CMM for Small Organizations in Iran

Z. Abbassi
Computer Engineering Department, Sharif University of Technology
z_abbasi@ce.sharif.edu

ABSTRACT

CMM (Capability Maturity Model) for software developed by the Software Engineering Institute has had a major influence on software process and quality improvement around the world, especially in the United States and Europe. But until 2002 no software organization in Iran had its certificate. This and similar problems have put Iran away from the competitive business world. The software CMM is focused on large projects and large organizations. 75% of organizations in Iran have fewer than 20 staff. So, we define organization who have between 5 and 20 staff as small organizations. It seems that CMM must become customized to small organizations both in scale and the application procedure. This paper proposes a five step model according to CMM and also appropriate for small organizations. It is concluded that, CMM as a successful model with the basic idea of continuous process improvement, can be applied to small organizations with making some changes.

Keywords: Capability Maturity Model, Software CMM, Small Organizations, Software Process, Process Improvement.

1. INTRODUCTION

Choosing an appropriate model and ensuring its compatibility with an organization's characteristics and business strategy is the first step of the process of founding a mature organization [1]. CMM (Capability Maturity Model) for software developed by the Software Engineering Institute has had a major influence on software process and quality improvement around the world, especially in the United States and Europe [2]. But until 2002 no software organization in Iran had its certificate. The experience of India shows that this standard causes higher confidence for foreign corporations. Most organizations in Iran, still don't use up-to-date systems, especially integrated systems. Great weakness is observed in project management of large scale software projects. These problems might cause great delays and they can even make a project fail. All these facts have put Iran away from the competitive business world [3]. The software CMM is focused on large projects and large organizations [4]. 75% of organizations in Iran have fewer than 20 staff [5]. It seems that CMM must become customized to small organizations both in scale and the application procedure.

2. An Overview of CMM

The Capability Maturity Model for Software (CMM) is a framework that describes the key elements of an effective software process. The CMM describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process.

The CMM covers practices for planning, engineering, and managing software development and maintenance. When followed, these key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality.

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 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 [7].

2.1. Level 1 - The Initial Level

At the Initial Level, the organization typically does not provide a stable environment for developing and maintaining software. When an organization lacks sound management practices, the benefits of good Software engineering practices are undermined by ineffective planning and reaction-driven commitment systems. During a crisis, projects typically abandon planned procedures and revert to coding and testing. Success depends entirely 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. The software process capability of Level 1 organizations is unpredictable because the software process is constantly changed or modified as the work progresses (i.e., the process is ad hoc). Schedules, budgets, functionality, and product quality are generally unpredictable. Performance depends on the capabilities of individuals and varies with their innate skills, knowledge, and motivations. There are few stable software processes in evidence, and performance can be predicted only by individual rather than organizational capability [7].

2.2. Level 2 - The Repeatable Level

At the Repeatable Level, 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. An objective in achieving Level 2 is to institutionalize effective management processes for software projects, which allow organizations to repeat successful practices developed on earlier projects, although the specific processes implemented by the projects may differ. An effective process can be characterized as practiced, documented, enforced, trained, measured, and able to improve.
Projects in Level 2 organizations have installed basic software management controls. Realistic project commitments are 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 when 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 a strong customer-supplier relationship.

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 [7]. Brief 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.
- The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project.
- 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.
- 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.
- 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 [8].

2.3. Level 3 - The Defined Level

At the Defined Level, the standard process for developing and maintaining software across the organization is documented, including both software engineering and management processes, and these processes are integrated into a coherent whole. This standard process is referred to throughout the CMM as the organization's standard software process. Processes established at Level 3 are used (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. There is a group that is responsible for the organization's software process activities, e.g., a software engineering process group, or SEPG . An organization-wide training program is implemented to ensure that the staff and managers have the knowledge and skills required to fulfill 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. 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 all projects.

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 [7]. Brief 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 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.
- The purpose of Training Program is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently.
- 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.
- 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.
- 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.

- The purpose of Peer Reviews is to remove defects from the software work products early and efficiently [8].

2.4. Level 4 - The Managed Level

At the Managed Level, 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 at Level 4. 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 predictable because the process is measured and operates within measurable limits. This level of process capability allows an organization to predict trends in process and product quality within the quantitative bounds of these limits. When these limits are exceeded, action is taken to correct the situation. Software products are of predictably high quality [7]. Brief descriptions of each of the key process areas for Level 4 are given below:

- The purpose of Quantitative Process Management is to control the process performance of the software project quantitatively.
- 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 [8].

2.5. Level 5 - The Optimizing Level

At the Optimizing Level, 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 is 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 project teams in Level 5 organizations analyze defects to determine their causes. Software processes are evaluated to prevent known types of defects from recurring, and lessons learned are disseminated to other projects.

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. Improvement occurs both by incremental advancements in the existing process and by innovations using new technologies and methods [7]. Brief 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 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 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 [8].

3. Applying CMM to small organizations

Some references [6], define a small organization as fewer than 50 software developers and a small project as fewer than 20 developers. Also, an organization with fewer than 20 people is considered extra small. Considering the software organizations in Iran and number of their staff, we define small organization those who have between 5 and 20 staff.

A general overview of SW-CMM shows that there are 52 goals, 316 key practices related to 18 key process areas. Also it proposes more than 20 roles, responsibilities and tasks (Table 1). Using SW-CMM in small organizations, where usually their prime goal is to survive, seems not to be reasonable. An obvious reason is that there is not a consistency between the number of roles, responsibilities and tasks to be done with the number of staff available in a small organization. Most of the time there won't even be enough people to fill these roles and take part in groups. But, if the organization seems to be chaos and does not have a high performance, even in the case of small organizations, an appropriate solution must be found. If not, failure would be very likely. Considering the success of CMM in many organizations around the world and its focus on continuous process improvement, if CMM can be scaled down in size and other matters, it would be very helpful. By use of CMM the organization can also grow and extend its scope of work and in this case, it won't be a small organization any more.

Table 1.

	Roles Defined in SW-CMM		Roles Defined in SW-CMM
1	Senior Manager	12	Software Quality Assurance Group
2	Project Manager	13	Contract Management Group
3	Software Project Manager	14	Software Subcontract Management Group
4	First-line Software Manager	15	Document Support Group
5	Software Task Leader	16	Hardware Engineering Group
6	Software Engineering Staff	17	Sales and Marketing Group
7	Software Engineering Group	18	Requirements Management Group
8	Software Process Engineering Group	19	Software Estimating Group
9	Software Configuration Control Board	20	Technology Change Management
10	System Test Group	21	Training Group
11	Systems Engineering Group	22	Software Configuration Management Group

4. Software Organizations' Situation in Iran

According to statistics published by High Council of Informatics[5], which all informatics organization are under its supervision, 13% of organizations have fewer than 5 people; 30% between 6 and 9; 32% between 10 and 19; 11% between 20 and 29; 5% between 30 and 39; 2% between 40 and 49; 2% between 50 and 59; 1% between 70 and 79; 1% between 80 and 89 and 2% more than 90 people working in them. So it can be concluded that 75% of the organizations in Iran have fewer than 20 staff, hence they will be in the small organizations' category (Figure 1 & 2).

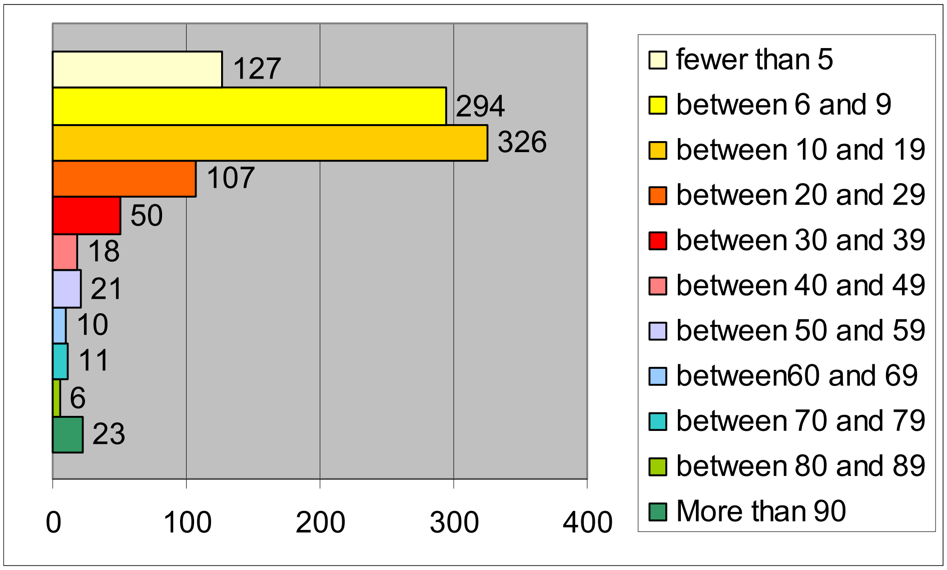

Figure 1.

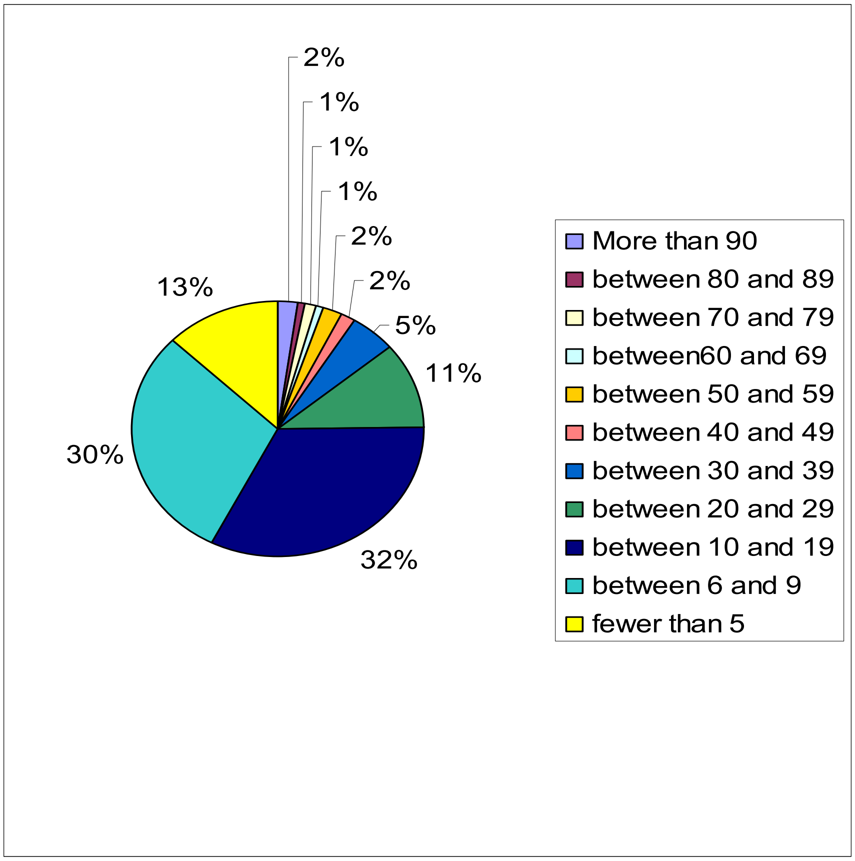

Figure 2.

5. Solution Proposal

To make CMM applicable to small organizations a 6 step plan is proposed:

1) First of all, the language of the organization must be interpreted to the terminology of CMM. It means that there may be some roles or groups in both the organization and CMM but with different names. Also some roles may exist in the organization but all done by a single person or even there may be a group defined in CMM only consisting of one person. These different facts must be mapped to each other through a defined process. All the information gained in this step must be documented for further use in other steps.

2) With a high probability there are roles and groups defined in CMM which are not present in the organization. Some of them are naturally omitted, for example if a small organization does not work with contractors or subcontractors, having a group to manage them are nonsense. But, some roles and groups cannot be omitted from CMM, which should be identified. Some activities such as customer communication, requirements management and documenting, planning, documenting processes and engineering are necessary for any software organization. Such activities may have different implementations in small and large organizations. These activities must be identified in an organization for an appropriate plan. Activities which are likely to emerge in the organization must also be mentioned. Documentation is required in this step, too.

3) Those activities which are found necessary to be done, according to both CMM and organization's structure, should be prioritized in their own level. A growing organization must do this analysis only for level 2, because after the implementation it may grow and

change into a large organization. But, an organization which does not have the tendency of growing can perform the analysis completely for all levels.

Now that the activities are determined, each must be defined at an appropriate level of detail. They also must be logically ordered to ensure a smooth evolution between tasks. Some tasks may be performed in parallel while others must follow one another sequentially. Task sequence depends on which task produce deliverables needed in other tasks, when critical resources are available or constraints are placed by the manager(s). The appropriate sequence of activities is the outcome of this step.

4) According to the sequence gained from step 3 and the organization's financial and workforce situation, a detailed plan and schedule must be established. In this plan each key process area will be seen as a maturity sub-level. After achieving all (achievement of all) key process areas of a level, the organization will be considered mature up to it. So, the outcome of this level will be a plan and schedule for implementation.

5) In this stage implementation will start. As mentioned above, key process areas will be done according to the sequence gained by the organization. When an organization has achieved the level 2 maturity, it has most likely grown in size and maturity to the degree that the original CMM is applicable. If not, steps 1 through 5 must be done iteratively.

In order to find out the situation of small organizations in Iran a questionnaire was designed. This questionnaire was spread among the senior managers of some small software organizations. The result of the survey conducted shows:

Almost all of the managers said that in spite of having fewer than 20 staff, they feel the need of a framework in which they could accomplish their tasks and for continuous process improvement. Without this framework, the organizations do not have a uniform method for their activities. Many of them suffered from disorder and they think their failures are due to this fact. The managers believe that continuing like this will not provide the means of improvement for their organizations. Those managers who knew about CMM thought that it is not appropriate for small organizations. After reviewing the solution mentioned before they agreed that it could be a substitute for CMM but it is necessary to be customized with their own organizational structure.

CONCLUSION

Software process improvement should be done to help the organization. This is true for both large organizations and small. Although, CMM is intended to focus on large projects and organizations [4] a procedure could be designed to make it applicable to small organizations as well. The solution proposed in this paper consists of 5 steps. At the first step, the language of the organization must be interpreted to the terminology of CMM. At step 2, roles which are not mentioned in the organization are identified. Those activities which are found necessary to be done, should be prioritized in their own level, and also they must be defined at the proper level of detail. In this step the activities must become ordered, the outcome is an activity sequence, according to which in the next step a detailed plan must be designed. The plan designed in the previous step now can be implemented. For the growing organizations which do the steps only for the second level of CMM, the steps must be repeated until they grow into a large organization.

REFERENCES

[1] David I. Cleland, and Lewis R. Ireland, *"Project Management: Strategic and Implementation"*, 4th Edition, 2002.

[2] Mark C. Paulk, "*Effective CMM-Based Process Improvement",* Proceedings of the 6th International Conference on Software Quality, pp. 226-237, October 1996.

[3] Software Development Report, Iran Telecommunication Research Center, September 2003.

[4] Mark C. Paulk, "*Extreme Programming from a CMM Perspective",* XP for Universe, Raleigh, NC, July 2001.

[5] Faranak Razeghi, " *Statistical Set of Informatics Organizations in Iran*" , Journal of the Iran's High Council of Informatics, Vol 19, Issue 94, pp. 50-56.

[6] Donna L. Johnson and Judith G. Brodman, "*Applying the CMM to Small Projects",* Proceedings of the 1998 Software Engineering Process Group Conference, Chicago, IL, 9-12 March 1998.

[7] Paulk, Mark C.; Curtis, Bill; Chrissis, Mary Beth Chrissis, and Weber, Charles, *"Capability Maturity Model for Software, Version 1.1"*, Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403, February 1993.

[8] Paulk, Mark C.; Weber, Charles V.; Garcia, Suzanne M. Garcia, Chrissis, Mary Beth; and Bush, Marilyn W. "*Key Practices of the Capability Maturity Model, Version 1.1",* Software Engineering Institute, CMU/SEI-93-TR-25, DTIC Number ADA263432, February 1993.