# Unit 10 Software Quality Assurance

**Structure:**



## 10.1 Introduction

In the last unit, we have discussed team organization and conflict management. In this unit we shall discuss yet another important topic viz. Software Quality Assurance.

Software developers continue to believe that software quality is something you begin to worry about after code has been generated. Nothing could be further from the truth. *Software quality assurance* (SQA) is an umbrella activity that is applied throughout the software process which encompasses various activities such as technical reviews, documentation, software compliance etc.. Now let's discuss SQA in detail.

**Objectives:**

After studying this unit, you should be able to:

- explain various software quality assurance activities
- discuss the characteristics of a quality software
- describe various quality standards

## 10.2 Software Quality Assurance Activities

Software Quality Assurance (SQA) encompasses (1) a quality management approach, (2) effective software engineering technology (methods and tools), (3) formal technical reviews that are applied throughout the software

process, (4) a multi-tiered testing strategy, (5) control of software documentation and the changes made to it, (6) *a* procedure to ensure compliance with software development standards (when applicable), and (7) measurement and reporting mechanisms.

Software quality assurance is related to the practice of quality assurance in product manufacturing. There are, however, some notable differences between software and a manufactured product. These differences all stem from the fact that the manufactured product is physical and can be seen whereas the software product is not visible. Therefore its function, benefit and costs are not as easily measured.

SQA is complicated by the complex nature of software quality – an attribute of computer programs that is defined as "conformance to explicitly and implicitly specified requirements." But when considered more generally, software quality encompasses many different product and process factors and related metrics.

To properly conduct software quality assurance, data about the software engineering process should be collected, evaluated, and disseminated. Statistical SQA helps to improve the quality of the product and the software process itself. Software reliability models extend measurements, enabling collected defect data to be extrapolated into projected failure rates and reliability predictions.

Product evaluation and process monitoring are the SQA activities that assure the software development and control processes described in the project's Management Plan are correctly carried out and that the project's procedures and standards are followed. Products are monitored for conformance to standards and processes are monitored for conformance to procedures. Audits are a key technique used to perform product evaluation and process monitoring. Review of the Management Plan should ensure that appropriate SQA approval points are built into these processes.

Product evaluation is an SQA activity that assures standards are being followed. Ideally, the first products monitored by SQA should be the project's standards and procedures. SQA assures that clear and achievable standards exist and then evaluates compliance of the software product to the established standards. Product evaluation assures that the software

product reflects the requirements of the applicable standard(s) as identified in the Management Plan.

Process monitoring is an SQA activity that ensures that appropriate steps to carry out the process are being followed. SQA monitors processes by comparing the actual steps carried out with those in the documented procedures. The Assurance section of the Management Plan specifies the methods to be used by the SQA process monitoring activity.

A fundamental SQA technique is the audit, which looks at a process and/or a product in depth, comparing them to established procedures and standards. Audits are used to review management, technical, and assurance processes to provide an indication of the quality and status of the software product.

The purpose of an SQA audit is to assure that proper control procedures are being followed, that required documentation is maintained, and that the developer's status reports accurately reflect the status of the activity. The SQA product is an audit report to management consisting of findings and recommendations to bring the development into conformance with standards and/or procedures.

**Self Assessment Questions**

1. Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process. (True / False)
2. _____ is an SQA activity that assures standards are being followed.
3. _____ is an SQA activity that ensures that appropriate steps to carry out the process are being followed.

## 10.3 Software Qualities

Qualities for which an SQE (Software Quality Engineering) evaluation is to be done must first be selected and requirements set for them. Some commonly used qualities are reliability, maintainability, transportability, interoperability, testability, usability, reusability, traceability, sustainability, and efficiency. Some of the key qualities are discussed below.

**1) Reliability**

Hardware reliability is often defined in terms of the Mean-Time-To-Failure, or MTTF, of a given set of equipment. An analogous notion is useful for

software, although the failure mechanisms are different and the mathematical predictions used for hardware have not yet been usefully applied to software.

***Definition: Software reliability***

> Software reliability is often defined as the extent to which a program can be expected to perform intended functions with required precision over a given period of time. Even more, it is concerned with techniques to compensate for unknown software errors and for problems in the hardware and data environments in which the software must operate.

**2) Maintainability**

It is analogous to the hardware quality of Mean-Time-To-Repair, or MTTR. While there is as yet no way to directly measure or predict software maintainability, there is a significant body of knowledge about software attributes that make software easier to maintain. These include modularity, self (internal) documentation, code readability, and structured coding techniques. These same attributes also improve sustainability, the ability to make improvements to the software.

***Definition: Software maintainability***

> Software maintainability is defined as the ease of finding and correcting errors in the software.

**3) Transportability**

Transportability is defined as the ease of transporting a given set of software to a new hardware and/or operating system environment.

**4) Interoperability**

Software interoperability is the ability of two or more software systems to exchange information and to mutually use the exchanged information.

**5) Efficiency**

Efficiency is the extent to which software uses minimum hardware resources to perform its functions.

**Software Metrics**

Software Metrics are quantitative values, usually computed from the design or code, that measures the quality in question, or some attribute of the

software related to the quality. Many metrics have been invented, and a number have been successfully used in specific environments, but none has gained widespread acceptance.

There are many other software qualities. Some of them will not be important to a specific software system, thus no activities will be performed to assess or improve them. Maximizing some qualities may cause others to be decreased. For example, increasing the efficiency of a piece of software may require writing parts of it in assembly language. This will decrease the transportability and maintainability of the software.

**Self Assessment Questions**

4. Software Reliability is measured in Mean-Time-To-Failure (MTTF). (True / False)
5. _____ is defined as the ease of finding and correcting errors in the software.
6. _____ is the ability of two or more software systems to exchange information and to mutually use the exchanged information.

## 10.4 Standards

Establishing standards and procedures for software development is critical, since these provide the framework from which the software evolves. Standards are the established criteria to which the software products are compared. Procedures are the established criteria to which the development and control processes are compared. Standards and procedures establish the prescribed methods for developing software; the SQA role is to ensure their existence and adequacy. Proper documentation of standards and procedures is necessary since the SQA activities of process monitoring, product evaluation and auditing rely upon unequivocal definitions to measure project compliance. We will discuss the ISO, CMM standards here.

### 10.4.1 ISO – Standards for software organization

ISO 9000 is a set of quality management standards, recognized worldwide, developed and controlled by the International Organization for Standardization (ISO) in Geneva, Switzerland.

The documents that define the ISO 9000 standard, adopted by more than 100 countries, provide a quality management system philosophy and guidance, as well as specifications to which the quality system must adhere.

This means that Avocent has established a systematic approach to quality management, and is managing its business to ensure that your needs are clearly understood, agreed and fulfilled. ISO 9001 (to which Avocent is certified) is the most stringent of the ISO 9000 standards. ISO 9001 requires that a company meet a set of requirements aimed primarily at achieving quality and customer satisfaction at all program stages from design through servicing. Avocent established a quality policy requiring support and "buy-in" from all staff levels. Our employees ensure that we continue to meet our quality objectives.

### 10.4.2 CMM (Capability Maturity Model)

**CMM (Capability Maturity Model)** is a model of process maturity for software development – an evolutionary model of the progress of a company's abilities to develop software. In November 1986, the American ***Software Engineering Institute (SEI)*** in cooperation with Mitre Corporation created the Capability Maturity Model for software. Development of this model was necessary so that the U.S. federal government could objectively evaluate software providers and their abilities to manage large projects. Many companies had been completing their projects with significant overruns in schedule and budget. The development and application of CMM helps to solve this problem. The key concept of the standard is organizational maturity. A mature organization has clearly defined **procedures** for software development and project management. These procedures are adjusted and perfected as required. In any software development company there are **standards for processes** of development, testing, and software application; and rules for appearance of final program code, components, interfaces, etc. Figure 10.1 shows various levels in CMM.

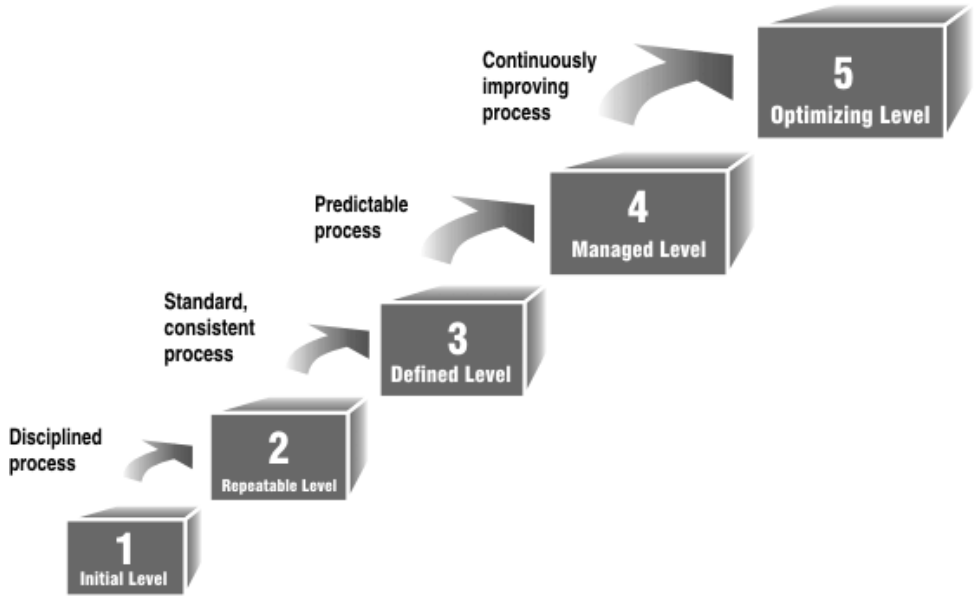


**Fig. 10.1: The CMM model defines five levels of organizational maturity**

1) **Initial level:** This is a basis for comparison with the next levels. In an organization at the initial level, conditions are not stable for the development of quality software. The results of any project depend totally on the manager's personal approach and the programmers' experience, meaning the success of a particular project can be repeated only if the same managers and programmers are assigned to the next project. In addition, if managers or programmers leave the company, the quality of produced software will sharply decrease. In many cases, the development process comes down to writing code with minimal testing.
2) **Repeatable level:** At this level, project management technologies have been introduced in a company. That project planning and management is based on accumulated experience and there are standards for produced software (these standards are documented) and there is a special quality management group. At critical times, the process tends to roll back to the initial level.
3) **Defined level:** Here, standards for the processes of software development and maintenance are introduced and documented (including project management). During the introduction of standards, a transition to more effective technologies occurs. There is a special

quality management department for building and maintaining these standards. A program of constant, advanced training of staff is required for achievement of this level. Starting with this level, the degree of organizational dependence on the qualities of particular developer decreases and the process does not tend to roll back to the previous level in critical situations.

4) **Managed level:** In this level there are quantitative indices (for both software and process as a whole) established in the organization. Better project management is achieved due to the decrease of digression in different project indices. However, sensible variations in process efficiency may be different from random variations (noise), especially in mastered areas.
5) **Optimizing level:** In this level improvement procedures are carried out not only for existing processes, but also for evaluation of the efficiency of newly introduced innovative technologies. The main goal of an organization on this level is permanent improvement of existing processes. This should anticipate possible errors and defects and decrease the costs of software development, by creating reusable components for example.

### 10.4.3 Comparison between ISO 9001 & SEI-CMM

Clearly there is a strong correlation between ISO 9001 and the CMM, although some issues in ISO 9001 are not covered in the CMM, and some issues in the CMM are not addressed in ISO 9001. The levels of detail differ significantly; chapter 4 in ISO 9001 is about five pages long; sections 5, 6, and 7 in ISO 9000-3 comprise about 11 pages; and the CMM is over 500 pages long. There is some judgment involved in deciding the exact correspondence, given the different levels of abstraction.

The clauses in ISO 9001 with no strong relationships to the CMM key process areas, and which are not well addressed in the CMM, are purchaser-supplied product and handling, storage, packaging and delivery. The clause in ISO 9001 that is addressed in the CMM in a completely distributed fashion is servicing. The clauses in ISO 9001 for which the exact relationship to the CMM is subject to significant debate are corrective action and statistical techniques.

The biggest difference, however, between these two documents is the emphasis of the CMM on continuous process improvement. ISO 9001 addresses the minimum criteria for an acceptable quality system. It should also be noted that the CMM focuses strictly on software, while ISO 9001 has a much broader scope: hardware, software, processed materials, and services.

The biggest similarity is that for both the CMM and ISO 9001, the bottom line is “Say what you do; do what you say.” The fundamental premise of ISO 9001 is that every important process should be documented and every deliverable should have its quality checked through a quality control activity. ISO 9001 requires documentation that contains instructions or guidance on what should be done or how it should be done.

The CMM shares this emphasis on processes that are documented and practiced as documented. Phrases such as conducting “according to a documented procedure” and following “a written organizational policy” characterize the key process areas in the CMM.

The CMM also emphasizes the need to record information for later use in the process and for improvement of the process. This is equivalent to the quality records of ISO 9001 that document whether or not the required quality is achieved and whether or not the quality system operates effectively. This statement is controversial in itself. Some members of the international standards community maintain that if you read ISO 9001 with insight (between the lines so to speak), it does address continuous process improvement. There is faith that weaknesses will improve over time, especially given regular surveillance audits. Corrective action can be interpreted in this way, although that may not be consistently done today. This will undoubtedly be one of the major topics for the next revision cycle for ISO 9001.

### 10.4.4 Other standards

Now let’s see various other standards available for measuring software quality.

**1) ANSI**

ANSI stands for the ***American National Standards Institute***, a general standards organization in the United States that facilitates the voluntary establishment of standards for many areas, including computing. The staffs

at ANSI don't create standards; they coordinate with organizations in the US to provide a neutral forum for the development of standards.

**2) CMMI**

The CMMI is the successor of the CMM. The CMM was developed from 1987 until 1997. In 2002 version 1.1 of the CMMI was released: v1.2 followed in August 2006. The goal of the CMMI project is to improve usability of maturity models for software engineering and other disciplines, by integrating many different models into one framework. It was created by members of industry, government and the SEI. The main sponsors included the Office of the Secretary of Defense (OSD) and the National Defense Industrial Association (NDIA) Systems Engineering Committee.

***Capability Maturity Model® Integration (CMMI)*** is a process improvement approach that provides organizations with the essential elements of effective processes. The latest version of CMMI ver 1.2 was released in August 2006. There are 3 constellations of CMMI in the new version. They are:

- CMMI Development
- CMMI Services
- CMMI Acquisition.

CMMI for Development v1.2 consists of 22 process areas with capability or maturity levels. CMMI is created by the Software Engineering Institute (SEI) and is available for download from the SEI.

CMMI should be adapted to each individual company; therefore companies are not "certified." A company is appraised (e.g. with an appraisal method like SCAMPI) at a certain level of CMMI. The results of such an appraisal can be published if released by the appraised organization.

**3) IEEE**

The IEEE, a non-profit organization, is the world's leading professional association for the advancement of technology. The full name of the IEEE is the ***Institute of Electrical and Electronics Engineers, Inc***., although the organization is referred to by the letters IEEE and pronounced Eye-triple-E.

**Self Assessment Questions**

7. Standards are the established criteria to which the software products are compared. (True / False)
8. SEI-CMM stands for _____.
9. ANSI stands for _____.

## 10.5 Summary

Let's recapitulate important points:

- Software quality is an umbrella activity that is applied throughout the software processes. This would encompass formal technical reviews, tools, testing strategy and software documentation.
- The current unit highlighted the software quality metrics, namely lines of code, number of modules and number of interfaces etc.
- A metric is measurable quantity.
- Standards and procedures are also essential elements in this scenario. Because, standards are established criteria for to which software products are compared and procedures are established criteria to which development and control are compared.
- Some of the popular standards include ISO, CMM, CMMI and IEEE.

## 10.6 Terminal Questions

1. What are the various software quality assurance activities?
2. What are the various measurable software qualities? Explain.
3. What is ISO? Explain its significance in the context of software industries.
4. What is CMM? Explain its various levels.
5. Compare and contrast ISO and CMM.
6. What are other software standards available apart from ISO and CMM?

## 10.7 Answers

### Self Assessment Questions

1. True
2. Product evaluation
3. Process monitoring
4. True
5. Software maintainability
6. Software interoperability
7. True
8. Software Engineering Institute – Capability Maturity Model
9. American National Standards Institute

**Terminal Questions**

1. Software Quality Assurance (SQA) activities includes the following:
   a. A quality management approach
   b. Effective software engineering technology (methods and tools)
   c. Formal technical reviews that are applied throughout the software process
   d. A multi-tiered testing strategy
   e. Control of software documentation and the changes made to it
   f. A procedure to ensure compliance with software development standards (when applicable)
   g. Measurement and reporting mechanisms. (Refer Section 10.2 for detail)

2. Some commonly used software qualities are reliability, maintainability, transportability, interoperability, testability, usability, reusability, traceability, sustainability, and efficiency. (Refer Section 10.3)

3. ISO 9000 is a set of quality management standards, recognized worldwide, developed and controlled by the International Organization for Standardization (ISO) in Geneva, Switzerland. The documents that define the ISO 9000 standard, adopted by more than 100 countries, provide a quality management system philosophy and guidance, as well as specifications to which the quality system must adhere. (Refer Sub-section 10.4.1)

4. CMM (Capability Maturity Model) is a model of process maturity for software development – an evolutionary model of the progress of a company's abilities to develop software. In November 1986, the American Software Engineering Institute (SEI) in cooperation with Mitre Corporation created the Capability Maturity Model for software. Development of this model was necessary so that the U.S. federal government could objectively evaluate software providers and their abilities to manage large projects. (Refer Sub-section 10.4.2)

5. The biggest difference between ISO and CMM documents is the emphasis of the CMM on continuous process improvement. ISO 9001 addresses the minimum criteria for an acceptable quality system. Also CMM focuses strictly on software, while ISO 9001 has a much broader scope: hardware, software, processed materials, and services. The

biggest similarity is that for both the CMM and ISO 9001, the bottom line is “Say what you do; do what you say.” The fundamental premise of ISO 9001 is that every important process should be documented and every deliverable should have its quality checked through a quality control activity. ISO 9001 requires documentation that contains instructions or guidance on what should be done or how it should be done. (Refer Sub-section 10.4.3)

6. The other most popular standards apart from ISO and CMM are ANSI, CMMI and IEEE. (Refer Section 10.5)