

# Comparative Study of Capability Maturity Model

Jasleen Kaur

Research Scholar, Amity University, Gurgaon, Haryana, India

## Abstract

Capability Maturity Model (CMM) broadly refers to a process improvement approach that is based on a process model. It was developed by the Software Engineering Institute (SEI).

CMM can be used to assess an organization against a scale of five process maturity levels. Each level ranks the organization according to its standardization of processes in the subject area being assessed. It provides integrated approach for process improvement while reducing redundancy, cost and complexity.

This paper is the comparative study of the Capability Maturity Model which covers purpose of using capability maturity model, staged representation, key process areas, structure of capability maturity model, behavior at five different levels, benefits, characteristics and difference between ISO 9000 and CMM.

## Keywords

?????? Missing ???????

## I. Introduction

Capability Maturity Model is not a software lifecycle model. It is a strategy for improving the software process. It was developed by Software Engineering Institute (SEI) of Carnegie-Mellon University in 1986.

## II. Purpose of Using CMM

The main purposes of using CMM is - To judge the maturity of software processes of an organization and to identify the key practices that are required to increase the maturity of these processes.

## III. Staged Representation

CMM describes five evolutionary stages in which an organization manages its processes. The Stage representation of CMMI is as shown in fig. 1.

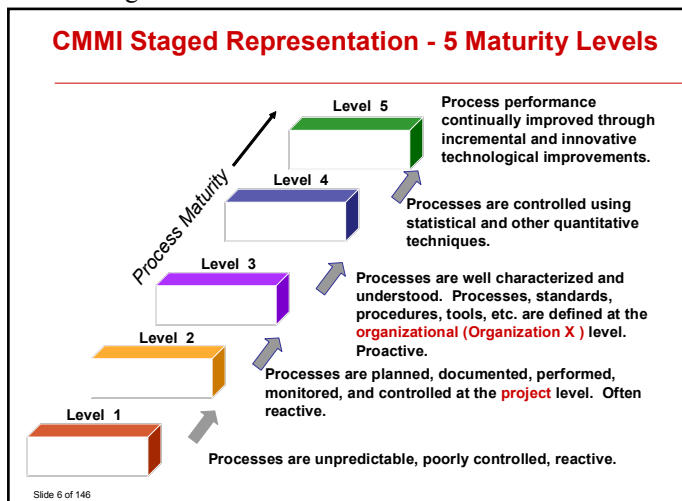


Fig. 1: Stage Representation of CMMI

## IV. Key Process Areas

It describes the function that must be present to satisfy good practice at particular level. Various characteristics are- Goal, commitment, Ability, Activity, Measurement, verification.

## V. CMM Structure

The CMM is composed of five maturity levels. With the exception

of Level 1, each maturity level is composed of several key process areas. The features specify the key practices that, when collectively addressed, accomplish the goals of the key process area. The goals of each key process area summarize its key practices and can be used in determining whether an organization or product has effectively implemented the key process area. This structure of the CMM is illustrated in table 1.

Table 1: Structure of CMM

LEVEL	KEY PROCESS AREAS
5 Optimizing	Defect Prevention Technology Innovation Process Change Management
4 Managed	Quantitative Process Management SW Quality Management
3 Defined	Organisation Process Focus Organisation Process Definition Peer Reviews Training Program Intergroup Coordination SW Product Engineering Integrated SW Management
2 Repeatable	SW Project Planning SW Project Tracking SW Subcontract Management SW Quality Assurance SW Configuration Management Requirements Management
1 Initial	

## VI. Behaviour At Five Levels

CapabilityMaturityModel specifies an increasing series of levels that a software development organization can be at. The higher the level, the better the software development process.

Behaviors at the Five Levels		
Maturity Level	Process Characteristics	Behaviors
5 Optimizing	Focus is on continuous quantitative improvement	Focus on "fire prevention"; improvement anticipated and desired, and impacts assessed.
4 Quantitatively Managed	Process is measured and controlled	Greater sense of teamwork and inter-dependencies
3 Defined	Process is characterized for the organization and is proactive	Reliance on defined process. People understand, support and follow the process.
2 Managed	Process is characterized for projects and is often reactive	Over reliance on experience of good people – when they go, the process goes. "Heroics."
1 Initial	Process is unpredictable, poorly controlled, and reactive	Focus on "fire fighting"; effectiveness low – frustration high.

Slide 10 of 146

Fig. 2: Behavior at Five Levels

**VII. Characteristics**

Maturity levels consist of a predefined set of process areas. The following sections describe the characteristics of each maturity level.

**FOR LEVEL 1: INITIAL**

1. No key processes
2. Weak management practices
3. Poorly controlled commitments
4. processes are ad hoc
5. practices are sacrificed for schedule
6. Practitioners resist discipline
7. Results are unpredictable

**FOR LEVEL 2: REPEATABLE**

1. Policies for managing a software project and procedures to implement those policies are established.
2. Project activities are planned and followed.

**FOR LEVEL 3: DEFINED**

1. Organisation focus on process definition and process usage
2. Process management infrastructure exists
3. Process work is part of organisation’s business
4. Organisational SW process exists
  - collection of best practices
  - tailored for each project
  - integrates different processes
  - basis for comparable measurement results
5. Training plans are created and followed (project and organisation levels)
6. More systematic technical coordination between different project groups

**FOR LEVEL 4: MANAGED**

1. Organization sets quality goals for both software products and processes.
2. Predictability is improved

**FOR LEVEL 5: OPTIMIZED**

1. Continuous process improvement in place
2. Processes and technology are continuously evaluated
3. Individuals are empowered to improve their processes
4. The causes of defects are eliminated as part of preventive quality work

5. New technologies can be utilised effectively to improve process capability

**VIII. Benefits of CMM**

1. Helps forge a shared vision of what software process improvement means for the organization
2. Defines set of priorities for addressing software problems
3. Supports measurement of process by providing framework for performing reliable and consistent appraisals
4. Provides framework for consistency of processes and product

**IX. Problems With CMM**

1. It is a goal, not a method
2. Being used just as stamp of approval
3. Doesn’t help in a crisis
4. Only for repetitive tasks
5. CMM is based on re-using past results for future software projects.

**X. Difference Between ISO 9000 and CMM**

In general, the CMM and ISO 9000 address similar issues and have the common concern of quality and process management. However, the genesis of each framework is distinctly disparate. The ISO focus is the customer-supplier relationship, attempting to reduce a customer’s risk in choosing a supplier. In contrast, the CMM strength is the attention on the software supplier to improve its internal processes to achieve a higher quality product for the benefit of the customer. The main difference between the two systems lies in their respective purposes: ISO 9001 specifies a minimal acceptable quality level for software processes, while the CMM establishes a framework for continuous process improvement. The differences between CMM and ISO are illustrated in table 2.

Table 2: Differences Between ISO and CMM

ISO 9000	Capability Maturity Model
It applies to any type of industry.	CMM is specially developed for software industry
ISO 9000 addresses corporate business process	CMM focuses on the software Engineering activities.
ISO 9000 specifies minimum requirement	CMM gets nto technical aspect of software engineering.
ISO 9000 restricts itself to what is required	It suggests how to fulfill the requirements.
ISO 9000 provides pass or fail criteria.	It provides grade for process maturity.
ISO 9000 has no levels.	CMM has 5 levels: Initial Repeatable Defined Managed Optimization

Certain process elements that are in ISO are not included in CMM like: 1. Contract management 2. Purchase and customer supplied components 3. Personal issue management 4. Packaging ,delivery, and installation management	Similarly other process in CMM are not included in ISO 9000 1. Project tracking 2. Process and technology change management 3. Intergroup coordinating to meet customer’s requirements. 4. Organization level process focus, process development and integrated management.
ISO 9000	Capability Maturity Model
ISO 9000 does not specify sequence of steps required to establish the quality system.	It reconnects the mechanism for step by step progress through its successive maturity levels.

**References**

[1] K.K. Aggarwal, *Software Engineering, New age international*,2006  
 [2] Pressman, *Software Engineering,TataMcGraw-Hill*,2008  
 [3] Pankaj Jalote, *Software Engineering,Narosa*,2010  
 [4] [www.oracle.com/.../oracle-wp-cloud-maturity-model-r3-01934.pdf](http://www.oracle.com/.../oracle-wp-cloud-maturity-model-r3-01934.pdf)  
 [5] Rajib Mall, *Fundamentals of software engineering, Prentice hall of india*,2009  
 [6] Ian Sommerville, *Software Engineering, Addison Wesley*,2007  
 [7] [www.managewithoutthem.com/blog/?p=460](http://www.managewithoutthem.com/blog/?p=460)  
 [8] Mathias Ekstedt, *Theory of software engineering, Industrial info system*,2008  
 [9] Ivan Marsic, *Software Engineering, Rutgers University*,2008  
 [10] Sue Conger, *New Software engineering, Global text project*,2008  
 [11] [home.etf.rs/.../Capability%20Maturity%20Model%20](http://home.etf.rs/.../Capability%20Maturity%20Model%20)