ASSESSMENT OF CMM AND ITS IMPACT ON SOFTWARE QUALITY

Dr. S. Balasubramanian
Management Consultant & Research Supervisor
Anna University Coimbatore

Mr. V. Antony Joe Raja
Research Scholar
School of Management Studies
Anna University Coimbatore

ABSTRACT

In software development, three major components determine the quality of a product: the people that develop a software system, the technology that is employed by them, and the organization of the process of development. In contrast to the former two components, the development process has been establishing itself as a major factor just recently starting maybe about ten years ago with the ever growing size and complexity of software projects. This may be due to the fact that development processes are primarily concerned with supporting management of software projects, and hence do not produce many seizable results in the form of products or the like.

INTRODUCTION

The Capability Maturity Model Software provides software organizations with guidance on how to gain control of their processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence. The CMM was designed to guide software organizations in selecting process improvement strategies by determining current process maturity and identifying the few issues most critical to software quality and process improvement. By focusing on a limited set of activities and working aggressively to achieve them, an Organization can steadily improve its organization-wide software process to enable continuous and lasting gains in software process capability. The staged structure of the CMM is based on principles of product quality that have existed for the last sixty years. In the 1930s, Walter Shewart, promulgated the principles of statistical quality control. His principles were further developed and successfully demonstrated in the work of W. Edwards Deming and Joseph Juran. These principles have been adapted by the SEI into a maturity framework that establishes a project management and engineering foundation for quantitative control of the software process, which is the basis for continuous process improvement.

STRUCTURE OF THE CMM

The CMM is composed of five maturity levels. With the exception of Level 1, each maturity level is composed of several key process areas. Each key process area is organized into five sections called common features. The common features specify the key practices that, when collectively addressed, accomplish the goals of the key process area. This structure of the CMM is illustrated in Figure 1.
Maturity levels
A maturity level is a well-defined evolutionary toward achieving a mature software process. The five maturity levels provide the top-level structure of the CMM.

Process capability
Software process capability describes the range of expected results that can be achieved by following a software process. The software process capability of an organization provides one means of predicting the most likely outcomes to be expected from the next software project the organization undertakes.

Key process areas
Each maturity level is composed of key process areas. Each key process area identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability at that maturity level. The key process areas have been defined to reside at a single maturity level. For example, one of the key process areas for Level 2 is Software Project Planning.

Goals
The goals summarize the key practices of a key process area and can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area.

Common features
The key practices are divided among five Common Features sections: Commitment to Perform, Ability to Perform, Activities Performed, Measurement and Analysis, and Verifying Implementation. The
common features are attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting. The Activities Performed common feature describes implementation activities. The other four common features describe the institutionalization factors, which make a process part of the organizational culture.

Key practices
Each key process area is described in terms of key practices that, when implemented, help to satisfy the goals of that key process area. The key practices describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area. For example, one of the practices from the Software Project Planning key process area is "The project's software development plan is developed according to a documented procedure."

KEY PROCESS AREAS OF THE CMM

Figure 2 lists the key process areas for each maturity level in the CMM. Each key process area identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability. The key process areas have been defined to reside at a single maturity level. The key process areas are building blocks that indicate the areas an organization should focus on to improve its software process. Key process areas identify the issues that must be addressed to achieve a maturity level.
The key process areas at Level 2 focus on the software project's concerns related to establishing basic project management controls. Descriptions of each of the key process areas for Level 2 are given below.

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. This agreement with the customer is the basis for planning (as described in Software Project Planning) and managing (as described in Software Project Tracking and Oversight) the software project. Control of the relationship with the customer depends on following an effective change control process (as described in Software Configuration Management).

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project. These plans are the necessary foundation for managing the software project (as described in Software Project Tracking and Oversight). Without realistic plans, effective project management cannot be implemented. The purpose of Software Project Tracking and Oversight is to establish adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans.

The purpose of Software Subcontract Management is to select qualified software subcontractors and manage them effectively. It combines the concerns of Requirements Management,
Software Project Planning, and Software Project Tracking and Oversight for basic management control, along with necessary coordination of Software Quality Assurance and Software Configuration Management, and applies this control to the subcontractor as appropriate.

The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and of the products being built. Software Quality Assurance is an integral part of most software engineering and management processes.

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. Software Configuration Management is an integral part of most software engineering and management processes. The key process areas at Level 3 address both project and organizational issues, as the organization establishes an infrastructure that institutionalizes effective software engineering and management processes across all projects. Descriptions of each of the key process areas for Level 3 are given below:

The purpose of Organization Process Focus is to establish the organizational responsibility for software process activities that improve the organization's overall software process capability. The primary result of the Organization Process Focus activities is a set of software process assets, which are described in Organization Process Definition. These assets are used by the software projects, as is described in Integrated Software Management.

The purpose of Organization Process Definition is to develop and maintain a usable set of software process assets that improve process performance across the projects and provide a basis for cumulative, long-term benefits to the organization. These assets provide a stable foundation that can be institutionalized via mechanisms such as training, which is described in Training Program.

The purpose of Training Program is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently. Training is an organizational responsibility, but the software projects should identify their needed skills and provide the necessary training when the project's needs are unique.

The purpose of Integrated Software Management is to integrate the software engineering and management activities into a coherent defined software process that is tailored from the organization's standard software process and related process assets, which are described in Organization Process Definition. This tailoring is based on the business environment and technical needs of the project, as described in Software Product Engineering. Integrated Software Management evolves from Software Project Planning and Software Project Tracking and Oversight at Level 2.

The purpose of Software Product Engineering is to consistently perform a well-defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently. Software Product Engineering describes the technical activities of the project, e.g., requirements analysis, design, code, and test.

The purpose of Intergroup Coordination is to establish a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently. Intergroup Coordination is the interdisciplinary aspect of Integrated Software Management that extends beyond software engineering; not only should the software process be integrated, but the software engineering group's interactions with other groups must be coordinated and controlled.

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently. An important corollary effect is to develop a better understanding of the software work products and of the defects that can be prevented. The peer review is an important and effective engineering method that is called out in Software Product Engineering and that can be implemented
via Fagan-style inspections [Fagan86], structured walkthroughs, or a number of other collegial review methods [Freedman90].

The key process areas at Level 4 focus on establishing a quantitative understanding of both the software process and the software work products being built. The two key process areas at this level, Quantitative Process Management and Software Quality Management, are highly interdependent, as is described below:

The purpose of Quantitative Process Management is to control the process performance of the software project quantitatively. Software process performance represents the actual results achieved from following a software process. The focus is on identifying special causes of variation within a measurably stable process and correcting, as appropriate, the circumstances that drove the transient variation to occur. Quantitative Process Management adds a comprehensive measurement program to the practices of Organization Process Definition, Integrated Software Management, Intergroup Coordination, and Peer Reviews.

The purpose of Software Quality Management is to develop a quantitative understanding of the quality of the project's software products and achieve specific quality goals. Software Quality Management applies a comprehensive measurement program to the software work products described in Software Product Engineering.

The key process areas at Level 5 cover the issues that both the organization and the projects must address to implement continuous and measurable software process improvement. Descriptions of each of the key process areas for Level 5 are given below:

The purpose of Defect Prevention is to identify the causes of defects and prevent them from recurring. The software project analyzes defects, identifies their causes, and changes its defined software process, as is described in Integrated Software Management. Process changes of general value are transitioned to other software projects, as is described in Process Change Management. The purpose of Technology Change Management is to identify beneficial new technologies (i.e., tools, methods, and processes) and transfer them into the organization in an orderly manner, as is described in Process Change Management. The focus of Technology Change Management is on performing innovation efficiently in an ever-changing world.

The purpose of Process Change Management is to continually improve the software processes used in the organization with the intent of improving software quality, increasing productivity, and decreasing the cycle time for product development. Process Change Management takes the incremental improvements of Defect Prevention and the innovative improvements of Technology Change Management and makes them available to the entire organization.

PROBLEM STATEMENT

The Problem: Software intensive projects notorious for cost overruns, schedule slips, and disappointing quality and they are becoming more ubiquitous and complex. Hence the Project is undertaken to Give Solutions to organizations, to establish and improve the software processes by which they develop and maintain their software work products, they progress through levels of maturity.

REVIEW OF THE LITERATURE

The literature contains a Overview on CMM (Capability Maturity Model), five levels of it and each key process areas at every levels of the process maturity. It talks about how software development takes place in an organization to develop quality software and its Impact on final Quality of Delivery.

PROJECT AS A CONVERSION PROCESS
The business analysis interview is structured around a grid of information around a grid of information. The grid forms a “H” hence the name. It consists of five components:

- **Inputs.** What the name person needs to do their job.
- **Outputs.** What the person produces.
- **Functionality.** What the person does.
- **Business Rules.** What rules govern the way the person works.
- **Data.** The people places and things the person needs to keep track of.

![Figure 3: Project Conversion Process](image)

**Inputs & Outputs:** By defining the inputs and outputs, the scope can be further refined. Obviously this should have happened at project inception but by defining what comes into the area, and what is produced, it helps define scope at a lower level of detail. It is likely that the questioning will go in loops.

**Functionality:** Functionality will be at different levels of granularity. One piece of functionality may be to check it is an existing customer. Another may be to check if the customer is part of a group (which is a lower level than checking the customer). Another may be to check customer details, which covers both above. At the first interview, it is better to keep focused on getting information rather than sorting information.

**Data:** The question “What are the people, places and things you want to keep track of?” is invaluable for a BA. The vast majority of users don’t think in terms of databases. Nor should they. They just keep track of things. Data comes up all through a discussion. When it does, drop it in this box.

**Business Rules:** As rules emerge, they should be dropped into the business rules box like data, they are woven through the BA is told.

**Inputs**
- To get detailed knowledge about CMM.
- To understand ERP assessment of service delivery of quality.
- To assess the various packaged solutions through ERP planning which effects the Final quality of delivery to defin the various customized solutions.
- To understand the 5 levels of CMM and to assess the disciplined, measured and Continuous improving process of quality.

**Outputs**
- The role of CMM and its impact on ERP solutions and final quality of delivery.
- The role of ERP and its implementations.
- Service delivery.

**Functionality**
- The need for certain activities to have been completed before a project can start.
- The standard by which the process, mechanisms and methodology is judged by the final outcome of the project.
• Converted or change information.
• Time limit constraint.
• The various specification or standard to be considerer

**Business Rules**
- Knowledge and expertise though various sources.
- Sampling techniques.

**Data**
- Primary and secondary data.
- Pilot survey.

In assessment to software development of CMM, the model look like this:

<table>
<thead>
<tr>
<th>What do other people give you?</th>
<th>What do you do?</th>
<th>What do you need to keep track of?</th>
<th>What rules apply?</th>
<th>What do you give other people?</th>
</tr>
</thead>
</table>

**The Need**: Improved software process capabilities.
How: improving through levels of process maturity by how they are defined, managed, Measured, controlled, and effective, this is done by institutionalizing policies, standards, Organization structures, and processes such as
- The SW_CMM[just CMM now]provides SW organizations guidance on what to do to gain control of their processes and how evolve towards a culture of S W engineering and management excellence(as in TQM,BPR)
- The CMM allows an organization to process improvement strategies by determined current state, weaknesses, and where to prioritotize
- Identify the Goals
- Commitments to perform-ensuring organizational support
- Abilities to perform-preconditions
- Activities performed-what-now how
- Measurement & Analyses-status/effectiveness
- Verifying Implementation-reviews/audits.

**CMM Guides Though**
- Have Policies stating what and who
- Have Resources committed
- Have Processes/Procedures documented
- Have Objective Evidence that processes are being followed by all
- Look for ways to improve

To survive, thrive and best the competition in today’s brutally competition world one has to manage the future. Managing the future means managing information. Today the difference between market leaders and followers, successfully companies and sick industries, is the way in which companies make use of information. Information means information generated by the company and the
information about the business environment. Companies generated huge amount of data-financial data, details about customers, purchase details, employee data and so on. Only an organization that makes the best possible use of this information can success. In order to manage the information, in order to deliver high quality information to the decision makers at the right time and in order to automate the process of data collection, collation and refinement, organization must make use of information technology as an ally.

**Objectives of the study**

- To access how CMM level certification delivers quality
- Why Indian customers buy services from non-certified companies. How does it affect the quality they receive?
- To analyze the different parameters or aspects of service delivery in the Indian context which is beyond the normal quality framework?
- To understand how ERP implementation happen.

**Scope of the study**

The scope of the study limited to
- Understanding of how CMM works
- Understandings of key practices and its impact on quality

**Research Methodology of the study**

The study is based on both primary data and secondary data

**A. Sampling frame**

Sampling frame includes all CMM levels certified companies and customers.

**B. Sampling procedure**

Since sample size is small an accessible, entire sampling frame is included in the study. That is, every CMM level certified companies who are engaged in delivering quality service.

**Limitations of the study**

The CMM level certified and non-certified companies and to identify the quality of the service they provide and its impact on final quality of delivery.

- To understand how CMM work.
- How service delivery beyond the quality framework.
- Assessment of service delivery beyond the quality framework.

**Tools employed in data collection**

The study will be conducted using personal interview method. Structured questionnaire will be used to conduct interview. The pilot Survey will be conducted with the help of questionnaire. Questionnaire will be refined after the pilot study. Feedback obtained from the pilot study shall be used to prepare a final questionnaire that is free from bias and any ambiguity.

**ORGANIZATIONAL HIERARCHY**

**Organizational Structure and Roles**

Although the CMM attempts to remain independent of specific organizational structures and models, it is necessary to express the practices in the CMM consistently using terminology related to organizational structure and roles, which may differ from that followed by any specific organization. The following sections describe the various concepts related to organizations, projects, and roles that are necessary for interpreting the key practices of the CMM.

**Organizational Roles**

A role is a unit of defined responsibilities that may be assumed by one or more individuals. The following descriptions of roles are frequently used in the key practices

- Manager
- Senior manager
Groups commonly referred to in the CMM are described below

- Software engineering group
- Software-related groups
- Software engineering process group
- System engineering group
- System test group
- Software quality assurance group
- Software configuration management group
- Training group

COLLECTION AND ANALYSIS OF PROCESS DATA

The key practices for the collection and analysis of process data evolve across the maturity levels.

- At Level 2, the data are primarily related to the size of the project's work products, effort, and schedule, and are defined, collected, and stored separately by each project. The data are shared between projects via informal mechanisms.
- At Level 3, each project has a defined software process tailored from the organization's standard software process. Data related to each project's defined software process are collected and stored in the organization's software process database. The data collected and stored may be different for each project, but the data are well defined within the organization's software process database.
- At Level 4, the organization defines a standard set of measurements based on the organization's standard software process. All projects collect this standard set of measurement data, as well as other project-specific data, and store them in the organization's software process database. The data are used by the projects to quantitatively understand and stabilize the process performance of the projects' defined software processes. They are also used by the organization to establish a process capability baseline for the organization's standard software process.
- At Level 5, data are used to select areas for technology and process improvements, plan these improvements, and evaluate the effects of these improvements on the organization's process capability.

FUTURE DIRECTIONS OF THE CMM

Achieving higher levels of software process maturity is incremental and requires a long-term commitment to continuous process improvement. Software organizations may take ten years or more to build the foundation for, and a culture oriented toward, continuous process improvement. Although a decade-long process improvement program is foreign to most U.S. companies, this level of effort is required to produce mature software Organizations. This time frame is consistent with experience from other industries, such as the U.S. automotive industry, that have achieved significant gains in process maturity [Gabor90].

Near-Term Activities
Tutorials and courses on the CMM are being presented at major conferences and seminars throughout the United States to ensure that the software industry has adequate awareness of the CMM and its associated products. CMM-based tools (e.g., the maturity questionnaire), software process assessment training, and software capability evaluation training are being developed and/or revised to incorporate the CMM. The near-term focus on CMM development activities will be oriented towards tailored versions of the CMM, such as a CMM for small projects and/or small organizations. CMM v1.1 is expressed in terms of the normative practices of large, government contracting organizations, and these practices must be tailored to the needs of organizations that differ from this template.

**Long-Term Activities**

During the next few years, the CMM will continue to undergo extensive testing through use in software process assessments and software capability evaluations. CMM-based products and training materials will be developed and revised as appropriate. The CMM is a living document that will be improved, but it is anticipated that CMM v1.1.

**CONCLUSION**

Continuous improvement applies to the maturity model and practices, just as it does to the software process. The potential impact of changes to the CMM on the software community will be carefully considered, but the CMM, the maturity questionnaire, and the software process assessment and software capability evaluation methods will continue to evolve as experience is gained with improving the software process. The SEI intends to work closely with industry, government, and academia in continuing this evolution.

The CMM provides a conceptual structure for improving the management and development of software products in a disciplined and consistent way. It does not guarantee that software products will be successfully built or that all problems in software engineering will be adequately resolved. The CMM identifies practices for a mature software process and provides examples of the state-of-the-practice (and in some cases, the state-of-the-art), but it is not meant to be either exhaustive or dictatorial. The CMM identifies the characteristics of an effective software process, but the mature organization addresses all issues essential to a successful project, including people and Technology, as well as process.

**REFERENCES**


4. Websites Visited
   - [www.teraquest.com](http://www.teraquest.com)
   - [www.sei.com/cmu](http://www.sei.com/cmu)
   - [www.projectperfect.com](http://www.projectperfect.com)
   - [www.google.com](http://www.google.com)