Capability Maturity Model Integration for Beginners

Wajdi Aljedaibi, Abed A. Alsulami

Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia

DOI: 10.23956/ijarcse/V7I5.01507

Abstract— the evaluation of the software industry has become a necessity to any company competing in the software market by producing high-quality and low-cost software products. In this article, we have present a simplified presentation of the Capability Maturity Model Integration (CMMI) to organizations eager to understand the basic structure. In addition we present a review of other important models in evaluating software development processes such as SPICE (ISO/IEC15504), Bootstrap, ISO 9001:2000, and GQM. 

Keywords— CMMI, Capability, Maturity, software process, quality.

I. INTRODUCTION

The need for software engineering emerged after the third generation of powerful computers were widely used, upon which new tactics for designing and developing software were needed. Such needs lead to the creation of new methods and/or techniques to develop software more efficiently while reaching higher levels of quality. From that time, software grow in size and became large, complex, distributed, and hard to maintain. Consequently, new software development processes were introduced to manage software projects difficulties and to shift towards more efficient software development processes producing high quality software products [1, 2]. Many software companies and/or government agencies aims to produce high quality products within preset budget and schedule constraints. Considering quality attributes early in the process will indeed raise quality [3], making resultant products more reliable. Needless to note, quality products are results of quality processes [2]. In other words, only quality software processes give software firms the capability to produce quality software products. Software process assessment aid to bring a focus to the software development process and gears efforts towards future improvements. In addition, an assessment report for each process is to be followed for each activity on which improvement goals are based. Capability Maturity Model Integration (CMMI) [1] is a framework for improving/managing processes that can help software firms to achieve higher levels of quality products with minimum cost [1, 4]. Moreover, CMMI introduces work activities that organizations can follow to manage, improve, and assess their processes within a framework of 5 Maturity Levels each of which indicates how an organization execute required software development processes.

II. BACKGROUND

Following a particular software process model is not necessarily sufficient to produce a quality product and deliver it on time. An assessment model that guarantee these processes meet standard software engineering practices is needed. The evaluation of software processes includes the appraisal and assessment of all process activities, process tools, and process methods of developing and testing products. The actual purpose of the appraisal effort is to clarify the strength and weakness of each stage of the process and how can we plan for improvement. While planning for a software activity sets goals and objectives, assessment uncovers to what extent these goals have been achieved [1]. Thorough understanding and proper documentation of a process are, however, prerequisites to successful process assessment. Improving software quality is mainly based on the quality of the software process, and thus, software process assessment is an activity of an essential importance [2, 4]. In essence, applying software process assessment is expected to result in more effective and efficient, i.e. mature, software development processes. However, such a shift in software process activities requires strong organizational commitment.

<table>
<thead>
<tr>
<th>Process Attributes</th>
<th>Key issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understandability</td>
<td>Is the process understandable for people who use it? To what extent?</td>
</tr>
<tr>
<td>Standardization</td>
<td>Is the process a generic standard? To what extent?</td>
</tr>
<tr>
<td>Visibility</td>
<td>Is the process finishing with clear result?</td>
</tr>
<tr>
<td>Measurability</td>
<td>Can we measure process activities?</td>
</tr>
<tr>
<td>Supportability</td>
<td>Do software tools support the process activities?</td>
</tr>
<tr>
<td>Acceptability</td>
<td>Is the process acceptable for people who use it?</td>
</tr>
<tr>
<td>Reliability</td>
<td>Did the process designed to avoid errors before they result in product errors?</td>
</tr>
<tr>
<td>Robustness</td>
<td>Can the process go on despite unexpected problems?</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Did the process designed to keep pace with developments, improvements, and changing organizational requirements?</td>
</tr>
</tbody>
</table>
Assessment methods depend heavily on measurement, through which a set of process-based attributes are quantified and monitored, see table 1. Moreover, many assessment models were established to evaluate software development processes such as SPICE (ISO/IEC15504), Bootstrap, ISO 9001:2000, GQM and CMMI models [2, 3, and 7]. Below we give an overview of the above models.

**SPICE Model**

Software Process Improvement and Capability determination (SPICE) model is a framework used to improve software development process and examine the ability of a process to produce quality software. It also compatible with ISO/IEC15504:2003 and ISO 12207[9]. The model introduces a full framework that includes process management model standards for processes assessment under consideration in addition to the construction, selection, use of assessment tools, and preparation for evaluators [10]. The main consideration of SPICE is to develop handle rating profiles, environment of the process is being assessed, facilitating self-assessment and guarantee to suitable for all kind of organizations and applications. The capability of the organization will be determined based on assessment results of product and processes according to the documents that guide goals and basic activities. SPICE represents a reference model to mature the process improvement in six levels, as shown in Fig.1 and described below:

1. **Not performed**: processes cannot produce products at this level.
2. **Performed informally**: the outcomes at this level depend on personal knowledge without planning or tracking.
3. **Planned and tracked**: processes and products at this level are confirmed by standards and requirement. Products will be delivered on time based on quality requirements.
4. **Well-defined**: at this level, are processes able to achieve the outcomes by applying software engineering principles.
5. **Quantitatively controlled**: at this level, the desired outcomes will be achieved by executing existing process consistently. Measuring management goals and performance quantitatively will enhance the capability of prediction.
6. The last level is **continuously improved**, by using several types of statistical methods to continuous improvement to achieve the future objectives.

![Fig. 1 SPICE Model](image1)

**Bootstrap Model**

A bootstrap model is process assessment framework for software development projects according to the ISO 9000 standard [11]. The objectives of this model are to provision the evaluation of the process fitness, determine the process’s strengths and weakness, plan for improvement to supporting achieving goals, and enhance the effectiveness. Assessment is performed during process activities rather than after its completion by collecting data from the organization’s survey developing the software. Finally, data will be stored in a central database.

![Fig. 2 ISO 9001:2000 Model](image2)
ISO 9001:2000 Model
ISO 9001:2000 is a general standard for quality management using Plan-do-check-act cycle for software project [12, 13, and 14]. According to the software context:

"Plan" meet customer requirement, goals, activities and task for high quality.

a) "Do" apply the software process.

b) "Check" observing and assess the process to guarantee that all requirement implemented for quality management.

c) "Act" continues all software process improvement activities.

GQM Paradigm
The Goal/Question/Metric Paradigm (GQM) [1, 15] is a flexible measurement model that enables goal-based assessment. The model is driven by a goal upon which a number of questions are generated to reflect how such a goal is achieved. Answers to the set of questions should yield a group of metrics from which goal assessment is possible Fig.3.

![GQM Paradigm](image)

**Fig. 3 GQM Paradigm**

### III. CAPABILITY MATURITY MODEL INTEGRATION

CMMI is a references framework presents a clear definition of what software firms should do to change their internal software development processes, which will lead to improving quality and performance. With four Capability Levels and five maturity levels, the CMMI model defines essential processes that are required to build low cost high end quality products. The CMMI model interconnects and integrates all related processes needed for a producing software products in a complete single model. It helps software organizations to answer questions such as:

- What are we good at?
- What are we improving?
- Which parts of a particular process that should be improved?
- Is the process we use working well.
- How practical is a particular activity within a process?
- Can we improve our final products?

CMMI also can help organizations detect and achieve measurable business goals, build high-quality products, increase customer's satisfaction and guarantee that we are working as professional as possible. Software organizations will be ranked based on specific and generic goals at a Capability Levels or a Maturity Levels.

**A Brief history**

In an attempt to assist companies and government organizations in the provision of high-quality products and low cost. The Carnegie Mellon University founded the CMMI, supported by US Department of Defence in 1986 [18]. When the role of information systems started to gain importance in the 1960s, the need to develop quality software. Despite the fact, the software companies and related industry were growing quickly, software development processes were unprincipled and projects failures were common. During the 80s, various US military projects including software sub-contractors were late on their delivery schedule and over-budget. At a later stage, the United States Air Force [27] provided funding at the Software Engineering Institute (SEI) to study causes of this problem and how can be overcome. In the meantime, Watts Humphrey had just begun improving process maturity models based on "The Quality Management Maturity Grid (QMMG). It was further enhanced by Crosby during 1979. QMMG is a matrix used by organizations to assess their process maturity. Then, the U.S. Air Force [27] requested from Humphrey to joined to SEI and making his process maturity model officially to assist the U.S. Department of Defence [28]. The conclusion of his investigation was a basic framework for the USA army to use as an objective evaluation of software development process and capability maturity subcontractors'. Watts Humphrey’s Capability Maturity Model (CMM) was released in 1988. A year after, a book titled ‘Managing the Software Process’ [21] have been published. In 1991, SEI published the Capability Maturity Model for Software development (SW-CMM) Version 1.0 rely on knowledge in utilizing the software process maturity model and the maturity survey for analyzing difficulties and development processes, In 1995 CMM Version 1.1 was published as a book after two years of becoming alive [16]. Consequently, there were other additional maturity models developed such as Software Acquisition CMM, Systems Engineering Capability Maturity Model (SE-CMM®), the People CMM, the Integrated Product Development Capability maturity model (IPD-CMM®), etc.

At the point when the assessment process of SW-CMM V2.0 was completed in 1997, the idea of a standalone CMM was discarded via the resolution to publish SW-CMM v2.0 underneath the CMM Integration (CMMI) [1] project. The integration project of CMM was a combined effort between government, SEI, and software industry to merge a variety of
maturity models including several specialties into a generic standard model that is used for organization level process development activities. Then at a later stage, CMMI team released the standard CMMI, training, and evaluation method, which combined software and engineering systems. Moreover, CMMI was created to be a foundation of upcoming integration of other fields. CMMI V1.1 was distributed in 2002. The enhancement of V1.1 was published as CMMI V1.2 by adding more examples and elaborations. At the same time, IPPD activities were combined and become more simplified. This version presented the conception of CMMI “Constellations “, which are the groups of CMMI elements created to enhance the difficulty facing some of the specific areas of activity. The three elements were developed in V1.2 are CMMI for Services (CMMI-SVC), CMMI for Acquisition (CMMI-ACQ) and CMMI for Development (CMMI-Dev) [16]. The three elements were released in separate time, which affected some misunderstandings [19]. In 2010 CMMI V1.3 was released with all the three categories SVC, ACQ and Dev at the same time. The main different between V1.2 and V1.3 were remove the IPPD addition, supporting agile model, and high level of maturity explanations.

CMMI levels
With twenty two process areas the CMMI for Development cover all activities needed by a successful quality software product development. We can choose one of two representations the Staged Representation (*maturity level*) or the Continuous Representation (*capability level*). The most common selection is staged representation, which divided process areas into five maturity levels. An organization, however, can also choose the most process areas that effect of them work on by using the continuous representation. The organization choosing Staged Representation will follow a complete model of process areas that are categorized by maturity Level. On the other hand, choosing Continuous Representation lead the focus of the organization to its process areas based on their interest or needs to improve on specific process areas which are categorized by Category. Each Process Area in the CMMI has:
1. Specific goals (SGs), and
2. Specific practices (SPs) to determine the expected best behaviors of projects and organizations.

There are also more than ten generic practices (GPs) under three main generic goals (GGs) that provide roadmap for organizational quality involving behaviours such as monitoring process performance, measuring quality, evaluating compliance and training.

Understanding Capability level
In a continuous representation, there are four Capability Levels as shown in fig. 4:
1. **Level 0:** Incomplete
2. **Level 1:** Performed
3. **Level 2:** Managed
4. **Level 3:** Defined

To achieve the best capability of specific process area the organization must achieve all specific goals on this process.

1. **Capability Level (0)** the process in Capability Level 0 is considered as incomplete, which mean the process area activities partially performed or dose not performed totally. At least one of the specific goals are not fulfilled.
2. **Capability Level (1)** the process in Capability Level 1 is considered as a performed that mean the process achieves the required work to generate Work Products. The specific goals are fulfilled. Unfortunately, the results of improvements can be lost throughout times of stress. The organization must be applying Generic goals (GGs) to guarantee that improvements are supported.
3. **Capability Level (2)** the process in Capability Level 2 is considered as a managed. The process in managed level is already performed, however in this level the process must execute with preset plan and according organization policy. The outputs are controlled, the process is monitored and evaluated by involving relevant stakeholders.
4. **Capability Level (3)** the process in Capability Level 3 is considered as a defined. A process is said to be defined if it is already a managed process. The custom fitted via the organization’s collection of standard procedures as per the organization’s fitting rules. These rules keep the process maintained and contribute to process in relation to the organizational procedure assets. The main different between Capability Levels 2 and 3 is the extent of guidelines, process depictions, and methods. In Capability Level 2, they can be very different in each process. Nevertheless, in Capability Level 3 they are fitted from the repository of standard processes to match a specific structural component and they are steadier, excepting the variation accepted by the guidelines.

![Fig. 4 Capability Levels](image-url)
Understanding Maturity levels

Maturity Levels are used to improve the overall performance of the organization based on a set of general and specific practices relevant to a set of predefined process areas. All CMMI models were built to reflect maturity in their content, which can help the organization to provide means of describing its performance. General and specific practices are linked to the general and specific objectives linked with each pre-outlined group of process areas whose accomplishment is essential for a transition to a higher Maturity Level. The organization using Maturity Levels are evaluated based on the ability of achieving the generic goals linked with each pre-outlined of levels and the specific goals of each process areas. The five Maturity Levels are set from 1 to 5 each level representing a foundation of continuous improvement of processes as shown in table 2.

1. **Level 1: Initial**
2. **Level 2: Managed**
3. **Level 3: Defined**
4. **Level 4: Quantitatively Managed**
5. **Level 5: Optimizing**

As you can see the Maturity Levels 2 and 3 have the same name as Capability Levels 2 and 3. This term was intended due to similarities between Maturity Levels and Capability Levels. The main difference between them is Maturity Levels are applied to describe structural progression according to a group of process areas. On the other hand, the improvement at Capability Levels characterize organizational based on each process area individually. Initial at Maturity Level 1, the organization usually cannot repeat its success which made by professional people effort. The processes are usually of an ad hoc and disordered nature because of the organization’s environment.

1. **Maturity Level (2) - Managed** the organization sets the foundation for the development of high-quality and low-cost software through the institutionalization of selected process areas in project management, business management, and support. The organization creates work groups that in turn define the software development strategy, develop work plans, and monitor & control work to ensure timely delivery. Additionally, the organization establishing agreements with customers, developing customer’s requirements and contractual requirements. Institutionalization is also managed for the configuration management to ensure the quality assurance of processes and products and the ability to measure, and analyze the performance of process areas. In addition, workgroups, work activities, processes and work products are managed based on the organization's policy. Organizations should also provide resources, training, and assign responsibilities for the implementation of each process. Stakeholders are also involved in this level and a periodic evaluation of the commitment to the process is made to involve senior management. In order to ensure that the practices proceed according to the plan drawn even in difficult times and can be adjusted when needed.

<table>
<thead>
<tr>
<th>Maturity levels</th>
<th>Processes Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizing</td>
<td>Improvement based on Quantitatively Managed characteristics</td>
</tr>
<tr>
<td>Quantitatively Managed</td>
<td>Monitoring using statistical analysis and Controlled via quantitative techniques.</td>
</tr>
<tr>
<td>Defined</td>
<td>The organizational procedures are well understood. Processes, standards and tools, are defined</td>
</tr>
<tr>
<td>Managed</td>
<td>Planned, documented, performed, monitored, and controlled</td>
</tr>
<tr>
<td>Initial</td>
<td>Unpredictable, can’t Managed or controlled, and always reactive.</td>
</tr>
</tbody>
</table>

2. **Maturity Level (3) - Defined** the processes are defined for business management, where they establish project management principles, business management, and best practices in software development to ensure that they meet the requirements. It should be noted that these are well defined and understood processes and they are sufficiently documented. The main feature of this level is that processes are usually described more firmly than at Maturity Level 2. The processes in level 3 are clearly defined in terms of inputs, outputs, standards (entry-exit), activities, and role distribution so that processes are managed understanding the processes activities, detailed measurements, and work products. Therefore, the processes are more consistent than the Maturity Level 2 except for permissible differences. At this level, further improvements are also made to activities associated with process areas at Maturity Level 2. General practices related with general objective 3 that have not been addressed at Maturity Level 2 are applied during Maturity Level 3.

3. **Maturity Level (4) - Quantitatively Managed** the organization sets quantitative goals for quality performance and operations is used as standards for the management of operations based on the needs of customers and end users. This process is carried out with statistical significance and is managed throughout the life of the operations. Unlike Maturity Level 3, the ability to predict in Maturity Level 4 is the most important characteristic of this level by applying statistical methods and other quantitative methods.

4. **Maturity Level (5) - Optimizing** the organization continuously improves its operations based on a quantitative understanding of the organization's business objectives and performance needs. To understand the underlying differences behind each process and its outcome, the organization uses a measureable methodology. The business objectives and organization performance are reviewed to monitor any change in the performance of activities. Using statistical methods
and quantitative methods, defined processes, standard workgroups and supporting technology are targeted at improvement and measurable activities. The fundamental difference at this level is that it is concerned with the performance of the organization in general, using data collected from multiple groups. Explore performance gaps or gaps based on data analysis to guide the process of organization improvement which in turn results in measurable performance improvement.

Understanding process areas
A process area is a collection of interconnected practices in an area that achieves a set of objectives when applied collectively to improve that area. In CMMI each process area is structured into three components:

1. Required,
2. Expected, and
3. Informative.

These components, as shown in fig. 5, are necessary to interpret the process area. Required component are the components necessary to achieve improvement in a particular process area. It can be expressed by generic and specific goals and used in the evaluation process as the basis for a decision that has been achieved in the process area. Expected component are those that describe important activities in achieving one of the required components and guide those implementing the improvements or conducting the assessment. The expected components in CMMI are specifics and generics practices. Informative component help to understand further details about originally requested and anticipated component, for example detailed sub-practices, notes, and work products.

<table>
<thead>
<tr>
<th>Maturity</th>
<th>Process area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizing</td>
<td>Causal analysis and resolution (CAR)</td>
</tr>
<tr>
<td></td>
<td>Organizational Performance Management (OPM)</td>
</tr>
</tbody>
</table>

Fig.5 Process Area Structure
<table>
<thead>
<tr>
<th>Quantitatively Managed</th>
<th>Organizational process performance (OPP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quantitative project management (QPM)</td>
</tr>
<tr>
<td>Defined</td>
<td>Organizational process definition (OPD)</td>
</tr>
<tr>
<td></td>
<td>Organizational process focus (OPF)</td>
</tr>
<tr>
<td></td>
<td>Organizational training (OT)</td>
</tr>
<tr>
<td></td>
<td>Integrated project management (IPM)</td>
</tr>
<tr>
<td></td>
<td>Risk management (RSKM)</td>
</tr>
<tr>
<td></td>
<td>Requirements development (RD)</td>
</tr>
<tr>
<td></td>
<td>Technical solution (TS)</td>
</tr>
<tr>
<td></td>
<td>Product integration (PI)</td>
</tr>
<tr>
<td></td>
<td>Verification (VER)</td>
</tr>
<tr>
<td></td>
<td>Validation (VAL)</td>
</tr>
<tr>
<td></td>
<td>Decision analysis and resolution (DAR)</td>
</tr>
<tr>
<td>Managed</td>
<td>Requirements management (REQM)</td>
</tr>
<tr>
<td></td>
<td>Project planning (PP)</td>
</tr>
<tr>
<td></td>
<td>Project monitoring and control (PMC)</td>
</tr>
<tr>
<td></td>
<td>Supplier agreement management (SAM)</td>
</tr>
<tr>
<td></td>
<td>Measurement and analysis (MA)</td>
</tr>
<tr>
<td></td>
<td>Configuration management (CM)</td>
</tr>
<tr>
<td>Initial</td>
<td>Process and product quality management (PPQA)</td>
</tr>
</tbody>
</table>

**IV. CONCLUSION**

CMMI [1] is a reference framework that provides a clear definition of what software firms must do to further improve their processes, which leads to improved performance and the production of high-quality products at low cost. The Quantitative Development has *twenty-two practical areas* on all the activities needed by successful software development software. It can be used in staged representation, five levels of maturity or continuous representation, and four levels of capability. Each process area has a set of specific goals to be achieved. It is divided into a set of specific practices and each particular practice produces a single work product. When all the specific goals are achieved, the process area is at the level 3. With the achievement of a set of generic goals for a certain level of maturity starting from level 2 of a set of process areas, the organization has achieved the corresponding Maturity Level of these processes.

**REFERENCES**


