Software process technology

Teamwork

**GROUP MEMBERS**

**班 级：151 班 级：**

**专业：Software Engineering 专业：**

**学号：201532120120 学号：**

**姓名：胡明志 姓名：**

**班 级：151 班 级：**

**专业：Software Engineering 专业：**

**学号： 学号：**

**姓名： 姓名：**

**Category**

[Chapter 1 The Process Maturity Framework 1](#_Toc17276)

[Chapter2 The Five Levels of Software Process Maturity 2](#_Toc29081)

[Chapter 3 Operational Definition of the capability maturity model 4](#_Toc19890)

[Chapter4 Using the CMM 7](#_Toc4553)

[Chapter5 Future Directions of the CMM 9](#_Toc22862)

[Conclusion 9](#_Toc12436)

[Summary： 10](#_Toc9647)

**Chapter 1 The Process Maturity Framework**

**1.1 Immature Versus Mature Software Organizations**

In a mature organization, managers monitor the quality of the software products and customer satisfaction. There is an objective, quantitative basis or judging product quality and analyzing problems with the product and process.

**1.2 Fundamental Concepts Underlying Process Maturity**

A software process can be defined as a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products.

Software process capability describes the range of expected results that can be achieved by following a software process.

Software process performance represents the actual results achieved by following a software process.

**1.3 Overview of the Capability Maturity Model**

It 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.

**Chapter2 The Five Levels of Software Process Maturity**

**2.1 The five level of software process maturity**

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

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.

**2.2 Understand the five level**

**2.2.1 initial level**

Although Level 1 organizations are frequently characterized as having ad hoc, even chaotic, processes, they frequently develop products that work, even though they may be over the budget and schedule. Success in Level 1 organizations depends on the competence and heroics of the people in the organization.

**2.2.2 Repeatable and Defined Levels**

To achieve Level 2, management must focus on its own processes to achieve a disciplined software process.

Level 3 builds on this project management foundation by defining,integrating, and documenting the entire software process.

**2.2.3 Understanding the Managed and Optimizing Levels**

The first responsibility, and the focus of Level 4, is process control. The software process is managed so that it operates stably within a zone of quality control.

The second responsibility, and the focus of Level 5, is continuous process improvement. The software process is changed to improve quality, and the zone of quality control moves. A new baseline for performance is established that reduces chronic waste. The lessons learned in improving such a process are applied in planning future processes. This is where the concept of addressing common causes of variation comes to the fore.

.

**2.3 Visibility Into the Software Process**

At Level 1, the software process is an amorphous entity – a black box – and visibility into the project's processes is limited.

At Level 2, the customer requirements and work products are controlled, and basic project management practices have been established.

At Level 3, the internal structure of the boxes, i.e., the tasks in the project's defined software process, is visible.

At Level 4, the defined software processes are instrumented and controlled quantitatively.

At Level 5, new and improved ways of building the software are continually tried, in a controlled manner, to improve productivity and quality.

**2.4 Process Capability and the Prediction of Performance**

First, as maturity increases, the difference between targeted results and actual results decreases across projects.

Second, as maturity increases, the variability of actual results around targeted results decreases.

Third, targeted results improve as the maturity of the organization increases.

**2.5 Skipping Maturity Levels**

Skipping levels is counterproductive because each level forms a necessary foundation from which to achieve the next level. The CMM identifies the levels through which an organization must evolve to establish a culture of software engineering excellence. Processes without the proper foundation fail at the very point they are needed most – under stress – and they provide no basis for future improvement.

The process improvement effort should focus on the needs of the organization in the context of its business environment. The ability to implement processes from higher maturity levels does not imply that maturity levels can be skipped.

**Chapter 3 Operational Definition of the capability maturity model**

**3.1 Internal Structure of the Maturity Levels**

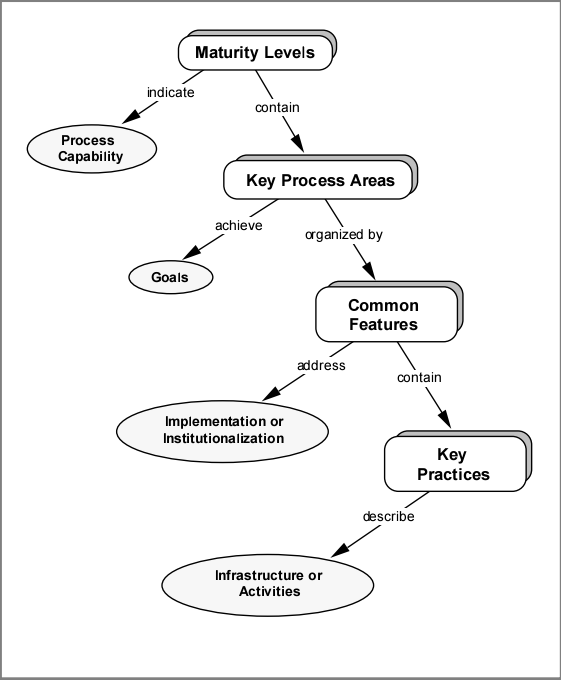


Figure 3.1 The CMM Structu

**3.2 Maturity Levels**

A maturity level is a well-defined evolutionary plateau toward achieving a mature software process

**3.3 Key Process Areas**

Each maturity level is decomposed into several key process areas 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.

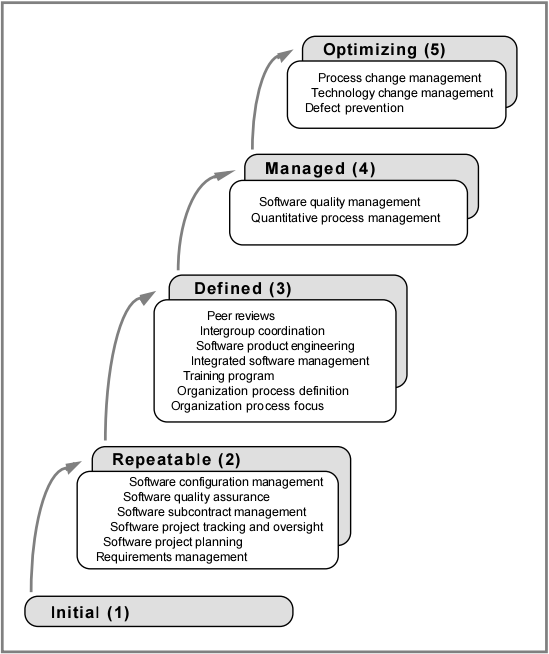


Figure 3.2 The Key Process Areas by Maturity Level

For instance, many of the project estimating capabilities described in the Software Project Planning key process area at Level 2 must evolve to handle the additional project data available at Levels 3, 4, and 5. Integrated Software Management at Level 3 is the evolution of Software Project Planning and Software Project Tracking and Oversight at Level 2 as the project is managed using a defined software process.

**3.4 Common Features**

The common features are attributes that indicate whether the

implementation and institutionalization of a key process area is effective,

repeatable, and lasting.

**3.5 Key Practices**

The key practices describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area.

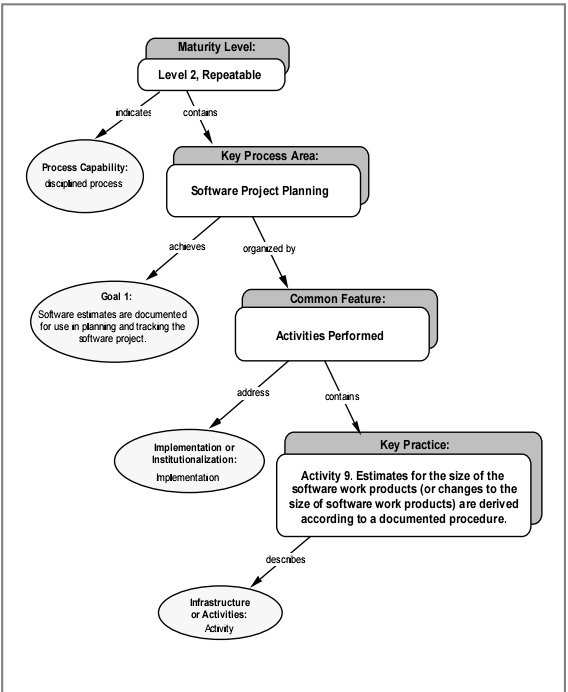


Figure 3.3 Building the CMM Structure: An Example of a Key Practice

The key practices should be interpreted rationally to judge whether the goals of the key process area are effectively, although perhaps differently, achieved.

**Chapter4 Using the CMM**

**4.1 Software Process Assessment and Software**

Capability Evaluation Methods Software process assessments focus on identifying improvement priorities within an organization's own software process.

Software capability evaluations are focused on identifying the risks associated with a particular project or contract for building high-quality software on schedule and within budget.

The CMM establishes a common frame of reference for performing software process assessments and software capability evaluations.

The first step in is to select a team.

The second step is to have representatives from the site to be assessed or evaluated complete the maturity questionnaire and other diagnostic instruments.

The team is now ready to visit the site being assessed or evaluated (step 4).

Beginning with the results of the response analysis, the team conducts interviews and reviews documentation to gain an understanding of the software process followed by the site.

At the end of the on-site period, the team produces a list of findings (step 5) that identifies the strengths and weaknesses of the organization's software process.

Finally, the team prepares a key process area profile (step 6) that shows the areas where the organization has, and has not, satisfied the goals of the key process areas.

In summary, the software process assessment and software capability

evaluation methods both:use the maturity questionnaire as a springboard for the on-site visit,use the CMM as a map that guides the on-site investigation,develop findings that identify software process strengths and weaknesses in terms of the key process areas in the CMM,derive a profile based on an analysis of the satisfaction of the goals within the key process area, and present their results, to the appropriate audience, in terms of findings and a key process area profile.

**4.2 Differences Between Software Process**

Assessments and Software Capability Evaluations

In spite of these similarities, the results of a software process assessment or software capability evaluation may differ, even on successive applications of the same method.

Software process assessments and software capability evaluations differ in motivation, objective, outcome, and ownership of the results.

Software process assessments are performed in an open, collaborative environment.

Software capability evaluations, on the other hand, are performed in a more audit-oriented environment.

This does not mean, however, that the results of software process assessments and software capability evaluations should not be comparable.

**4.3 Other Uses of the CMM in Process Improvement**

For software engineering process groups or others trying to improve their software process, the CMM has specific value in the areas of action planning, implementing action plans, and defining processes.

The software engineering process group must next determine which process improvements are needed, how to effect the change, and obtain the necessary buy-in.

**Chapter5 Future Directions of the CMM**

**5.1 CMM future directions**

CMM implementation of a higher level of software process maturity is gradual and requires long-term commitment to continuous improvement processes. Software organizations may take more than ten years to lay the foundation for a continuous process improvement and a culture oriented basis.

**5.2 What the CMM Does Not Cover**

The CMM does not currently address expertise in particular application domains,advocate specific software technologies, or suggest how to select, hire,motivate, and retain competent people. Although these issues are crucial to a project's success, some of these issues have been analyzed in other contexts. They have not, however, been integrated into the CMM.The CMM was specifically developed to provide an orderly, disciplined framework within which to address software management and engineering process issues.

**5.3 Near-Term Activities**

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.

**5.4 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 will remain the baseline until at least 1996. This provides an appropriate and realistic balance between the needs for stability and for continued improvement.

**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.

**Summary：**

The CMM is a framework representing a path of improvements recommended for software organizations that want to increase their software process capability.

CMM refers to “The capacity maturity model”,the whole name in English is Capability Maturity Model for Software.It is a description of each phase of the software organization's practice in defining, implementing, measuring, controlling, and improving its software process.The core of CMM is count the software development as a process, and according to this principle of process monitoring and research on software development and maintenance, in order to make it more scientific and standardized, so that enterprises can better achieve business goals.

There are at least four uses of the CMM that are supported:

1. Assessments teams will use the CMM to identify strengths and weaknesses in the organization.

2. Evaluation teams will use the CMM to identify the risks of selecting among different contractors for awarding business and to monitor contracts.

3. Managers and technical staff will use the CMM to understand the activities necessary to plan and implement a software process improvement program for their organization.

4. Process improvement groups, such as an SEPG, will use the CMM as a guide to help them define and improve the software process in their organization.