigni Guerra; R. Maria Thienne Colombo - Tecnologia da Informação - Qualidade de Produto de software. PBQP Software - Jun-2009.

[16] H. Vargas Gomes, Metodologia de Projeto de Software Embarcado Voltada ao Teste - Dissertação de Mestrado Universidade Federal do Rio Grande do Sul - 2010.

[17] J. Cazuhiro Ossaada; L. Eduardo G. 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.

[19] H. Watts - The Team Software (TSP SM) - CMU/SEI-2000-TR-023 ESC-TR-2000 023 - SEI Joint Program Office - HQ ESC/DIB 5 Eglin Street Hanscom AFB, MA 01731-2116 - November - 2000 by Carnegie Mellon University.

[20] SOFTEX: <http://www.softex.br/mpsbr/>, Acesso em: Janeiro 2013.

[21] I.Sommerville - Engenharia de Software. 9a ed. São Paulo: Editora Pearson Education do Brasil, 2007.

[22] R. Pressman, S. Engenharia de Software. 6. ed. Rio de Janeiro: MCGraw-Hill, 2011.

[23] SEI. - CMMI for Development, Version 1.3 CMMI-DEV, V1.3 - Pittsburgh: Carnegie Mellon University, November 2010.