ARUNAI PUBLICATIONS

Asian Journal of Applied Research

DOI: http://dx.doi.org/10.20468/ajar.2017.12.003

Research Article

A Detailed Study on CMMI Process Improvements and Methods

S. Vince Raicheal[1], M. V. Srinath[2]

Department of Computer Applications, STET Women's College, Mannargudi, Tamil Nadu, India

ABSTRACT

Software systems incorporated belong to a wide range with the use of software system; a need also arises for the proper maintenance thereof. Nowadays, the expenses and effort required for software maintenance are of great scale. Software societies are on the path of growth for effective functioning and progress for producing palpable results for customers and stakeholders. A promising method for attaining this is acceptance of a process improvement model. Capability maturity model integration (CMMI) is a tool for executing the best practices for activities regarding products and services in organizations. The most significant benefits of software process maturity models and standards within the organizations are customer satisfaction, rework reduction, and quality improvement. This paper presents a detailed study on CMMI process improvements and methods. The process improvements explain the three sub-processes, namely, process measurement, process analysis, and process change. It helps to understand the CMMI process procedures and strategies in a useful manner. The staged CMMI model is also discussed in this paper. The best practices in the CMMI models have been extensively evaluated by users.

Address for correspondence:
Dr. M. V. Srinath, Director, Department of Computer Applications, STET Women's College, Mannargudi, India. E-mail: sri_induja@rediffmail.com

Keywords: Capability maturity model integration, Capability maturity model integration process improvements, Software process, Software systems

Received: 18th October 2017
Accepted: 20th October 2017
Published: 05th December 2017

INTRODUCTION

Capability maturity model integration (CMMI) has found wide usage for evaluating the organizational maturity and process capability throughout the world over the past decades.[1,2] Nowadays, most of the organizations use to regular CMMI calculations and appraisals. They have self-assurance in CMMI due to its wide-ranging descriptions of how diverse good practices fit together. Moreover, there is a continuous demand from industry for inexpensive, better software that has to be delivered to ever-tighter targets. Accordingly, many software concerns have motivated software process improvement as a way of improving the quality of their software, minimizing costs, or accelerating their development processes.[3] Process improvement is defined as understanding the existing processes and making alterations to these processes to improve product quality, development time, and reduce costs.[4] Two different approaches are used for process improvement and change. They are

1. Process maturity approach and
2. Agile approach.

The method maturity approach is devoted to enhancement of process and project management and presenting sensible software package engineering practices into the software business.[5] The extent of process maturity reproduces the extent to which sensible procedural and management coaching have been enforced within the structure development processes. The key objective of this approach is to enhance the method foregone conclusion and merchandise quality. This approach, non-moving in plan-driven development usually necessitates enlarged "overhead", within the sense that events square measure given that are not directly associated with programming. The agile approach motivates reduction of overheads and repetitious development within the software process.[6] The principal characteristics of the agile technique square measure speedy delivery of responsiveness and practicality to fulfill the ever-changing client necessities. The focus of the technique is code being established and minimizes the documentation and formality.[7]

People usually believe that improving a software development process can yield improved quality of software. It depends on calculating the number of product defects and connecting these defects to the software process. The objective is to minimize product faults by examining and modifying the process. Hence, the changes required for locating defects are minimized and the fault detection can be improved. Figure 1 shows the factors affecting the software product quality.

The four important factors that affect the quality of any software are as follows: Process quality, people quality, development technology and cost, time, and schedule.[8] The impact of each of these factors is influenced by the size and the type of the software. For very large systems that contain separate

sub-systems, initiated by development teams who may be working in different places, the major factor that affects product quality is the software process. The main difficulties with large projects lie in project management, integration, and communications. There is generally a combination of experience and abilities in the team members with the development process generally taking place over a number of years, and the development team is volatile or unstable. There may be revolutionary changes over the lifetime of the software project.[9-11]

The product quality or method quality relationship is a smaller amount recognizable once the merchandise depends and, intangible, to some extent, on intellectual developments that cannot be machine-controlled. The standard of the software is not influenced by its engineering method, however, rather, on its style method, wherever over information expertise and skills are vital. In some circumstances, the method used would be the foremost substantial reason for the merchandise quality, though, especially for innovative applications, the folks concerned within the method have additional impact on computer code quality than the method used.[12]

PROCESS IMPROVEMENT METHODS

The practice of process improvement involves a cyclical procedure which is shown in Figure 2.

It includes three sub-processes:

1. Process measurement: Aspects of the current product or the project are estimated. The objective is to develop the measures on the basis of the goals of the organization involved in the process improvement. This practices a baseline which supports of the customer's choice if process improvements have been effective.
2. Process analysis: The current process is measured, and process bottlenecks and weaknesses are recognized. Process models called process maps which define the process may be established during this step. The analysis may be motivated by considering the process features such as rapidity and robustness.
3. Process change: Process changes are recommended for addressing some of the recognized process weaknesses. These are presented. This is followed by the restarting of the cycle to gather data relating to the efficiency of the modifications.

Process improvement is an enduring activity, with each of the steps in the improvement process lasting several months. It is also a continuous activity as, with new processes presented, there are bound to technologies in the business environment, and new procedures have to grow to respond to these changes.

Process measurement

Process measurements are quantifiable data about the software practice such as the time taken to execute a process activity. Process measurements can be utilized to ascertain whether or not the efficacy of a process has been developed. At this stage, there are three types of process metrics can be collected:

1. The time taken for the completion of a specific process can be calendar time, the total time dedicated to the process or the time spent on the process by specific engineers, and so on.

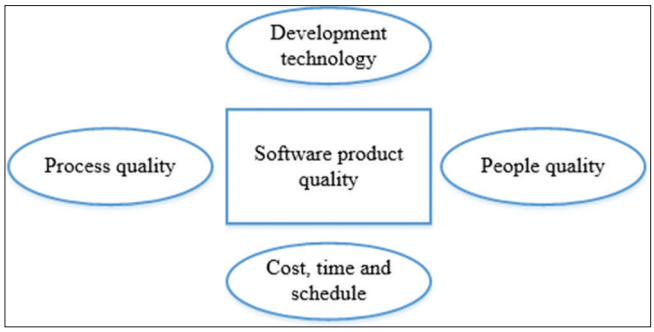

Figure 1: Factors affecting software product quality

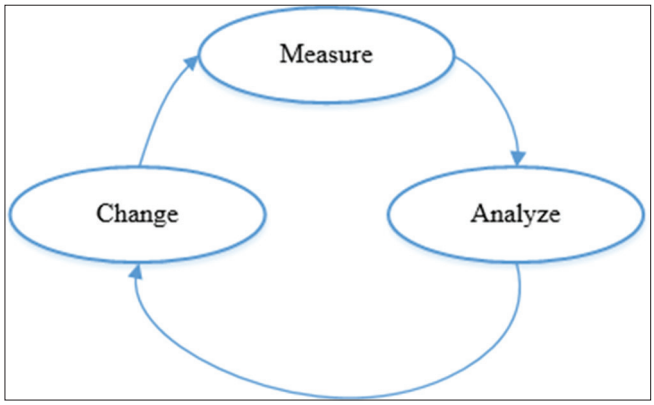

Figure 2: Process improvement lifecycle

2. The resources required for a specific process. Resources might contain total effort in the form of travel costs, computer resources, or person-days.
3. The number of incidence of a specific event. Samples of events which might be observed comprise the number of defects exposed during the number of requirements for changes requested, code inspection and the average number of lines of code changed in response to a requirements change.

An ultimate difficulty in process measurement is known what data about the process can be collected to support process improvement. Basili and Rombach[13] have suggested what they call the goal-question-metric (GQM) paradigm, which has found extensive use in software and process measurement. Basili *et al.*[14] explained how this method has been used in an enduring, measurement-based process improvement platform in the US space agency NASA. Figure 3 shows the GQM paradigm.

The GQM paradigm is utilized in process improvement to provide answers to the following three critical questions:

1. Why are we presenting process improvement?
2. What information do we need to help recognize and measure improvements?
3. What product and process measurements are essential to deliver this information?

These questions are directly associated to abstractions that include GQM in the GQM paradigm: They are

1. Goals: A goal is something that the association demands for accomplishment. It needs not be directly concerned with the process attributes but instead of how the process disturbs

the products or the organization itself. Examples of goals might be an improved level of process maturity, increased product reliability, or a smaller product development time.
2. Questions: These are modifications of goals where precise areas of uncertainty linked to the goals are recognized. Usually, a goal has a number of related questions which require answers. Examples of questions associated with the goal of shortening product development times might be "How can the time required to finalize product requirements with customers be reduced?," "How many of our tests are effective in discovering product defects?," and "Where are the bottlenecks in our current process?"
3. Metrics: These are the measurements required for collection to help answer the questions and to check whether or not process improvements have accomplished the expected goal. Answering the above questions requires the collection of data on the time taken to execute each process activity (normalized by system size), the number of defects discovered per test run, and the number of formal communications between clients and customers for each requirement change.

The benefit of using the GQM method in process improvement is that it distinguishes organizational goals and concerns from precise process concerns. It offers a basis for determining what data should be suggested and collected, which data should be examined in different ways, depending on the question it required to answer.

Process analysis

Process analysis is the study of procedures that help appreciate their key features and how such processes are accomplished in practice by the people involved. Process analysis has a number of closely associated objectives:

1. To understand the relationships among the process actions and the measurements that have been made.
2. To understand the actions involved in the processes and the relationships among these activities.
3. To relate the precise process or processes under examination to comparable processes elsewhere in the organization, or to idealized processes of a similar type.

The most usually used techniques of process analysis are as follows:

- Ethnographic studies: In this analysis, process contestants are detected as they work. They may be used for recognizing the nature of software development as a human activity. Such analysis exposes refinements and difficulties which may not be disclosed by questionnaires and interviews.
- Questionnaires and interviews: Software engineers and managers working on a project are interrogated about what actually is going on. The answers to questions in a questionnaire are developed during personal interviews with those involved in the process.

Process change

Process change includes making alterations to the existing process. It should be determined by improvement goals such as minimize the number of defects exposed during integration testing by 25%. After the changes have been executed, the process measurements are used to measure the effectiveness of the changes. There are five key phases in the process change process. These are explained in the following Figure 4.

1. Improvement identification: This phase is concerned with using the results of the process analysis for discovering the ways to tackle quality issues, cost inefficiencies, or schedule bottlenecks that have been seen during process analysis. Furthermore, new process structures, methods, processes, and tools can be used to address the process problems. For instance, an enterprise may trust that many of its software difficulties stem from requirements difficulties. With the help of an engineering best practice guide,[15] numerous requirements of engineering practices which could be introduced or changed may then be recognized.
2. Improvement prioritization: This phase is concerned with the evaluation of possible changes to the process and ordering them for implementation. When introduce them all at once, it is necessary to decide which is the most important. Furthermore, there is the need to make these decisions based on the requirement to improve precise process areas, the impact of a change on the organization, the costs of introducing a change, or other factors. For instance, an enterprise may consider the overview of requirements of management processes to manage the developing requirements to be the peak priority in process change.
3. Process change introduction: This phase facilitates new methods, procedures, and tools into place and incorporates them with other process activities. It should allow enough time to present changes and confirm that these changes are compatible with other process activities and structural standards and procedures. This may include acquiring tools for the requirements management and designing processes to utilize these tools.
4. Process training: Process training is a sin qua now for attaining the full benefits of process changes. The software engineers involved need to know the changes which have been suggested and how to accomplish the new and changed processes. Usually, process changes are executed

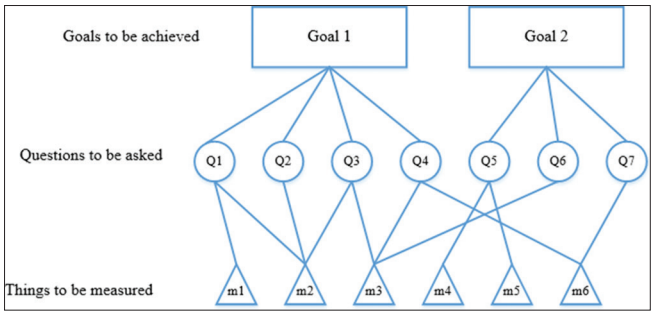

Figure 3: Goal-question-metric paradigm

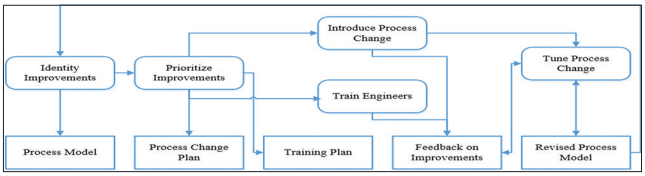

Figure 4: Process change steps

without sufficient training, and the effect of these changes is to worsen rather than increase the product quality. In this case of requirements management, a description of the process activities and an overview to the tools that have been nominated, and the process training might include a discussion of the charge of requirements management.

5. Change tuning: At this stage, the proposed process changes will never be entirely operative soon after introduction. It requires a tuning phase where minor difficulties can be identified, and modifications to the process can be recommended and introduced. This tuning stage may past several months until the development engineers are satisfied and happy with the new process.

CMMI PROCESS IMPROVEMENT FRAMEWORK

The CMMI model is very difficult, with more than 1000 pages of explanation. The main model components are as follows:

1. A set of process areas which are associated with software process actions. The CMMI recognizes 22 process areas which are appropriate to software process capability and improvement. These are systematized into four groups in the continuous CMMI system. These groups and related process areas are listed in Table 1.
2. A number of goals, which are abstract explanations of an appropriate state which should be managed by an organization. The CMMI process has precise goals which are connected with each process area, describing the state appropriate for that area. It also explains the generic goals which are related with the institutionalization of good training.
3. A set of good practices, which are explanations of ways of attaining a goal. Numerous specific and generic practices may be related with each goal within a process area. Although the CMMI identifies it as the goal the way that the goal is reached is significant. Organizations may use any suitable practices to accomplish any of the CMMI goals they do not have to accept the practices suggested in the CMMI.

Staged CMMI model

A CMMI valuation includes observing the processes in an organization and rating these processes or process areas on a six-point measure which relates to the level of maturity in every process area.[16] Figure 5 shows the staged CMMI model. The six-point scale allocates a level of maturity to a process area which is defined as follows.

Level 0 incomplete

At least, one of the precise goals related to the process area is not fulfilled. There are no generic goals at this level as institutionalization of an inadequate process does not make sense.

Level 1 performed

The goals related with the process area are fulfilled, and for each process, the scope of the work to be achieved is explicitly set out and interconnected to the team members.

Table 1: Process areas in CMMI

Category	Process area
Process management	OPD
	OPF
	OT
	OPP
	OID
Project management	PP
	PMC
	SAM
	IPM
	RSKM
	QPM
Engineering	REQM
	RD
	TS
	PI
	VER
	VAL
Support	CM
	PPQA
	MA
	DAR
	CAR

OPD: Organization process definition, OPF: Organization process focus, OT: Organization training, OPP: Organization process performance, OID: Organizational innovation and deployment, PP: Project planning, PMC: Project monitoring and control, SAM: Supplier agreement management, IPM: Integrated project management, RSKM: Risk management, QPM: Quantitative project management, REQM: Requirements management, RD: Requirements development, TS: Technical solution, PI: Product integration, VER: Verification, VAL: Validation, CM: Configuration management, PPQA: Process and product quality management, MA: Measurement and analysis, DAR: Decision analysis and resolution, CAR: Casual analysis and resolution, CMMI: Capability maturity model integration

Level 2 managed

At this level, the goals related to the process area are encountered, and organizational policies are in place which indicates when each process should be used. Project plans should be identified and documented, defining the project goals. Process monitoring procedures and resource management should be in place across the organization.

Level 3 defined

The focus of this stage is focused on organizational deployment and standardization of processes. Every project has a managed process which is modified to suit the project requirements from a distinct set of organizational processes. Process assets and process measurements should be collected and used for upcoming process improvements.

Level 4 quantitatively managed

At this stage, there is an organizational concern for utilizing statistical and other quantitative approaches to control the sub-processes. That is, collected process

and product measurements should be used in process management.

Level 5 optimizing

This is the highest level of the staged CMMI model. Here, the organization should utilize the product and process measurements to ensure process improvement. Trends must be examined and the processes modified to suit the changing business requirements.

Continuous CMMI model

Continuous maturity models do not categorize an organization on the basis of distinct levels. Relatively, they are finer-grained models which consider individual or groups of practices and evaluate the use of good practice within each process group. Therefore, maturity assessment is not a single value but a set of values presenting the organization's maturity for each process/process group. Continuous CMMI reflects the process areas these are shown in Table 1. It allocates a capability assessment level from 0 to 5 to each process area. Usually, organizations work at diverse maturity levels for different process areas.

Subsequently, the outcome of a continuous CMMI assessment is a capability profile presenting each process area and its related capability assessment. A portion of a capability profile which shows processes at diverse capability levels is shown in Figure 6.

This illustrates the level of maturity in configuration management, for example, as high, but that risk management maturity is very little. A company may progress actual and target capability profiles where the target profile reproduces the capability level which they would like to reach for that process area.

The principal benefit of the continuous model is that enterprises will choose and choose processes for improvement supported their own wants and needs. For instance, associate degree enterprise that progresses software for the part trade could concentrate on enhancements in configuration management, system specification, and validation, whereas an internet development company is also a lot of anxious concerning customer-facing procedures. The staged model desires firm to concentrate on the various stages successively.

CONCLUSION

This paper has presented a detailed study on CMMI process improvements and methods. Many other establishments are trying numerous process improvement programs for software development and recognizing it as highly making problematic to make the changes necessary to be more severe and well-organized in their engineering practices. The three sub-processes of process improvements have been detailed in this paper. CMMI models are planned to define discrete stages of process improvement. In the staged representation model, the maturity levels offer a recommended order for approaching process improvement in stages so that not all the process areas are addressed at the same time. The CMMI frameworks require a consistent and regular structure for their concepts for supporting consistent implementation and interpretation. The

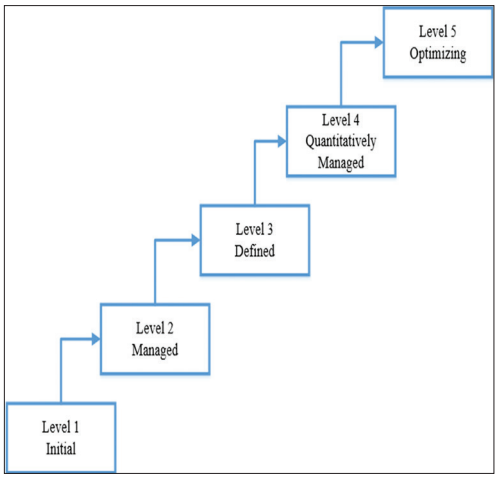

Figure 5: Capability maturity model integration staged maturity model

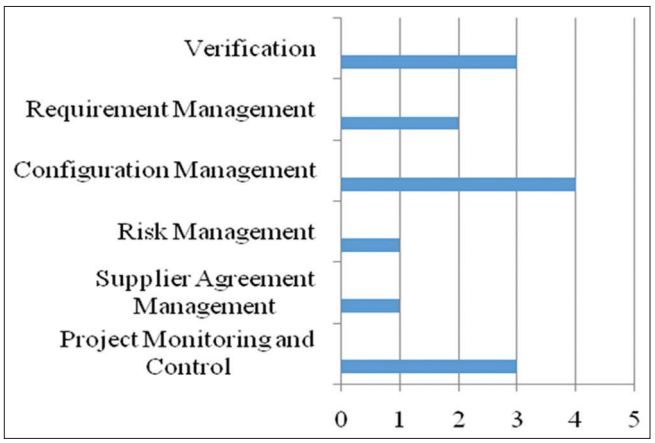

Figure 6: Process capability profile

frameworks are required to be abstract and flexible for utilize in an extensive range of environments.

REFERENCES

1. Ho D, Kumar A, Shiwakoti N. "Maturity Model for Supply Chain Collaboration: CMMI Approach". In: 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM); 2016. p. 845-9.
2. Khan AA, Keung J. Systematic review of success factors and barriers for software process improvement in global software development. IET Softw 2016;10:125-35.
3. Domingues P, Sampaio P, Arezes PM. Integrated management systems assessment: A maturity model proposal. J Clean Prod 2016;124:164-74.
4. Iqbal J, Ahmad RB, Nasir MH, Niazi M, Shamshirband S, Noor MA. Software SMEs' unofficial readiness for CMMI®-based software process improvement. Softw Qual J 2016;24:997-1023.
5. Doss D, Henley R, McElreath D, Gokaraju B, Goza R, Lusk G. Process Improvement: Urban vs. Rural Personnel Perspectives of a Derivative CMMi Process maturity framework. In Proceedings of the Southwest Academy of Management; 2017. p. 44-51.
6. Silva FS, Soares FS, Peres AL, de Azevedo IM, Vasconcelos AP, Kamei FK, *et al*. Using CMMI together with agile software development: A systematic review. Inf Softw Technol

2015;58:20-43.
7. Soares FS, de Lemos Meira SR. An Agile Strategy for Implementing CMMI Project Management Practices in Software Organizations. In 2015 10th Iberian Conference on Information Systems and Technologies (CISTI); 2015. p. 1-4.
8. Fenton N, Bieman J. Software Metrics: A Rigorous and Practical Approach. Boca Raton: CRC Press; 2014.
9. Mane M, Joshi M, Kadam A, Joshi S. software reliability and quality analyser with quality metric analysis along with software reliability growth model. Int J Comput Sci Inf Technol 2014;5:3803-6.
10. Gordieiev O, Kharchenko VS, Fusani M. Evolution of Software Quality Models: Green And Reliability Issues. In ICTERI; 2015. p. 432-45.
11. Paternoster N, Giardino C, Unterkalmsteiner M, Gorschek T, Abrahamsson P. Software development in startup companies: A systematic mapping study. Inf Softw Technol 2014;56:1200-18.
12. Ke SZ, Huang CY, Peng KL. Software Reliability Analysis Considering the Variation of Testing-Effort and Change-Point. In Proceedings of the International Workshop on Innovative Software Development Methodologies and Practices; 2014. p. 30-9.
13. Basili VR, Rombach HD. The TAME project: Towards improvement-oriented software environments. IEEE Trans Softw Eng 1988;14:758-73.
14. Basili V, Zelkowitz M, McGarry F, Page J, Waligora S, Pajerski R. SEL's software process improvement program. IEEE Softw 1995;12:83-7.
15. Sommerville I, Sawyer P. Requirements Engineering: A Good Practice Guide. New York: John Wiley & Sons, Inc.; 1997.
16. Falessi D, Shaw M, Mullen K. Achieving and maintaining CMMI maturity level 5 in a small organization. IEEE Softw 2014;31:80-6.

Cite this article: Raicheal SV, Srinath MV. A Detailed Study on CMMI Process Improvements and Methods. Asian J Appl Res 2017;3(12):10-15.

Source of Support: Nil, **Conflict of Interest:** None declared.