USING THE INTEGRATED CAPABILITY AND MATURITY MODEL IN THE DEVELOPMENT PROCESS OF SOFTWARE SYSTEMS

Marian Pompiliu CRISTESCU [1], Ana-Maria Ramona STANCU [2]

[1]Lucian Blaga University of Sibiu, Sibiu, Romania
[2]Dimitrie Cantemir Christian University, Bucharest, Romania

Abstract
In this paper is presented the CMMI model - a complex model that integrates aspects related to features such as capacity and maturity. It provides a modern approach to software development and improvement processes that support organizations in increasing their own performance. CMMI models can be used to guide the improvement of a project, a division, or a whole organization. Application of the CMMI model in software engineering and organizational development is a process improvement approach that provides organizations with the essential elements for streamlining processes. In addition, the paper describes briefly aspects related to the architecture, components and main features of the CMMI.

Key words: software engineering, model, system, organization, software quality.

JEL classification: O31, O32

1. Introduction

The Capability Maturity Model Integration Model (CMMI) is an interdisciplinary approach to system engineering and is currently being developed by a group of industry, administration and software experts from the Institute of Software Engineering (SEI) of University of Carnegie-Mellon, United States of America.

The model describes the framework of 5 evolution phases or levels of capability and maturity of development processes in an organization.

According to the Institute of Software Engineering, CMMI models "support the integration of traditional, independent organizational functions,

[1] *Faculty of Economics, Sibiu, Romania, marian.cristescu@ulbsibiu.ro*
[2] *ana_maria_ramona@yahoo.com*

setting process improvement objectives and priorities, providing guidance on quality processes, and a benchmark for assessing current processes."

CMMI models currently address three areas of interest:

1. Product and service development - CMMI for Development (CMMI-DEV) (CMMI Product Team, 2010),
2. Setting up, managing and delivering services - CMMI for Services (CMMI-SVC)(CMMI Product Team, 2010),
3. Acquisition of products and services - CMMI for Acquisition (CMMI-ACQ) (CMMI Product Team, 2010).

Any CMMI model, depending on the scope, can also be used to assess the process maturity of an organization.

Initially, the CMMI was designed for software engineering, but later it has been generalized over the years to include other areas of interest, such as developing hardware products, providing all sorts of services, and acquiring products and services.

The word "software" does not appear in the CMMI definitions.

This generalization of concepts of improvement makes the CMMI model extremely abstract.

It is not as specific to software engineering as its predecessor, CMM Software (M. C. Paulk, 1993).

The objective of the CMMI model was to improve the use of maturity models by integrating several different models into a single framework. Major sponsors include the Office of Secretary of Defense and the National Defense Industrial Association, both in the United States.

The CMMI model is the successor to the Capability and Maturity Model (CMM), known as CMM Software.

The latter was developed from 1987 until 1997. The Software Engineering Institute began in the mid-1990s a complex and tedious research & development process that ended with the obtaining and implementation of the Capability and Integrated Maturity Model.

Following the release of version 2.0 of the SW-CMM v1.1 model, which took place in 1993 (M. C. Paulk, 1993), the Institute of Software Engineering abandoned research in this direction and directed its research and development activities towards a General Capability and Maturity Model in which the characteristics of several fields of study can be found and integrated.

2. Choosing the appropriate CMMI model

The CMMI model consists of a central part, the nucleus, which extends as it includes new areas / disciplines.

This development method gives the model the integrating character.

In addition, he respects the development principles of Systems Engineering and Software Engineering, CMMI-SE / SW (Systems Engineering and Software Engineering).

Starting from these principles, so far models have been developed such as Integrated Product and Process Development CMMI-SE / SW / IPPD (Integrated Product and Process Development) and Initial Identification and Analysis Model of Procurement, i.e. CMMI-SE / SW / IPPD / SS (Supplier Sourcing, formerly Acquisition).

The following table lists the main areas of use of these models.

Table 1. CMMI models

CMMI model	Targeted field
SE/SW	(Software/System Engineering)
SE/SW/IPPD	(Software/System Engineering) focusing on the timely and continuous involvement of stakeholders.
SE/SW/IPPD/SS	(Software/System Engineering) focusing on the timely and continuous involvement of stakeholders and identifying and analyzing suppliers.

2.1. Structuring the model through two representations

In the development and implementation of CMMI models, at least two objectives that have been pursued in the integration process have been identified from the outset. These are:

1. Following the "stage-by-stage" maturity levels, it was concluded that they represent a major constraint on the CMM model in terms of the flexibility it needs when used within organizations. They pursue

vigorously the improvement of business processes by adapting them to the goals they have set;

2. Organizations are interested in switching from CMM v1.1 to the new CMMI model. The transition process must take into account the "protection" of the major investments these organizations have made so far.

In order to achieve these objectives, a solution was proposed and implemented, which involved obtaining a model through two representations, in stages (CMMI Product Team, 2002a), and continued (CMMI Product Team, 2002b).

It should be noted that, from the point of view of flexibility, "continuous" representation is better outlined and provides a more elaborate method of improvement, which follows the general objectives of the organization. In addition, this representation is considered to be more compatible with ISO 15504 / SPICE (Software Process Improvement and Capability Determination).

For those organizations that do not allow consistent investment to switch from CMM v1.1 to the CMMI, it is recommended to use the "staged" representation of the model.

The documentation of each of the two representations of the model is very well done and the documents, structured in seven chapters, are available and can be downloaded from the SEI website (CMMI Product Team, 2002a) (CMMI Product Team, 2002b).

2.2. Differentiated model representations

Analyzing the two representations of the model we find that they have relatively the same content, consisting of a number of 22 process domains (PAs). Moreover, there is a certain concordance between the general and specific objectives of the two representations. The transition in one direction of the results obtained from the evaluation in both the continuous and the staged representations is accomplished through a process known as equivalent phasing.

At the top level of the two representations there are significant differences. In the "stepwise" representation, the maturity levels defined in CMM v1.1 are identified, while the domains of processes are assigned to the four higher levels of the total of five maturity levels (Managed, Defined, Quantitatively Managed, and Optimizing).

Within the CMM model, the following maturity areas are defined:

- level 1 (initially) - is characterized by the actual development of the software;
- level 2 (repeatable) - the main feature of which is the process of tracking the way projects are deployed and planning and stabilizing the core requirements;
- level 3 (defined) - is characterized by activities aimed at defining and institutionalizing the software development process to ensure product quality control;
- level 4 (managed) - its main feature is the process of product quality planning and tracking of software measurement activities;
- level 5 (optimized) - the main feature of which is the continuous improvement of process performance.

Replacing maturity levels with capability levels (CLs) is an action specific to continuous representation, and this is intended to assign an individual measure for each process domain (PAs).

In the following table, the maturity and capability levels identified in the two model views maturity levels of the old CMM v1.1, as well as ISO TR 15504 (SPICE) (TR = Technical Report, i.e.) are highlighted.

Analyzing the data shown in the table identifies the similarities between the model representation of the model and the old SW-CMM model, as well as those between the continuous representation and the ISO 15504 standard.

Table 2. Maturity and Capability Levels of CMMI, CMM, and ISO 15504 models (adaptation after (H. M. Abdullah, A. M. Bhatti, 2009))

Representation: **Level:**	**CMMI in Maturity Stages**	**CMMI Continuing Capability**	**CMM v1.1 (1993) Maturity**	**ISO 15504 Capability**
5	*Optimized*	*Optimized*	*Optimized*	*Optimized*
4	*Quantitatively Managed*	*Quantitatively Managed*	*Managed*	*Predictable*
3	*Defined*	*Defined*	*Defined*	*Established*
2	*Managed*	*Managed*	*Repeatable*	*Managed*
1	*Initial*	*Performed*	*Initial*	*Performed*
0	-	*Incomplete*	-	*Incomplete*

2.3. Structure of the CMMI model

In activities related to increasing the performance of software development processes and, implicitly, general performances of organizations, it is recommended to use CMMI models, whose design allows descriptions of discrete levels of increase of these performances. Furthermore, the maturity levels, specific to the "staged" representation of the model, provide an algorithmic way of approaching, in stages, the process of increasing the general performances of organizations.

The maturity levels, presented in the "stepwise" representation of the CMMI model, influence the organization of the process domains shown in Figure 1. The practical, general and specific actions described here are associated with the general and specific objectives that are found in these areas of lawsuits. In addition, good practice rules are organized by common features.

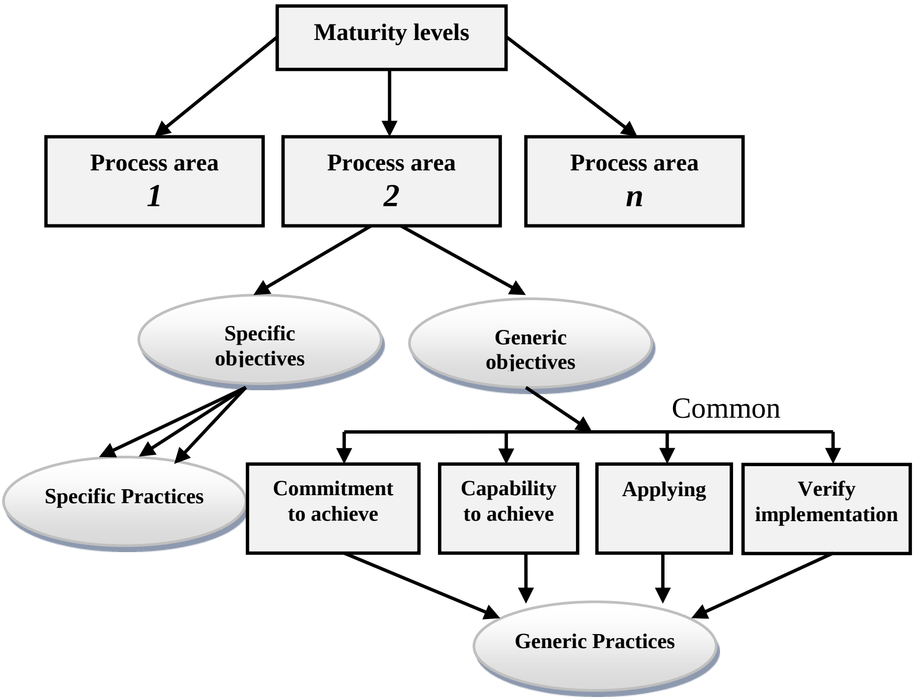

Figure 1. Organization of process areas, depending on maturity levels
(adaptation after (H. M. Abdullah, A. M. Bhatti, 2009))

Once the maturity level required to conduct activities aimed at increasing the global and specific performances of organizations associated with the selected process domains, the graphical representation is based on good practice rules that can be used to improve the business processes of organizations. Thus, it is envisaged to agree the processes that take place within the organization with the domains of processes implemented within the CMMI model. This must be done before using the CMMI model. The relationship action aims at enhancing the performance of processes within organizations, aiming at aligning the level of compliance, of organizations, with the standards described and implemented in CMMI models.

2.3.1. Objectives, best practices and objects associated with process areas

The main difference between v1.1. of the CMM model and the two representations of the CMMI model is represented by the number of process domains. Thus, we are dealing with a number of 22 process domains associated with CMMI, while version v1.1. of the CMM model implements a number of 18 process areas. The following table shows the areas of processes arranged in four categories (PACs). In addition, there are also process categories covered by ISO 15504.

Table 3. PACs from CMMI vs. ISO 15504

PACs from the continuous CMMI representation	PACs of the ISO 15504 standard
Process Management	Organization
Project Management	Management
Engineering	Engineering
Technical support	Technical support
	Client-Supplier

The relationship between maturity levels, specific to stepping representation, and domains of process categories (PACs), owned by continuous representation, is performed in matrix format.

At process areas level (PA - Process Arreas), we find a number of up to three specific Goals (SGs), which, in turn, have more specific practices (SPs). At the level of the standard are implemented the objectives and rules of good practice, which form the basis of its operational algorithm and which lead to the obtaining of typical objects (results).

Table 4. Examples of hierarchically exposed items

Model Item Designation	Effective name	Evaluation relevance	Comment
Representation (2)	Continuous	None	
Categories Process Areas (4)	Project Management	None	
Fields of Processes (22)	Risk management	None	
Generic Objectives (5)	Achieve specific goals	Requirement	"It must be fulfilled."
Generic Practices (18)	Identify the purpose of the work	Anticipated	"Possible alternatives."
Specific Objectives (48)	Identify and analyzes risks	Requirement	"It must be fulfilled."
Specific Practices (168)	Evaluate, classify, prioritize risks	Anticipated	"Possible alternatives."
Typical artifacts	Nominal risk list	None /Informative	"Just information."

The activities included in the process of improving the quality and performance of the organization are implemented as a generic cycle containing a number of 5 generic goals (GPs) wich are detailed at a 18 levels of generic practices (GPs). The current form of implementation of the standard is characterized by an increased level of redundancy due to the large number of generic objectives and practices, which is a major limitation of the standard, which is often noted by the organizations that use it. Table 4 presents a set of standard-specific elements hierarchically arranged.

In addition to the elements presented above, the model also contains the following components:

- sub-practices;
- examples;
- development;
- specific emphasis discipline.

These components of the model are not useful in assessing organizations' performance, but are highly suited to assessing practical activities and developing knowledge in software engineering.

3. Use of ARC and SCAMPI tools in the evaluation process

In the CMM model, CBA-IPI (CMM-based Appraisals for Internal Process Improvement, ie, improvement of core internal processes through evaluations implemented in the CMM model) is implemented as a valuation method, (D. K. Dunaway, 2001). Following the development of the CMMI model, a new assessment method, known as the SCMI (Standard CMMI Appraisal Method for Process Improvement), SCAMPI Upgrade Team, 2011, has become available. The development process has already reached version 1.3, a version with a high degree of complexity that includes all the evaluation requirements covered by the CMMI (Appraisal Requirements for CMMI).

When it is not desired to obtain a capability or maturity classification, it does not appear mandatory to carry out a complete SCAMPI assessment. Table 3 presents a set of 3 types of assessments that meet the requirements of the ARC method, which requires a high, medium and minimum resource consumption.

Table 5. Types of ratings available, according to the ARC method

Type of evaluation	Compatible ISO 15504	Use of SCAMPI	Requires evaluator	Team size
Class A	X	Possible	X	5-11
Class B	-	Partial	-	2-7
Class C	-	Partial	-	2-3

3.1. CMMI model limitations

- **objective and best practice identification system**: the current version of CMMI has a major drawback to users. This is the identification, by numbering, of objectives and good practices. The way to identify the objectives of a process area is by numbering "from 1 to n". Starting from this identification system, it is used to specify a specific objective of a process area by a "SG 3" label, and a generic target is associated with a "GG 5" tag. In practice, an objective is associated with a specific index, to which another index of "from 1 to m" should be added. It allows identifying, by the assigned number, the good practices associated with that objective. Moreover, the rules of good practice, specific, imply a classification of capability and a specification of it by a mechanism involving the inclusion of a dash-type symbol at its specific level. According to this procedure, a good practice rule can be labeled "SP 3.3-1", while a generic practice rule will have a tag like "GP 5.1". The use of these identification methods at the level of the organizations has led to the conclusion that an efficient method is to assign an order number to all specific objectives (1-48), or to add an abbreviation, the name , for each process area that looks like "MA-SG 1" or "RSKM-SG 3". Reducing the complexity of this identification mechanism requires assigning a benchmark for the capability level to advanced good practice rules such as "SP 2.1-3". This is an insignificant part of good "SP 2.2" practices that would lead to another variant, such as "SP 2.2-1";

- **two representations**: at the level of organizations there is an approach that requires the argumentation of the correctness of the choice of the two representations. Due to the identification of the two major disadvantages represented by redundancy and incoherence, it is necessary to accept a compromise that would allow the resolution of situations involving the emergence of conflicting interests. The experience of many organizations in the field of standardization processes implies the determination of a common vision, to the detriment of the singular, and the orientation towards optimal solutions in the process of representation;
- **redundancies**: many organizations believe that a major disadvantage, of the representation process, is the repetitive description of generic goals and practices, at the level of the 22 domains of processes implemented in the used models. A start-up approach to process development is recommended, followed by the description of each process area when avoiding a major pattern restriction.

3.2. Advantages of the CMMI model

The fact that the SW-CMM model was developed mainly for large organizations such as ministries and other state bodies and was not designed and developed to meet the needs and objectives of small and medium-sized organizations is an impediment and, implicitly, , a major limitation in the use of these organizations by these organizations. As a result, the design, development and implementation of the CMMI model has gone from the common needs and objectives of all organizations, and although not a highly performing tool, its facilities are much easier to use than a number an increasing number of organizations, irrespective of their size and the degree of complexity of their objectives.

- **capacity levels specific to each process area**: due to the functions of the CMMI model, and in particular its continuous representation, organizations using this representation have the ability to adapt their performance enhancement processes, and how they evolve, according to the objectives of the processes business and own needs, short or long term. Thus, these organizations resort to an effective method by choosing the areas of processes they want to focus on and allocate important resources to increase the performance of business processes;
- **small projects**: to increase its usefulness, the model must be developed in such a way that it can also be available and effective for smaller scale projects. This leads to a substantial reduction in fixed and additional costs imposed by the use of the model;
- **accentuations of disciplines**: because the CMMI model is one of a high standard of generality and standardization, it is associated with a large set of disciplines. The developers of the model have implemented the possibility of using mainly certain disciplines within the model components, which allows the elimination of the risk of generalization of these components, thus offering the possibility to customize them to the specific needs and objectives of different types of organizations;
- **three classes of assessments**: earlier versions of the CMMI, have brought to light a major limitation that refers to the excessive consumption of resources needed for the evaluation process. This restriction limits, quite a lot, the number of organizations capable and available to support these consumption and, implicitly, to use the

model in business performance evaluation activities. The development of the new evaluation methods, implemented by SEI institute in the form of the B and C classes, provide increased possibilities for efficiency of evaluation activities. Thus, they focus on minimizing costs by reducing resource requirements, demanded by older evaluation methods, resources that only large organizations could make available. Thanks to these facilities, both the efficiency of the evaluation process and the number of organizations that enable them to carry out such activities have increased.

4. Conclusions

A multitude of models has been proposed in the literature, but each with its drawbacks, some of which are common in most cases. One of the problems of the revised models is that none permits analysis based on non-existent fault data, i.e., a history of software use that has its operational phase for a period of time without any detected failure. Another common problem with models is that they only bear modest reliability requirements. There is no solution to this problem.

Most revised models also share the feature that they have been developed to shape reliability enhancement attributes. This is appropriate when a program has been developed by a well-disciplined team, and reports of developmental defects or, at least, defect statistics are available. However, from the point of view of the software user, it is more realistic to assume that only operational data records are available.

Over the past two decades, a large number of analytical models have been proposed and studied to assess the quality of software systems. Each model has to make certain assumptions about the software development process as well as about the testing environment. The environment can change depending on the software application, the development cycle, and the engineering design team's capabilities.

References:

- H. M. Abdullah, A. M. Bhatti, "The Optimum Measures Set Decision (OMSD) Model Deciding on Optimum Set of Measures in Software Organizations", Master Thesis Software Engineering Thesis no: MSE-2009:01, School of Engineering Blekinge Institute of Technology, Ronneby, Sweden, January, 2009;

- *** CMMI Product Team: “CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD/SS, V1.1, Staged),” Carnegie Mellon University / Software Engineering Institute, Technical Report CMU/SEI-2002-TR-012, Mar. 2002;
- *** CMMI Product Team: “CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Continuous Representation (CMMI-SE/SW/IPPD/SS, V1.1, Continuous),” Carnegie Mellon University / Software Engineering Institute, Technical Report CMU/SEI-2002-TR-011, Mar. 2002;
- *** CMMI Product Team: “CMMI for Development, Version 1.3,” Carnegie Mellon University / Software Engineering Institute, Technical Report CMU/SEI-2010-TR-033, Nov. 2010;
- *** CMMI Product Team: “CMMI for Services, Version 1.3,” Carnegie Mellon University / Software Engineering Institute, CMU/SEI-2010-TR-034, Nov. 2010;
- *** CMMI Product Team: “CMMI for Acquisition, Version 1.3,” Carnegie Mellon University / Software Engineering Institute, Technical Report CMU/SEI-2010-TR-032, Nov. 2010;
- *D. K. Dunaway and S. Masters*: “CMM-Based Appraisal for Internal Process Improvement (CBA IPI) Version 1.2 Method Description,” Carnegie Mellon University / Software Engineering Institute, Technical Report CMU/SEI-2001-TR-033, Nov. 2001;
- *M. C. Paulk, C. V. Weber, W. Curtis, and M. B. Chrissis*: “Capability Maturity Model for Software, Version 1.1,” Carnegie Mellon University / Software Engineering Institute, Technical Report CMU/SEI-93-TR-024 ESC-TR-93-177, Feb. 1993;
- *** SCAMPI Upgrade Team: “Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document” Handbook CMU/SEI-2011-HB-001, Mar. 2011.