SOFTWARE QUALITY ASSURANCE IN CAPABILITY MATURITY MODEL INTEGRATION

Rajnipriya Dhawan

Information Technology, DAV Institute of Management, Faridabad, (India)

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 QualityAssurancepractices in the organization to assure "Quality" of Software ProductDevelopment. 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**

I. 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 taskof 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 roles range from acting as an extension of development for debugging software products, to development process 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), CarnegieMellon University’s SEI took theinitial steps toward development of what is termed a Capability Maturity Model(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 situation where 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.

II. 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 appropriatevisibility 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 **TableI**. For each level of maturity, the roles forthe software quality group are shown.

Table I: Organization MaturityandSQA 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

III. THE STRUCTURE ANALYSIS OF CMMI

There are two different kinds of representations. The first is continuous representation, which is utilized in measuring project capability of an enterprise. For its limitation on contents used in continuous capability appraisal, thus, it only represent that enterprise has achieve certain level of implementing capacity in specific project. While, the second representing method is periodic method, it is applied in measuring the maturity of enterprise, i.e. the comprehensive capacity of enterprise software development. Based on periodic method, the contents of appraisal are randomly selected by experts from any project or even any stage of any project, thus, comparing with continuous capability appraisal method, it is more objective and comprehensive, however, they are similar in the nature. This essay mainly study on the basis of continuous representation of CMMI.

This paper mainly focuses on CMMI managed level to design measurement index system and to provide morefeasible and effective assessing method for a large number of software developing enterprises at level 2 or below. CMMI include 5 Maturity levels, 22 key process areas. The details are shown in the Table II. Besides, the latest CMMI model recruits Integrated Teaming, Organizational Environment for Integration and Integrated Suppliers Management process areas, however, this 3 process areas target large software enterprises and software enterprises need managing large amount of suppliers, and those needs are not apply in small and middle sized software enterprises. Thus, this essay will not refer to those three process areas.

Table II Maturity Levels

Maturity Level	**Key Process Area**
5 Optimizing	OID: Organizational Innovation and Deployment
	CAR: Causal Analysis and Resolution
4 QuantitativelyManaged	OPP: Organizational Process Performance
	QPM: Quantitative Project Management
3 Defined	RD: Requirements Development
	TS: Technical Solution
	PI: Project Integration
	Ver: Verification
	Val: Validation
	OPF: Organizational Process Focus
	OPD: Organizational Process Definition
	OT: Organizational Training
	IPM: Integrated Project Management
	RskM: Risk Management
	DAR:Decision Analysis and Resolution
2 Managed	ReqM: Requirements Management
	PP: Project Planning
	PMC: Project Monitoring and Control
	SAM: Supplier Agreement Management
	MA: Measurement and Analysis
	PPQA: Process and Product Quality Assurance
	CM: Configuration Management
1 Initial	

Maturity levels advance gradually in due order, if need to achieve certain maturity level, take CMMI level 3 for instance, in addition to satisfy the total 11 process areas of CMMI level 3, the 7 process areas of level 2 should also be satisfied, and so on.

Every process area includes some specific goals and generic goals, systematically, every specific goal contains several specific practices and, every generic similarly involves several generic practices. The interrelationship among those process areas, goals and practices are shown in Figure 1.

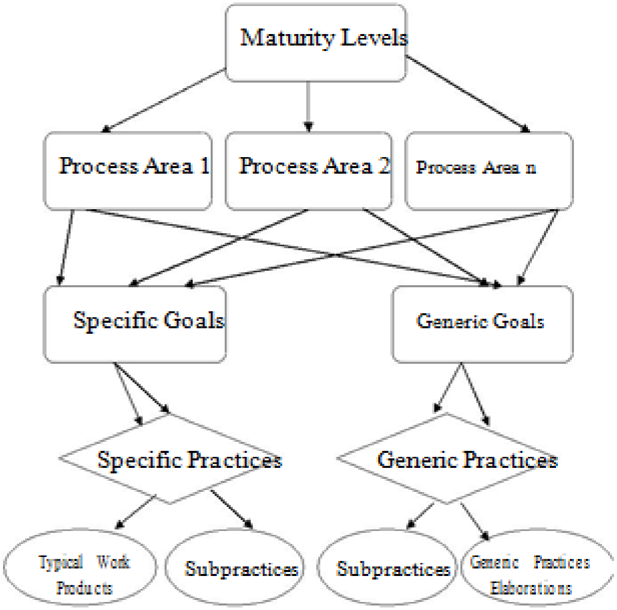

Figure 1. CMMI Structure Diagram

From the diagram above, to evaluate the maturity of software enterprises, all the process areas of relevant level and bellow need to be assessed, so that to evaluate which level that the enterprise has achieved, ultimately the detail operation (to be more specific, typical work products and sub-operations etc.) has been assessed whether the needs have been satisfied.

The index system of measuring software process quality, which is described in this paper, is mainly focus on the specific goals and generic goals in maturity level 2. The appraisal indexes are selected properly from goals within this level and determine weight of every index is the priority in creating quantitative appraisal model.Within the second level (managed level) of CMMI, 7 key process areas, 15 specific goals and 1 generic goal are involved. Those who need a specification is, every process area include a generic goal, however, this essay set the amount of generic goal as 1, which is due to the reason that generic goal essentially is the priority before achieve several specific goals, therefore, the generic goals in different processes are substantially the same. The detailed relationship of key process areas and goal items in managed levels is shown as Table III.

Table III Key Process Areas and Goal Items

KPA	**Specific Goal(SG)/Generic Goal(GG)**
RequirementsManagement (REQM)	SG1:Manage Requirements
Project Planning (PP)	SG2 : Establish Estimates
	SG3: Develop a Project Plan
	SG4 : Obtain Commitment to the Plan
Project Monitoringand Control (PMC)	SG5: Monitor Project Against Plan
	SG6: Manage Corrective Action to Closure
Supplier AgreementManagement (SAM)	SG7: Establish Supplier Agreements
	SG8: Satisfy Supplier Agreements
Measurement andAnalysis (MA)	SG9 : Align Measurement and Analysis Activities
	SG10: Provide Measurement Results
Process and Product Quality Assurance(PPQA)	SG11: Objectively Evaluate Processesand Work Products
	SG12: Provide Objective Insight
ConfigurationManagement (CM)	SG13: Establish Baselines
	SG14: Track and Control Changes
	SG15: Establish Integrity
	GG1: Systematize Managing Process

IV. SOFTWAREQUALITY ASSURANCE AS PER CMMIPROCESS AREA

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

4.1 Requirements Development
4.2 Requirements Management
4.3 Product Integration
4.4 Verification
4.5 Validation

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. butonly 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.

IV. 1SQA IN CMMI REQUIREMENTS DEVELOPMENT

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

SQA role: To observe (audit) thatdocumented standards, processes, andprocedures 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 characteristicsdiscussed here.The respective roles ofSQC 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.

4.1 SQA In Cmmi Requirementsmanagement

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 therequirements 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) thatdocumented 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.

4.2 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) thatdocumented standards, processes, andprocedures 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.

4.3 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) thatdocumented 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.

4.4 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.

4.5 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 usedinternally, 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.

V. 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 QualityPlanning, 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. TheCMMI (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] Abram, J. W. Moore, Guide to Software Engineering Body ofKnowledge, IEEE Computer Society, USA,2004.

[3] ESA Board for SoftwareStandardization and Control, Guide to software quality assurance, ESA

[4] Publications Division, Noordwijk,TheNetherlands, 1995.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.

[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.

[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] Rekha Chouhan, Dr.Rajeev Mathur (August2012) International Journal of Advanced Research in Computer Engineering & Technology,Volume1,Issue 6. Role of Software Quality Assurance in Capability Maturity Model Integration.