ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 6, August 2012

Role of Software Quality Assurance in Capability Maturity Model Integration

Rekha Chouhan[1] Dr.Rajeev Mathur[2]

[1]Research Scholar, Jodhpur National University, JODHPUR

[2] Director, CS, Lachoo Memorial College of Science and Technology, JODHPUR

ABSTRACT
With increasing demand for software products with high 'Quality', it has become imperative for organizations to adopt quality models like (International Standard Organization) ISO 9001,(Capability Maturity Model Integration)CMMI or Six Sigma etc. to set and sail on their quality journey. The strong emphasis on Software Quality Assurance in these models coupled with the modern day mantra of "Prevention", the need for pro-active Quality Assurance is higher than ever.Software Quality Assurance (SQA) is a planned and systematic approach necessary to provide adequate confidence that an item or product conforms to established standards, procedures, and policies. This paper addresses the role of Software Quality Assurance practices in the organization to assure "Quality" of Software Product Development. As the organizations grow and change, the needs and roles also change. Depending on the type of product and organization itself, the life cycles may differ and the tasks done by the quality organization evolve. The evolution takes familiar tracks, following patterns based upon the maturity of the organization and other factors. The Software Engineering Institute(SEI) Maturity Model and other standards are relevant in understanding the importance and roles for the quality group. Capability Maturity Model Integration (CMMI) is a process improvement approach to software development. This paper describes the importance of SQA for any organization growth.

Keywords:CMMI,ISO 9001, SEI, SQA.

1. INTRODUCTION

Software Quality Assurance (SQA) is a formal process for evaluating and documenting the quality of the work products produced during each stage of the Software Development Lifecycle (SDLC). The primary objective of the SQA process is to ensure the production of high-quality work products according to stated requirements and established standards.The word assurance means 'guarantee'. So the Quality Assurance Group's role is to guarantee that the product is of high quality. The main task of Software Quality Assurance Group is to examine the overall software development process and to create and enforce standards and methods to improve it with the goal of preventing bugs from ever occurring. With this definition, it is imperative that the SQA helps an organization in continuous performance improvement and strive for perfection. The roles of software quality assurance can be described with the tasks they undertake. The rolesrange from acting as an extension of development for debugging software products, to

developmentprocess definition and control. Verification and validation, acceptance testing, measurement, metrics, and process consulting are also roles that software quality groups sometimes assume.

International andother organizations provide the tools for "self-assessment" of an organization's SQA system and its operation. The Capacity Maturity Model (CMM) developed by the Software Engineering Institute (SEI), Carnegie Mellon University's SEI took theinitial steps toward development of what is termed a Capability MaturityModel(CMM) in 1986, when it released the first brief description of thematurity process framework. The initial version of the CMM was released in1992. After 1993, the SEI expanded the original Software Development andMaintenance Capability Maturity Model (SW-CMM) through diversification. In the late 1990s a new developmental direction was taken development ofintegrated CMM models. Development of specialized CMM models involveddevelopment of different sets of key processes for model variants for differentdepartments that exhibited joint processes. In practice, this created a situationwhere departments that applied different CMM variants in the same organizationfaced difficulties in cooperation and coordination. The CMMIapproach solved these problems at the same time as the modules better conformedto the emerging ISO/IEC 15504 standard. The CMMI model, like the original CMM models, is composed of five levels.The CMMI capability levels are the same as those of the original, apartfrom a minor change related to capability level 4, namely:

- Capability maturity level 1: Initial
- Capability maturity level 2: Managed
- Capability maturity level 3: Defined
- Capability maturity level 4: Quantitatively managed
- Capability maturity level 5: Optimizing.

A substantial change has nonetheless evolved with respect to the processesincluded in the models. The 18 key process areas of CMM Key Process Areas (KPAs) were replaced by 25 process areas (PAs). The PAs areclassified by the capability maturity level that the organization is required tosuccessfully perform. For each process area, objectives, specific practices andprocedures are defined.Application of the five levels of the CMM enables the organization to evaluateits achievements and determine what additional efforts are needed to reach thenext capability level. Process areas are generic, with the model defining "what" and leaving the"how" to the implementing organizations, i.e., the choice of life cycle model,design methodology, software development tool, programming language, anddocumentation standard.

Capability Maturity Model Integration (CMMI) is an approach to process improvement in which SQAplay a major role.Everyone in a software development organization takes part in both the CMMI processes and any improvement initiatives for those processes.

2. ORGANIZATION GROWTH AND MATURITY

One of the very first IEEE standards in the field of Software Engineering discipline was IEEE standard for Software Quality Assurance (SQA) [1], with the main purpose to provide uniform, minimum acceptable requirements for preparation and content of SQA plans [2]. According to [3], the SQA is software projects assurance that products and procedures conform to standards and plans. By using the SQA plan the software projects define their SQA activities.Within the CMM the SQA is one of the key process areas defined at CMM level 2, [4]. According to CMM definition, the purpose of the SQA is to provide management with appropriate

visibility into the process being used by the software project and of the products being built.

Organization maturity is not an indication of the age of the group. It can be defined as a measure of the formality of the processes used by software development. For healthy organizations, this slowly evolves through the maturity levels. The level of maturity of the organization roughly correlates with the role of the software quality group. The relationship of the software quality assurance group role to (Software Engineering Institute) SEI's Capability Maturity Model (CMM) is shown in **Table I**. For each level of maturity, the roles for the software quality group are shown.

Table I: Organization Maturity and SQA Roles

SEI Maturity Level	Role of SQA
1- Initial	Testing
2- Repeatable	Quality hurdle
3- Defined	Oversight, Metrics
4- Managed	Process and Risk management
5- Optimizing	Reference, Oversight

3. SOFTWAREQUALITY ASSURANCE AS PER CMMIPROCESS AREA

CMMI identifies a core set of Software Engineering process areas as:

3.1 Requirements Development
3.2 Requirements Management
3.3 Technical Solution
3.4 Product Integration
3.5 Verification
3.6Validation

CMMI also covers other process areas such as Process Management, Project Management, Quantitative Project Management, Causal Analysis and Resolution, Decision Analysis and Resolution, Organizational Process Definition ,Support process area, etc. but only the core Software Engineering development processes have been take here for discussion. It is also interesting to note that SQA and SQC (Software Quality Control) are processes defined within CMMI and they are under the support process area. In CMMI, SQA and SQC is defined as Process and Product Quality Assurance.

3.1SQA in CMMI Requirements Development

The CMMI Requirements Development process area describes three types of requirements:-customer requirements, product requirements, and product component requirements.

SQA role: To observe (audit) that documented standards, processes, and procedures are followed. SQA would also establish software metrics in order to measure the effectiveness of this process. A common metric for measuring the Requirements process would be the number of errors (found during system testing) that could be traced to inaccurate or ambiguous requirements. SQA would collect the metrics for monitoring and continuous improvement. SQA is more of an audit role here, and may sample actual Requirements whereas SQC is involved in the Verification of all Requirements. The type of requirement need not be just the functional aspect (or, customer user facing requirements) they could also include product and component requirements. The product requirements e.g. Supportability, Adaptability, and Reliability etc. are characteristics discussed here. The respective roles of SQC and SQA is the same for all types of requirement (customer and product) with SQC focusing on the 'internal deliverable' and SQA focusing on the process of how the internal deliverable is produced, as per the formal definition.

3.2 SQA in CMMI Requirements Management

The purpose of (CMMI) Requirements Management is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products. This process involves version control of the Requirements and the relationship between the Requirements and other work products. One tool used in Requirements Management is a Traceability Matrix. The Traceability Matrix maps where in the Software a given requirement is implemented, it is a kind of cross reference table. The traceability matrix also maps which test case verify a given requirement. There are other processes within Requirements Management and CMMI should be referenced for further information.

SQA role: To observe (audit) that documented standards, processes, and procedures are followed. SQA would also establish metrics in order to measure the effectiveness of this process. A common metric for measuring the Requirements Management would be the how many times the wrong Version was referenced. Another measure (for the Traceability Matrix) would be lack of test coverage, that is defects detected in the shipped product that were not tested due to the fact that they were not referenced in the Traceability matrix that referenced the requirements.

3.3 SQA in CMMI Technical Solution

The purpose of (CMMI) Technical Solution is to design, develop and implement solutions to requirements. Solutions, designs and implementations encompass products, product components and product-related life-cycle processes either singly or in combinations as appropriate. This is the main Design and Coding processes. CMMI puts the design and build together.

SQA role: To observe (audit) that documented standards, processes, and procedures are followed. SQA would also establish metrics in order to measure the effectiveness of this process. Clearly testing the end product against the requirements (which is itself a SQC activity) will reveal any defects introduced during this (the Technical solution) process. The number of defects is a common measure for the Design and Build phase. This metric is usually further quantified by some form of scope, for example defects per 100 lines of code, or per function. It is important that the defect may not always be a functional (or Customer facing defect) it could be that a required adaptability scenario is absent from the Design and coded solution.

3.4 SQA in CMMI Product Integration

The purpose of Product Integration is to assemble the product from the product components, ensure that the product, as integrated, functions properly and deliver the product. This is the final Integration and 'move to production' or product delivery. For large Software packages (consider SAP, Oracle Financials etc.) the assembly process is huge and the potential for errors is high. This process does not involve any coding but pure integration and assembly.

SQA role: To observe (audit) that documented standards, processes, and procedures are followed. SQA would also establish metrics in order to measure the effectiveness of this process. One measurement for this would be the defects found that resulted from the interface specifications (part of the Product requirements), potential process improvements could be to find other, perhaps less ambiguous ways of specifying interfaces. For example a development team may move to XML or Web Services for all interfaces, SQA could then measure the defects and report back to management and development as to the effectiveness of this change.

3.5 SQA in CMMI Verification

The purpose of Verification is to ensure that selected work products meet their specified requirements. These activities are only carried out by SQC, the role of SQA would be to make sure that SQC had documented procedures, plans etc. by audit. SQA would also measure the effectiveness of the Verification processes by tracking defects that were missed by SQC during Verification. Note the term Verification, as opposed to Validation. In essence Verification answers the question 'Are we building the product correctly' while Validation answers the question 'Are we building the correct product'. Validation demonstrates that the product satisfies its intended purpose when place in the correct environment while Verification refers to building to specification. Design or Code reviews are examples of Verification. These terms Verification and Validation are often mixed, CMMI makes this distinction, although "verification" and "validation" at first seem quite similar in CMMI models, on closer inspection we can see that each addresses different issues. Verification confirms that work products properly reflect the requirements specified for them. In other words, verification ensures that "we built it right." While SQC carries out all the Verification activities, the Verification process itself is still subject to SQA and process improvement.

3.6 SQA in CMMI Validation

Validation confirms that the product, as provided will fulfil its intended use. In other words, validation ensures that "we built the right thing." As with Verification, Validation is mainly the domain of SQC. The term Acceptance Test could also apply to Validation, in most cases the Acceptance test is carried out by a different group of people from the SQC team that performed Verification, as the product was being built. In the case where an application is going to be used internally, then the end user or business representative would perform the Acceptance testing. Wherever this is done it is in essence a SQC activity. As with Verification, SQA makes sure that these processes conform to standards and documented procedures. The Validation process itself is subject to continuous improvement and measurement.

4. PROCESS AND PRODUCT QUALITY ASSURANCE (PPQA)

Process and Product Quality Assurance (PPQA) is the main SQA, Software Quality Assurance process area within CMMI. Although there are many definitions of Software Quality Assurance SQA, its main function within CMMI (under Process and Product Quality Assurance PPQA) is centred on conformance and compliance to previously defined process descriptions, standards and procedures. There is often confusion between SQA and software testing (which is actually a part of Software Quality Control SQC); this resource will clarify the difference between SQA and SQC. Process and Product Quality Assurance (PPQA) within CMMI is often the stimuli for process improvement initiatives as PPQA identifies non-conforming products, defects, and (using Causal Analysis and Resolution, CAR) the activities that caused these defects can be identified.The purpose of Process and Product Quality Assurance (PPQA) is to provide staff and management with objective insight into processes and associated work products.

The Process and Product Quality Assurance process area involves the following:

- Objectively evaluating performed processes, work products and services against the applicable process descriptions, standards and procedures.
- Identifying and documenting noncompliance issues.

- Providing feedback to project staff and managers on the results of quality assurance activities.
- Ensuring that noncompliance issues are addressed.

The Process and Product Quality Assurance process area supports the delivery of high-quality products and services by providing the project staff and managers at all levels with appropriate visibility into, and feedback on processes and associated work products throughout the life of the project. The practices in the Process and Product Quality Assurance process area ensure that planned processes are implemented, while the practices in the Verification process area ensure that the specified requirements are satisfied. These two process areas may on occasion address the same work product but from different perspectives. Projects should take advantage of the overlap in order to minimize duplication of effort while taking care to maintain the separate perspectives.

Objectivity in process and product quality assurance evaluations is critical to the success of the project. Objectivity is achieved by both independence and the use of criteria. A combination of methods providing evaluations against criteria by those not producing the work product is often used. Less formal methods can be used to provide broad day-to-day coverage. More formal methods can be used periodically to assure objectivity. Examples of ways to perform objective evaluations include the following:

- Formal audits by organizationally separate quality assurance organizations.
- Peer reviews which may be performed at various levels of formality.
- In-depth review of work at the place it is performed (i.e., desk audits).
- Distributed review and comment of work products.

Traditionally, a quality assurance group that is independent of the project provides this objectivity. It may be appropriate in some organizations, however, to implement the process and product quality assurance role without that kind of independence. For example, in an organization with an open, quality-oriented culture, the process and product quality assurance role may be performed, partially or completely, by peers; and the quality assurance function may be embedded in the process. For small organizations, this might be the most feasible approach. If quality assurance is embedded in the process, several issues must be addressed to ensure objectivity. Everyone performing quality assurance activities should be trained in quality assurance. Those performing quality assurance activities for a work product should be separate from those directly involved in developing or maintaining the work product. An independent reporting channel to the appropriate level of organizational management must be available so that noncompliance issues can be escalated as necessary.

For example, in implementing peer reviews as an objective evaluation method:

- Members are trained and roles are assigned for people attending the peer reviews.
- A member of the peer review who did not produce this work product is assigned to perform the role of QA.
- Checklists are available to support the QA activity.
- Defects are recorded as part of the peer review report and are tracked and escalated outside the project when necessary.

Quality assurance should begin in the early phases of a project to establish plans, processes, standards, and procedures that will add value to the project and satisfy the requirements of the project and the organizational policies. Those performing quality assurance participate in establishing the plans, processes, standards, and procedures to ensure that they fit the project's needs and that they will be useable for performing quality

assurance evaluations. In addition, the specific processes and associated work products that will be evaluated during the project are designated. This designation may be based on sampling or on objective criteria that are consistent with organizational policies and project requirements and needs. When noncompliance issues are identified, they are first addressed within the project and resolved there if possible. Any noncompliance issues that cannot be resolved within the project are escalated to an appropriate level of management for resolution.

This process area applies primarily to evaluations of the activities and work products of a project, but it also applies to evaluations of non-project activities and work products such as training activities. For these activities and work products, the term "project" should be appropriately interpreted.

5. CONCLUSION

There are so many reasons why a company should consider SQA. It is all about business survival and SQA is just one of the many tools the company should effectively use. And just like a tool it has to be effectively used to its maximum. If the tool is not used to its full extent the tool will just be a financial burden to the company. An important aspect of SQAs involvement with process improvement is that they (SQA) only take the measures back to the process owners and it is the process owners (i.e. the design team) that suggest a new standard or procedure. This new standard or procedure is then used in future projects. The role of SQA is even more removed from development; they are mainly providing the role of an Auditor. In addition SQA will collect measurements of the effectiveness (and cost) of the processes in order to implement continuous process improvement. One of the most important parameters that need to be managed as part of a project is software "Quality". This involves good Quality Planning, Quality Control and Quality Assurance practices to be in place. Organizations that understand their needs and drive their improvements accordingly are more likely to obtain significant improvements in performance. CMMI is an approach to process improvement, in which SQA plays a major role. Every software development organization takes part in both the CMMI processes and any improvement initiatives for those processes.The CMMI (Capability Maturity Model Integration) model is widely used to implement quality assurance in an organization. So to achieve CMM level, company has to follow all standards and procedures of SQA. Application of more highly elaborated software quality management methods increases the organization's capability to control quality and improve software process productivity.

REFERENCES

[1] IEEE Standard, IEEE Standard for Software QualityAssurance Plans, IEEE 730-2002.

[2] A. Abram, J. W. Moore, Guide to Software Engineering Body of Knowledge, IEEE Computer Society, USA,2004.

[3] ESA Board for Software Standardization and Control, Guide to software quality assurance, ESA Publications Division, Noordwijk, The Netherlands, 1995.

[4] M.B. Chrissis, M. Konrad, S. Shrum, CMMI: Guide for Process Integration and Product Improvement, Addison-Wesley, Boston, 2004.

[5] "The Capability Maturity Model – Guidelines for Improving the Software Process" Carnegie Melon University, SEI (Pearson Education Series)

[6] "Practices of High Maturity Organizations" – Mark C Paulk 1999 SEPG Conf, Atlanta.

[7] Felschow, A. (1999) "Understanding the Capability Maturity Model (CMM) and the role of SQA in the software development maturity", in G. G. Schulmeyer and J. I. McManus (eds), Handbook of Software Quality Assurance, 3rd edition, Prentice Hall, Upper Saddle River, NJ, pp. 329–350.
[8] Schulmeyer, G. G. (1999) "Standardization of software quality assurance –where is it all going?", in G. G. Schulmeyer and J. I. McManus (editions), Handbookof Software Quality Assurance, 3rd edn, Prentice Hall, Upper Saddle River, NJ, pp. 91–113.
[9] Paulk, M. C., Weber, C. V. Curtis, B., Chrissis, M. B. (1995) The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley, Reading, MA.
[10] ElEmam, K. (1998) The Internal Consistency of the ISO/IEC 15504 Software Process Capability Scale, International Software Engineering Research NetworkTechnical Report ISEERN-98-06.