Volume 3, Issue 12, December 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Total Quality Management in Software Development Process

Muneshwara M.S[1] ,Anil G.N[2]
Department of CSE, B.M.S Institute of Technology
Avalahalli Yelahanka Bangalore-560064,
Karnataka,India

Abstract— ***'Quality' is an important characteristic for a product. Software is intangible by nature, and hence defining quality is a challenge. Software development processes also contribute to the quality of the software. But, the standard does not emphasize on the software process. Quality can also be achieved by following a well defined process standard. Two standards CMM, by Software Engineering Institute of Carnegie M University and ISO9000 are widely used in the software industry. But CMM has helped several organizations to achieve Business Excellence. Organizations with CMMI perform better than that of organizations that are CMM certified.***

Keywords— ***CMM, CMMI, ISO 9001,ISO 9000,maturity levels***

I. Introduction

Computers are becoming increasingly important day by day. Their applications are in every possible industry. It is almost impossible to run this world without computers. Consider a situation, where an airport ATC has a technical problem. Its software is corrupt and computer systems are not working properly. Such a situation can cause havoc, as flights approaching the airfield have no instructions, about landing. This can lead to a disaster, resulting in loss of life of many. This shows us how important is software in managing critical systems like ATC. It also plays a major role in many other critical systems like launching rockets, missiles, and also several business critical systems like Banking Software, stock market etc.

Quality of the software being developed is a critical success factor, for the companies in software industry. Clients demand for best quality software and are often not ready to pay more. Hence it is a challenge for a software company to develop good quality software, at lowest possible cost. The goals of Software Engineering are generally recognized to be the development of software that is reliable, economical, and which works as intended. These goals are familiar to any good industrial engineer who has been trained to consider production in terms of quality, reliability, and economy[4]. the roles of Industrial Engineers remain the same in the software industry, as that of their role in any manufacturing industry. 'Optimal resource allocation for staffing and equipment is much the same whether the people are engineers, programmers, or welders, and the equipment is computers, terminals, or printers'[4].

Two approaches can be followed to ensure software quality. One is focused on a direct specification and evaluation of the quality of software product, while the other is focused on assuring high quality of the process by which the product is developed'[9]. In my opinion, the second option, i.e. assuring high quality of the process by which the product is developed, is the best way to develop a quality software product.

There are standards like ISO standard 25010 (replaces since 2011 the ISO 9126) which describe the quality characteristics of a software. There are attributes for every characteristic mentioned in the above document, which measure quality of software, but no guidelines or procedures are given about how to develop a quality software product. 'In the software industry, ISO 9000 and the Software Engineering Institute's (SEI) standards, i.e. Capability Maturity Model (CMM), CMMI and PCMM are main quality certifications to achieving TQM and business excellence[3]. These quality standards clearly define best practices for a Quality Management System in the Software Industry.

A. Quality

The quality of a software can be determined by comparing a set of inherent characteristics with a set of requirements. If those inherent characteristics meet all requirements, a high quality of the software is achieved. If those characteristics do not meet all requirements, poor quality software is achieved. The quality of a software depends on a set of inherent characteristics, requirements, and how well they comply with each other.

B. Quality management system

A quality management system is a set of interlinked elements that organizations use to supervise how quality policies are processed and quality objectives are met.A process-based QMS is used to implement and check how its quality policy is applied and objectives are achieved. A QMS is a process-based network of various interrelated and interlinked processes (elements).

Each process uses resources to convert inputs to outputs. Since the output of one process is the input of another, they interact and are interrelated by means of such input-output relationships. the above process interactions create a single process-based QMS.

II. IMPORTANCE OF SOFTWARE QUALITY

We saw how important software is in our lives in today's world. Quality of the software being developed is a critical success factor, for the companies in software industry. Clients demand for best quality software and are often not ready to pay more. Hence it is a challenge for a software company to develop good quality software, at lowest possible cost. Industrial Engineers always deal with efficiency and effectiveness of a product. Achieving effectiveness is not difficult for software engineers, where as efficiency is.

'Two approaches can be followed to ensure software quality. One is focused on a direct specification and evaluation of the quality of software product, while the other is focused on assuring high quality of the process by which the product is developed' (Sacha, 2005). In my opinion, the second option, i.e. assuring high quality of the process by which the product is developed, is the best way to develop a quality software product. There are standards like ISO standard 25010 (replaces since 2011 the ISO 9126) which describe the quality characteristics of a software. There are attributes for every characteristic mentioned in the above document, which measure quality of software, but no guidelines or procedures are given about how to develop a quality software product. 'In the software industry, ISO 9000 and the Software Engineering Institute's (SEI) standards, i.e. Capability Maturity Model (CMM), CMMI and PCMM are main quality certifications to achieving TQM and business excellence'(Jain and Gupta, 2011). These quality standards clearly define best practices for a Quality Management System in the Software Industry.

High quality software meets the needs of users as it provides reliability, maintainability and portability and easily integrated with other tools. We will look at how to achieve quality, the tradeoffs involved, modeling quality improvement, and standard designed to ensure quality. If we don't focus on quality of a product then: we tend to produce components with hidden defects, We have to spend a lot of time fixing these., we have little time for anything else even after spending so much of time in defect checking we tend to produce poor quality software. From the requirements to the delivery of the software product, the development process is complex. It involves a series of stages, each with feedback paths.

The very first stage of the software life cycle is the requirements stage, where the company representatives coordinate with the clients to understand the need of the customers. They also keep in mind future trends, and business objectives of the company in this stage. Later the design team will design a system, which comprises of both hardware and software. Once the design is finalized, the project is implemented in separate units, and they are tested. The next stage is the system integration stage, where the sub systems are integrated and the whole system is tested. The last stage is the operation and maintenance stage, where the system is deployed and regular maintenance is performed.

III. EVOLUTION OF SOFTWARE QUALITY STANDARDS

ISO 9001 is based on eight Quality Management Principles, which are included within the requirements of the standard, and are implemented to improve organizational performance:

1.Customer focus
2.Leadership
3.Involvement of people
4.Process approach
5. System approach to management
6. Continual improvement
7. Factual approach to decision making
8. Mutually beneficial supplier relationships

ISO 9001 applies in situations when:

a) Design is required and the product requirements are stated principally based on performance terms, or when the establishment needs to be done, and

b) Confidence in product conformity can be attained by adequate demonstration of a supplier's capabilities in design, development, production, installation and servicing.

ISO 9002 applies in situations when:

a) The specified requirements for product are stated in terms of an established design or specification, and

b) Confidence in product conformity can be attained by adequate demonstration of a supplier's capabilities in production, installation and servicing.

IV. CAPABILITY MATURITY MODEL (CMM)

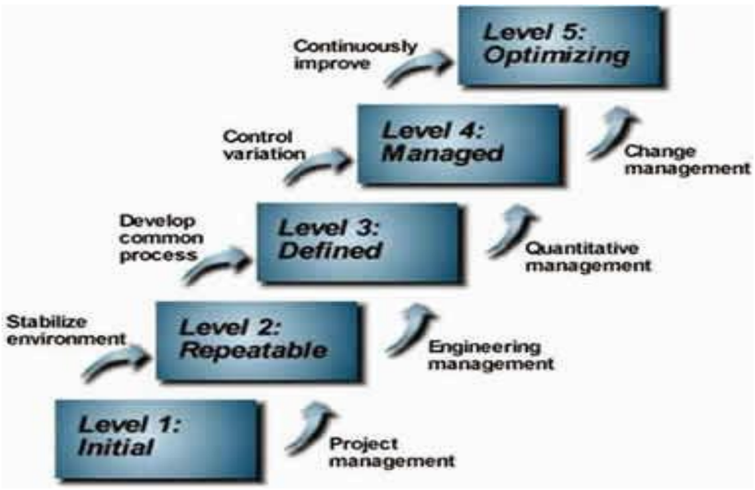

Figure 1. levels of CMM

The Capability Maturity Model for Software 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' (Paulk et.al, 1993). The CMM model has five different maturity levels. Each level signifies how mature the processes in the organization are. Each of these maturity levels have a set of processes, goals and objectives, by achieving which there can be an improvement in the quality of the process, and hence also a better quality end product.

A. Five levels of CMM

1. Initial Level

In the initial level, there is no stable environment in the organization for the development and maintenance of the software. Processes are chaotic, not organized. Any success brought in the organization is, may be because of individual efforts. The processes are continuously changed, and often the budget, functionality and even the product quality is unpredictable.

2) Repeatable Level

An organization is said to be in the repeatable level, if there are policies for managing software projects. The procedures for implementing these policies are also defined by the organization. An organization, in this level manages new projects based on the experiences from the past, with similar themes. The managers can track costs, schedules and functionality. Software Project Standards are defined, and are also faithfully followed. Any relationship with suppliers is strongly established. It is disciplined, as planning and tracking of the project is more stable.

3) Defined Level

An organization in this level, has documented, standard processes for developing and maintaining software across the organization. Both software engineering and management processes are integrated by the organization and function as one coherent single entity. There is a dedicated team in the organization to manage the software process. Training programs are organized to educate the employees, about their roles and responsibilities. Due to the presence of a well defined standard software process, the management has good insight into technical progress on all projects. We can call the organization standard and consistent, as both software engineering and management principles are used together as one standard process, and it is stable and repeatable.

4) Managed Level

Quantitative and qualitative goals are set for both software products and processes. Productivity and quality are measured for important software processes. Control is achieved by narrowing the variation in the processes and products. Risks involved in venturing into new domains are known and are carefully managed. The software process capability in this level can be termed as predictable.

5) Optimizing Level

"Continuous Process Improvement", is the focus of the whole organization. An organization in this level has the ability to identify weaknesses and strengthen the process proactively. Improvement occurs both by incremental advancements in the existing processes and also by using new technologies or methods.

B. Process capability and prediction of performance

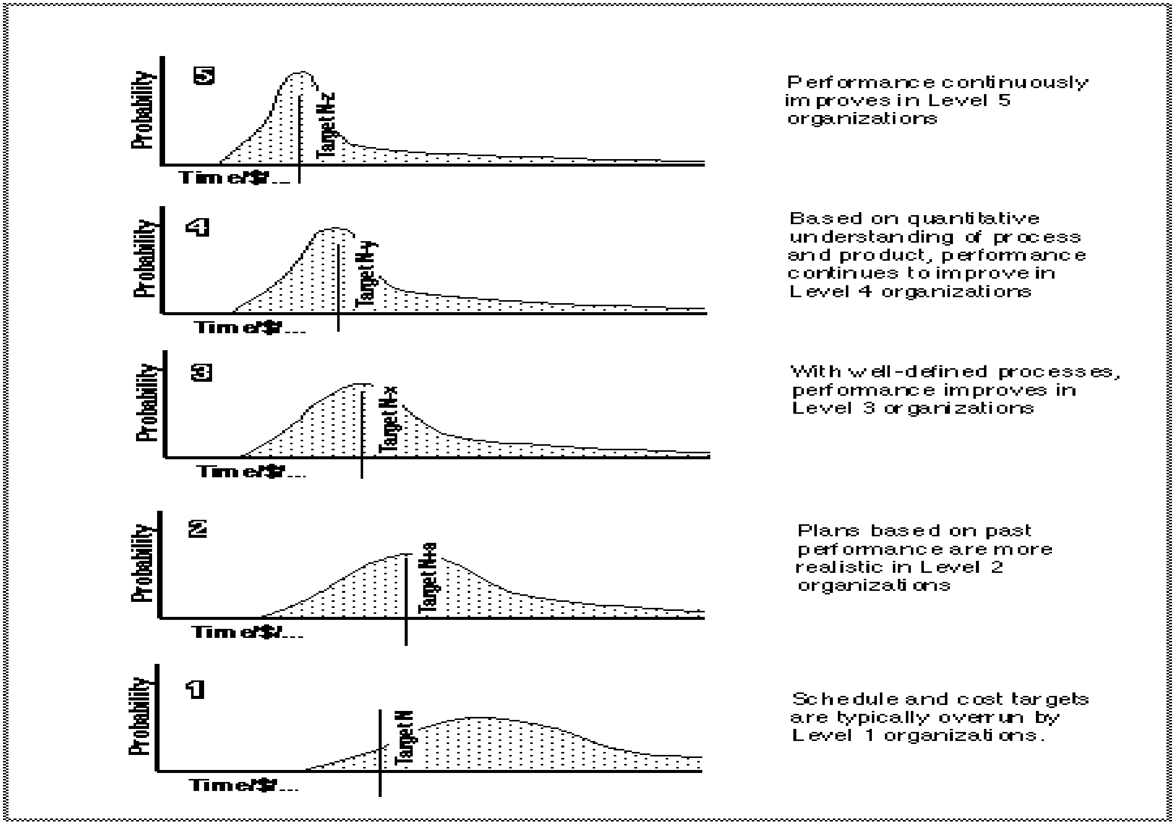

Figure 2. Capability Maturity Model

The maturity of an organization's software process helps to predict a project's ability to meet its goals' [8]. There is an illustration of this notion in the figure. With the increase in the maturity level, this variation in meeting the targets is decreasing.E.g. consider an imaginary situation where an organization has delivery dates for nine projects on January 1st of 2011. As the organization moves up in the maturity levels, the delivery dates for these projects is closer to the given date. For a level 1 organization, they very often miss the delivery date by a larger margin, and level 5 organizations must be able to meet the delivery date with a noticeable accuracy. This is because the software processes in the level 5 organization are constructed carefully and with the vast experience, they know how to take calculated risks. In the figure 2, we can observe this phenomenon, the area under the curve to the right of the target line, signifies the maximum probability that the firm reaches the targets on time.

Even when dealing with unprecedented systems, for an organization with maturity level 5, this variation in meeting the targets is less. This is because of the fact that management and engineering practices of the matured organization, help in tackling the situation, always better than that of an organization in the level 1. There is earlier detection of defects, and handling them, increases the stability and performance of the project, this eliminates rework in the later phases of the software lifecycle. In some cases unpractical projects, are identified in the earlier stages, and hence prevent the investment in the loss.

ISO9000 vs CMM

CMM seems to be a better quality standard in the Indian IT industry. 'Among the quality certified organizations, CMM certified software organizations have better TQM practices and business excellence as compared to ISO certified organizations. Because ISO 9000 provides general guidelines for all the organizations, which can be used for TQM purposes, whereas CMM stresses on process improvement and provides guidance for stable, capable, and mature processes by identifying the KPAs in software development. Quality is an important characteristic of software' (Jain and Gupta, 2007).

ISO certification stresses documentation over process improvement, while CMM focuses on continuous process improvement' [3]. Probably this is why, companies with ISO certification lag behind the ones who are CMM certified. Improving the process must be the ultimate goal of a standard, rather than documentation. Documentation is also an essential activity but, more emphasis on process improvement is necessary for an organization to prosper.

Budlong and Peterson say, 'CMM compliance may also change the manner in which a company interacts with its customers because there are stringent requirements for maintaining a high maturity level. Highly rated companies are more adept at handling quick demands by the customer. Fortunately, compliance leads to higher quality software at lower cost' [7].Several software companies have experienced a reduction in defects that ranged from as low as 10 per cent to as high as 80 per cent. One organization reported a 45 per cent decrease in the reduction error rate, while two other companies' product error rates decreased from by 1.89 per thousand source lines of code and by 0.59 per thousand noncommented source statements' [7].The above opinions by experts show that CMM is one the best software process quality standards in the world.

V. **CMMI**

The CMM was initially developed by the Software Engineering Institute (SEI) at Carnegie Mellon University during the late 1980s . This work is sponsored by U.S department of defence. Thus, CMM (and CMMI) are tailored to the needs and according to the characteristics of governmental organizations to a certain extent.

A Capability Maturity Model (CMM) is a reference model of mature practices , used to improve and appraise a group's capability to perform that discipline. Even for non-technical processes like managing and developing workforces CMMs were created(the P-CMM).

CMMI provides guidelines for an integrated approach towards process improvement and a phrased approach to introducing improvements.

CMMI consists of three disciplines.

- CMMI for Development (CMMI-DEV) - Product and service development
- CMMI for Services (CMMI-SVC) - establishment of services, management and delivery
- CMMI for Acquisition (CMMI-ACQ) - deals with Product and service acquisition.

At that point, the CMMI project was formed with the integration of three of these various CMMs among its major goals: the CMM for Software (SW-CMM), the Electronic Industries Alliance Interim Standard (EIA/IS) , and the Integrated Product Development (IPD) CMM . As a result of these , CMMI came up with a model framework that could be parametrized depending on a selected set of principles that an organization deems most relevant in order to achieve their goals[11].

A.Maturity levels of CMMI

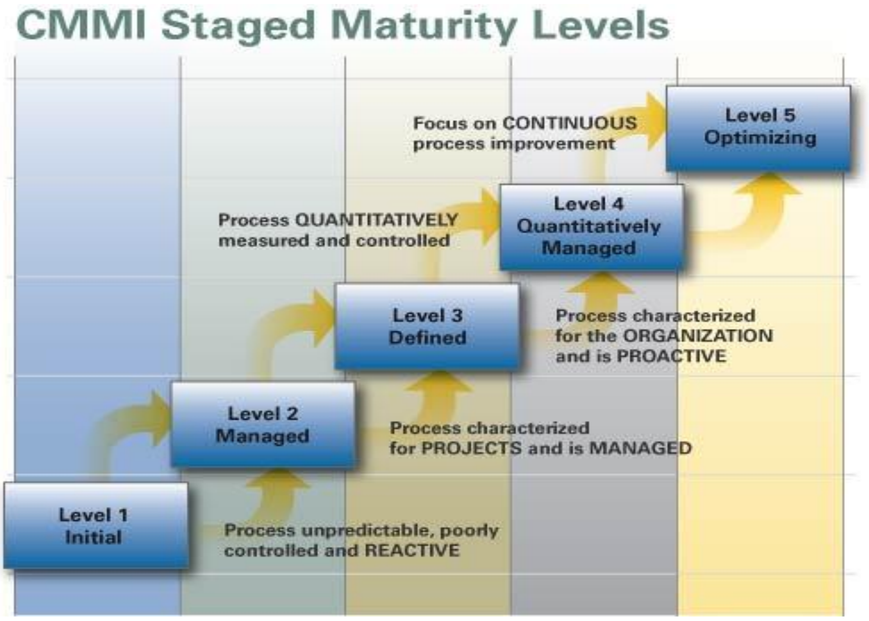
CMMI Staged Maturity Levels
Focus on CONTINUOUS process improvement
Level 5 Optimizing
Process QUANTITATIVELY measured and controlled
Level 4 Quantitatively Managed
Level 3 Defined
Process characterized for the ORGANIZATION and is PROACTIVE
Level 2 Managed
Process characterized for PROJECTS and is MANAGED
Level 1 Initial
Process unpredictable, poorly controlled and REACTIVE

Figure 3. CMMI maturity level

- **CMMI Maturity level 1-**At maturity level 1, processes are usually chaotic. The organization does not provide a stable environment. Success in these depends on the competence of the people in the organization and not on the use of proven processes.

- **CMMI Maturity Level 2 -** Managed
  CM - Configuration Management
  MA - Measurement and Analysis
  PMC - Project Monitoring and Control
  PP - Project Planning
  PPQA - Process and Product Quality Assurance
  REQM - Requirements Management
  SAM - Supplier Agreement Management
- **CMMI Maturity Level 3 -** Defined
  DAR - Decision Analysis and Resolution
  IPM - Integrated Project Management IPPD
  OPD - Organizational Process Definition IPPD
  OPF - Organizational Process Focus

OT - Organizational Training
PI - Product Integration
RD - Requirements Development
RSKM - Risk Management
TS - Technical Solution
VAL - Validation
VER - Verification

- **CMMI Maturity Level 4 -** Quantitative Management
  QPM - Quantitative Project Management
  OPP - Organizational Process Performance

- **CMMI Maturity Level 5 -** Optimizing
  CAR - Causal Analysis and Resolution
  OID - Organizational Innovation and Deployment[1]

B.CMM vs CMMI

Capability Maturity Model (CMM), the very first CMM is a 5 level assessment model.
The reason behind the development of CMM was to help U.S government in evaluating the provider's (software providers mainly) capability in handling large and complex projects. This model helps in solving problems relates to scheduling and budgeting. CMM became very successful and it was modified for other facets of an organization such as system engineering, people etc. However, CMMs are not without any problems. Many organizations found them to be contradicting and quite overlapping. It also lacked clarity and standardization.

CMMI or CMM Integration was developed mainly to focus on integrating current and upcoming models. It is an upgraded CMM model and it illustrates process improvements for organizations in software development. CMMI seem to favor large and bureaucratic organizations CMMI's exclusive focus on the process.

C. Advantages of CMMI over CMM

- It has a detailed coverage of product life cycle
- CMMI enables the users to eliminate the barriers that typically exist in different parts of an organization which can never be addressed
- CMMI Promotes collaboration between systems and software engineering.
  It is user friendly and allows them to choose model representation that best suits their objectives.

V. CONCLUSION AND FUTURE WORK

CMMI will stay the same for a while, but when it changes, anything is possible. While the current version and architecture of CMMI may continue to evolve along its current trajectory, this is only one possibility. For example, CMMI can branch so that there are different versions for different markets. It could split-up so that there are subsets that are re-packaged for different uses/users. Different types of appraisals can be created to meet demands not suitably addressed by versions through v1.3 of CMMI and the appraisal methods. Imagine, for example, versions of CMMI and of appraisals that focus on ongoing improvement in bottom-line performance, or versions that meet the specific targeted needs of start-ups and their venture backers. Imagine appraisers and consultants specifically qualified to work with lean, agile, start-ups, enterprise, operational services, technical debt, or DevOps, each with a version of CMMI, training, and appraisals suited specifically to their business and without the ambiguity currently experienced with only one version of everything for everyone.. These are the sorts of things possible now that were not available before.

ACKNOWLEDGMENT

The authors would like to thank the editor, mysterious reviewers for their valuable suggestions that appreciably improved the quality of this paper, specially thank to the HOD, Dept of CS&E for his constant encouragement and providing us the overwhelming support and guidance to write this paper. Finally also thankful our dear faculties Department of CSE, B M S Institute of Technology, Yelahanka,, Bengaluru, Karnataka, India.

REFERENCES

[1] CMMILevels and Certification www.cmmilevels.com

[2] Dymond, K.M., Govilkar, V.V., and Kumar, A., (2000). A Case Study of Life at Level 5: Does Your Lung Capacity
Really Increase from Breathing the Thin Air Higher Up? Proceedings of European SPEG 2000 Conference.
Krasnapolsky, AMSTERDAM.

[3] Jain, A., and Gupta, S.L., (2011). Significance of Quality Certification Towards Business. International Journal of Innovation, Management and Technology. 2 (3): pp. 204-209.

[4] Jinkins, R. D., and Rummel, P. A., (1990). Is there a place for IE in software development? Industrial Engineering, [online] 22(12), 27+. Retrieved from: http://go. galegroup.com/ ps/ i.do ?id=GALE %b7CA 9732475&v =2.1&u= inbhc & it=r&p=AONE&sw=w [accessed 26th Jan, 2013].

[5] Khan, P. M., and Sufyan Beg, M.M., (2012). " Measuring Cost of Quality (CoQ) - on SDLC projects is indispensible for effective Software Quality Assurance", International Journal of Soft Computing and Software Engineering [JSCSE], 2(9), pp. 1-15.

[6] Kan, S.H., Basili, V., and Shapiro, L.N., (March 1994). Software Quality: An Overview from the Total Quality Management Perspective, IBM Systems Journal, vol. 33(1): pp. 4-19.

[7] McManus, J., and Trevor, Wood-Harper., (2007). "Software engineering: a quality management perspective", The TQM Magazine, 19 (4), pp.315 – 327.

[8] Paulk, M.C., Curtis, B., Chrissis, M.B., Weber C.V., (1993). Capability Maturity ModelSM for Software, Version 1.1. Software Engineering Institute: CMU/SEI-93-TR-024 ESC-TR-93-177.

[9] Sacha, K., (2005). Evaluation of Software Quality. In: Proceedings of the 2005 conference on Software Engineering: Evolution and Emerging Technologies, Krzysztof Zieliński and Tomasz Szmuc (Eds.). IOS Press, Amsterdam, The Netherlands, The Netherlands, 381-388.

[10] Sommerville,I. (2008). Software Engineering 7th Ed.New Delhi: Pearson.

[11] http://www.dthomas.co.uk/dtalm/products/technologies/what-is-cmmi.htm