

Requirements Management under the Capability Maturity Model

Capability Maturity Model Background

In 1986, the Software Engineering Institute (SEI) at Carnegie Mellon, with funding from the Department of Defense, began developing a process maturity framework to help developers improve their software process. The effort was intended to provide the federal government with a method for assessing the capability of its software contractors. By 1991, this framework became known as the Capability Maturity Model (CMM). The current version of the CMM, Version 1.1, is being used by various government agencies to qualify software subcontractors and is also being used by a large number of commercial firms to improve and measure their software process.

Capability Maturity Model Framework

The CMM provides a framework for continuous process improvement based on many small organized evolutionary steps defined in five maturity levels that lay successive foundations. These five maturity levels define an ordinal scale for measuring the maturity of an organization's software process and for evaluating its software process capability. They also help an organization prioritize its improvement efforts.

A maturity level is a well-defined evolutionary plateau toward achieving a mature software process. Each maturity level comprises a set of process goals that, when satisfied, stabilize an important component of the software process. Achieving each level of the maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organization.

Organizing the CMM into the five levels shown in Figure 1 prioritizes improvement actions for increasing software process maturity. The labeled arrows indicate the type of process capability being institutionalized by the organization at each level of the maturity framework.

Capability Maturity Model (CMM) - Overview

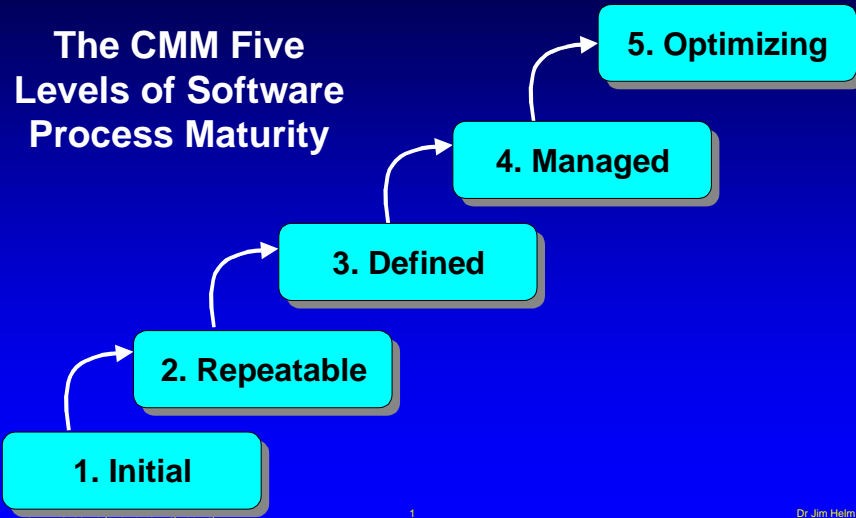


Figure 1 Capability Maturity Model Overview

The five levels are shown in Figure 2 and can be briefly described as:

1. Initial

The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.

2. Repeatable

Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.

3. Defined

The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

4. Managed

Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.

5. Optimizing

Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

Characterizations of the Five Maturity Levels

- | | |
|---------------|--|
| 1) Initial | Ad hoc, even chaotic. Success depends on individual heroics. |
| 2) Repeatable | Basic processes established to track cost, schedule, and functionality. Earlier successes can be repeated. |
| 3) Defined | A process is documented, standardized, and integrated into the organization. All projects use a tailored version of the process. |
| 4) Managed | Measures of the process and product quality are collected. Process and products are quantitatively understood and collected. |
| 5) Optimizing | Continuous process improvement by quantitative feedback and introducing innovative technologies. |

SWEN 5130 Requirements Engineering

1

Dr. Jim Helm

Figure 2 Characterization of the Five Maturity Levels

These five levels reflect the fact that the CMM is a model for improving the capability of software organizations. The priorities in the CMM, as expressed by these levels, are not directed at individual projects. A troubled project might well prioritize its problems differently from the taxonomy given by the CMM. Its solutions might be of limited value to the rest of the organization, because other projects might have different problems or be unable to take advantage of its solutions because they lack the necessary foundation to implement the solutions. The CMM focuses on processes that are of value across the organization. .

Behavioral Characterization of the Maturity Levels

Maturity Levels 2 through 5 can be characterized through the activities performed by the organization to establish or improve the software process, by activities performed on each project and by the resulting process capability across projects.

Level 1: The Initial Level

At the Initial Level 1, the organization typically does not provide a stable environment for developing and maintaining

software. Over commitment is a characteristic of Level 1 organizations, and such organizations frequently have difficulty making commitments that the staff can meet with an orderly engineering process, resulting in a series of crises. During a crisis, projects typically abandon planned procedures and revert to coding and testing. Success depends on having an exceptional manager and a seasoned and effective software team. Occasionally, capable and forceful software managers can withstand the pressures to take shortcuts in the software process; but when they leave the project, their stabilizing influence leaves with them. Even a strong engineering process cannot overcome the instability created by the absence of sound management practices. In spite of this ad hoc, even chaotic, process, Level 1 organizations frequently develop products that work, even though they may exceed the budget and schedule. Success in Level 1 organizations depends on the competence and heroics of the people in the organization and cannot be repeated unless the same competent individuals are assigned to the next project. Thus, at Level 1, capability is a characteristic of the individuals, not of the organization.

Level 2: The Repeatable Level

At the Repeatable Level 2, policies for managing a software project and procedures to implement those policies are established. Planning and managing new projects is based on experience with similar projects. Process capability is enhanced by establishing basic process management discipline on a project by project basis. Projects implement effective processes that are defined, documented, practiced, trained, measured, enforced, and improvable.

Projects in Level 2 organizations have installed basic software management controls. Realistic project commitments are made, based on the results observed on previous projects and on the requirements of the current project. The software managers for a project track software costs, schedules, and functionality; problems in meeting commitments are identified as they arise. Software requirements and the work products developed to satisfy them are baselined, and their integrity is controlled. Software project standards are defined, and the organization ensures they are faithfully followed. The software project works with its subcontractors, if any, to establish an effective customer-supplier relationship.

Processes may differ between projects in a Level 2 organization. The organizational requirement for achieving Level 2 is that there

be organization-level policies that guide the projects in establishing the appropriate management processes.

The software process capability of Level 2 organizations can be summarized as disciplined because planning and tracking of the software project is stable and earlier successes can be repeated. The project's process is under the effective control of a project management system, following realistic plans based on the performance of previous projects.

Level 3: The Defined Level

At the Defined Level 3, a standard process (or processes) for developing and maintaining software is documented and used across the organization. Referred to throughout the CMM as the organization's standard software process, this standard process includes both software engineering and management processes and integrates them into a coherent whole. Processes established at Level 3 (and changed, as appropriate) to help the software managers and technical staff perform more effectively. The organization exploits effective software engineering practices when standardizing its software processes. A group within the organization is assigned responsibility for software process activities. An organization-wide training program is implemented to ensure that the staff and managers have the knowledge and skills required fulfilling their assigned roles.

Projects tailor the organization's standard software process to develop their own defined software process, which accounts for the unique characteristics of the project. This tailored process is referred to in the CMM as the project's defined software process. It is the process used in performing the project's activities. A defined software process contains a coherent, integrated set of well-defined software engineering and management processes. A well-defined process can be characterized as including readiness criteria, inputs, standards and procedures for performing the work, verification mechanisms (such as peer reviews), outputs, and completion criteria. Because the software process is well defined, management has good insight into technical progress on the project.

The software process capability of Level 3 organizations can be summarized as standard and consistent because both software engineering and management activities are stable and repeatable. Within established product lines, cost, schedule, and functionality are under control, and software quality is tracked. This process capability is based on a common, organization-wide

understanding of the activities, roles, and responsibilities in a defined software process.

Level 4: The Managed Level

At the Managed Level 4, the organization sets quantitative quality goals for both software products and processes. Productivity and quality are measured for important software process activities across all projects as part of an organizational measurement program. An organization-wide software process database is used to collect and analyze the data available from the projects' defined software processes. Software processes are instrumented with well-defined and consistent measurements. These measurements establish the quantitative foundation for evaluating the projects' software processes and products.

Projects achieve control over their products and processes by narrowing the variation in their process performance to fall within acceptable quantitative boundaries. Meaningful variations in process performance can be distinguished from random variation (noise), particularly within established product lines. The risks involved in moving up the learning curve of a new application domain are known and carefully managed.

The software process capability of, Level 4 organizations can be summarized as being quantifiable and predictable because the process is measured and operates within quantitative limits. This level of process capability allows an organization to predict trends in process and product quality within the quantitative bounds of these limits. Because the process is both stable and measured, when some exceptional circumstance occurs, the "special cause" of the variation can be identified and addressed. When the pre-defined limits are exceeded, actions are taken to understand and correct the situation. Software products are of predictably high quality.

Level 5: The Optimizing Level

At the Optimizing Level 5, the entire organization is focused on continuous process improvement. The organization has the means to identify weaknesses and strengthen the process proactively, with the goal of preventing the occurrence of defects. Data on the effectiveness of the software process are used to perform cost/benefit analyses of new technologies and proposed changes to the organization's software process. Innovations that exploit the

best software engineering practices are identified and transferred throughout the organization.

Software teams in Level 5 organizations analyze defects to determine their causes. They evaluate software processes to prevent known types of defects from recurring and disseminate lessons learned throughout the organization.

Chronic waste, in the form of rework, can be found in any system simply due to random variation. Organized efforts to remove waste result in changing the system, that is, in improving the process by changing "common causes" of inefficiency to prevent the waste from occurring. While this is true of all the maturity levels, it is the focus of Level 5.

The software process capability of Level 5 organizations can be characterized as continuously improving because Level 5 organizations are continuously striving to improve the range of their process capability, thereby improving the process performance of their projects. Improvements occur by incremental advancements in the existing process and by innovations using new technologies and methods. Technology and process improvements are planned and managed as ordinary business activities.

CMM Key Process Areas Related to Requirements Management

The key process areas identify the issues that must be addressed to achieve a maturity level. Figure 3 CMM Key Process Areas shows that in Level 2 Repeatable, requirements management is a key process.

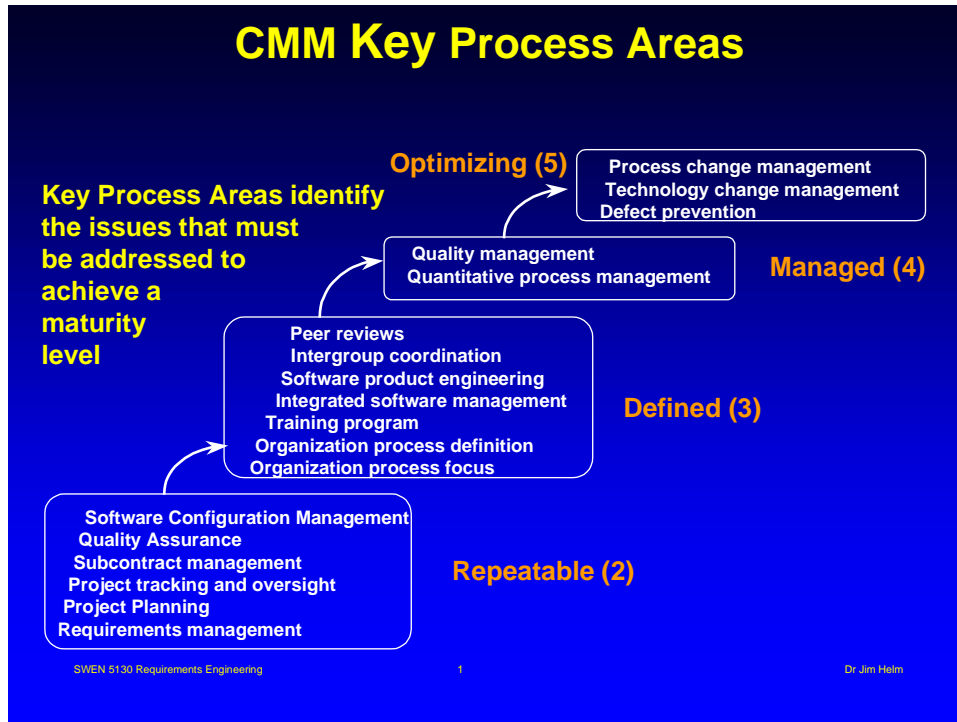


Figure 3 CMM Key Process Areas

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 projects. Two goals of the RM key process area are:

1. Software requirements are controlled to establish a baseline for software engineering and management use.
2. Software plans, products, and activities are kept consistent with the system requirements allocated to software.

The software teams and management make a commitment that the project follows a written organizational policy that:

1. The requirements are documented.
2. Managers and systems test, system design, quality assurance, etc review requirements.
3. Plans, work products and activities are changed to be consistent with changes to requirements.

The software teams and management ensure an ability to perform the following:

- Ability 1 - Responsibilities are established for analyzing requirements
- Ability 2 - The allocated requirements are documented
 1. Technical (what the system does) and non-technical (milestones, schedule, costs) requirements are documented
 2. Acceptance criteria are established and documented
- Ability 3 - Resources and funding are provided for managing requirements
 1. Trained individuals are assigned to manage requirements
 2. Appropriate support tools are provided
- Ability 4 - Members of the group are trained to perform their requirements management activities

The activities performed by the groups and overseen by management are:

- Activity 1
The software-engineering group reviews the allocated requirements before they are incorporated into the system baseline.
- Activity 2
The software group uses the allocated requirements as the basis for plans and activities. Requirements are managed and controlled.
- Activity 3
Changes are reviewed and incorporated into the project.

Measurements are made and analyzed to determine the status of activities for managing the allocated requirements. Examples of the types of measurements include:

- Examples
 1. Measures of the process itself
 2. Status of each of the allocated requirements (proposed, open, approved, incorporated)
 3. Change activity, cumulative number of changes

The Capability Maturity Model is also involved with the implementation of verification activities performed by a Verification and Validation team. The following are three verification activities:

1. Activities for managing requirements are periodically reviewed with senior management.
2. Activities are reviewed by project managers on a periodic and event-driven basis
3. Software quality assurance reviews, audits, activities and work product for managing requirements.

At the Defined Level 3 of CMM the following summarizes the requirements management process:

- Software requirements are developed, maintained, documented, and verified by systematically analyzing the allocated requirements
- Effective methods for analysis are used that are feasible and appropriate to implement, consistent, clearly stated, testable and complete
- Each requirement is analyzed to verify it can be tested. Methods for verifying are identified and documented.
- Requirements traceability supports analysis, coverage, and change management
- The software requirements document is reviewed and approved with the customers and end users if appropriate.
- The software requirements document is placed under configuration management.

Conclusion

The Software Engineering Institute's Capability Maturity Model process classifies software processes as initial, repeatable, defined, managed and optimizing. The CMM identifies key process areas that should be used at each of the levels. This unit introduced and discussed the requirements management key process area.