BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 6

DCA3245

SOFTWARE PROJECT MANAGEMENT

Unit 10

Software Quality Assurance

Table of Contents

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 (SQA).

A general perception of software development team is that quality is something that we need to do after completion of coding, and the responsibility of quality assurance rallies with testing team. It is totally incorrect. Quality assurance is responsible of entire team and the quality assurance is an umbrella activity that is applied throughout the lifecycle of the project across all phases. It not only involves testing of the code that is developed, but also includes activities such as technical reviews, documentation, compliance etc. Now let’s discuss SQA (Software Quality Assurance) in detail.

1.1 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*

2. SOFTWARE QUALITY ASSURANCE ACTIVITIES

Software Quality Assurance (SQA) is a continuous activity carried throughout the project lifecycle. It comprises of

1. An approach that determines the technique that need to be adopted for managing the quality in the project
2. Software engineering methodologies and tools
3. Technical reviews that need to be applied throughout the software processes
4. Testing strategy for the project
5. Documentation mechanism and effective ways of controlling the changes to the documents
6. Procedure for software compliance when needed to be obtained
7. Metrics gathering and reporting mechanisms

In the broad sense, the software quality assurance is comparable with similar practices followed in product manufacturing. However, there are some major changes as well between software project and product manufacturing in terms of quality assurance. One obvious reason for the difference is that manufactured product is physical and can be seen and felt; hence the cost and benefit can be easily measured. The same does not apply to software product.

When a product is of high quality, its specifications are met during development.

The term "quality" refers to the qualities or attributes of a product, with "attributes" meaning those details that may be measured and compared to industry norms.

The term "quality" in the context of software testing refers to the extent to which a software product or system satisfies requirements and delivers on user expectations. It includes several factors that add up to the software's overall quality or dependability.

Superiority comes in two flavours:

1. The term "design quality" is used to describe an object's conformance to the criteria established by its creators. Quality of design is affected by factors such as material grade, tolerances, and performance requirements.
2. Quality of conformity refers to how well the final product matches the original design. The higher the conformity degree, the greater the conformity quality.

Software Quality Assurance:

As stated earlier, SQA runs through the entire life cycle of the project and is a complicated and complex process. The main aim of the SQA is to ensure that the highest quality is maintained for the computer programs that are developed. By definition, quality is "conformance to explicitly and implicitly stated requirements for the software product". Hence the activity covered in the SQA tries to improve upon the processes that develop the software product.

In order to properly execute the software quality assurance processes, we need to have the data. The data that tells about the effectiveness of the current practices followed in the project. Hence the data need to defined, collected, evaluated, and analyzed. This statistical analysis helps the SQA to be more predictable and reliable.

The two major activities that determine the success of the SQA are the product evaluation and monitoring of the process. The project plan describes about the goals of the project and the activities to be carried out in order to achieve the goal. Now, the SQA will help in ensuring that the activities and the procedures stated in the project plan are carried out correctly to develop the product properly and also the developed product is evaluated to confirm that the product is as per the specs and the goal of the project is met. Proper reviews of the plan and regular audits of the process are highly important for the success of the project.

SQA relies on careful forethought and the execution of a wide range of procedures that are interwoven across the many phases of developing software.

- The right approach to development processes is in place.
- Independent evaluations and audits are carried out to ensure that the projects are following the established standards and protocols.

- The documentation is created to help with the upkeep and improvement.
- Mechanisms are in place and used to control modifications, and documentation is created as needed during development rather than after.
- All high-risk product areas are given extra attention during testing;

Goal of Software Quality Assurance:

Software quality assurance (SQA) aims to guarantee that finished software is up to par in all respects. Through the use of methodical and organised procedures, SQA hopes to improve both the software development cycle and the final product. The main objectives of SQA are as follows:

- SQA verifies that the software satisfies all requirements described in the project documentation to guarantee compliance. Verifying that the software does what it's supposed to and meets user requirements.
- Preventing bugs from appearing in the first place is one of SQA's main goals. Defects can be avoided altogether with the aid of SQA's practises such code reviews, static analysis, and comprehensive documentation.
- To improve the software development process itself is one of SQA's primary goals. In order to boost productivity and efficiency, it is necessary to improve development processes, codify best practises, and adopt quality standards.
- Standardisation and Consistency: SQA ensures that all members of a team or organisation adhere to the same standards when it comes to development. This aids in creating consistent quality across all software initiatives.
- Programme Quality Assurance (SQA) performs both verification (checking that the programme conforms to the specifications) and validation (making sure the product satisfies the requirements of its intended users). This requires extensive testing and validation.

Role of SQA: When SQA is appropriately staffed, reports to an autonomous management structure, and understands its mission as assisting the development and maintenance teams in raising product quality, it can be a useful tool.

- Check the completeness of all development and quality plans;
- Assist with design and code inspections as moderators;
- Check the test procedures for conformity with specifications;
- Analyse a sizable subset of all test data to make sure everything is going according to plan;
- Perform routine audits of supply chain management operations to ensure conformity to benchmarks;
- Take part in all reviews of the project and report any conflicts of interest if proper procedures and standards have not been followed.

Now, let's look at the product evaluation. In SQA's context we define all the procedures and standards defined are also considered methods for developing software. Hence the product evaluation ensures that the standards and procedures defined for the project are followed properly and follows the compliance as stated. Finally the product evaluation will also assure that the software product developed is as per specified requirements through the different testing strategies.

Process monitoring on the other hand deals with the activities that are carried out during the project execution. Process monitoring is an SQA activity that ensures that appropriate steps to carry out the process are being followed. The process monitoring in SQA compares the actual steps that are carried out during the project execution against the documented steps to record and correct any deviation in the practice. For this purpose the Audits are carried out regularly. The Audit looks at the process and product development in depth. The goal of the audit is to compare the actual practice followed in the process implementation against the established procedures and standards. Audits help to review management, technical, and compliance processes to obtain the status of the project and provide indication on the quality of the product being developed.

SQA Management Plans:

An essential document for ensuring the quality of a software product throughout its development life cycle is the Software Quality Assurance (SQA) Management Plan. This plan details the strategy, processes, and activities that will be implemented to achieve this goal. It is a resource for the whole development team and offers a framework for applying quality assurance practises. An SQA Management Plan will typically consist of the following main elements:

- Describes the document's background, scope, and target readers. The value of SQA to the project is described.
- Describes the development team's commitment to a set of quality standards, guidelines, and best practises. Some examples of this are recommended coding practises, design principles, and testing methods.
- Come up with a SQA strategy that will be used all through the project.
- Determine the most appropriate software engineering tasks for the current project.
- Verify SQA team competence
- During the planning phase of the project, the plan is created and reviewed by all stakeholders involved.

SQA Activities:

- The SEI recommends a set of SQA activities that address quality assurance
- SQA Management Plan
- Set The Check Points
- Multi testing Strategy
- Measure Change Impact
- Manage Good Relations

The plan governs the software quality assurance (SQA) activities carried out by the software engineering team and the SQA team group. In the strategy, we've pinpointed: There Will Be an Evaluation

- There Will Be Audits And Reviews Done
- Criteria that are relevant to the work at hand
- Error tracking and reporting protocols
- The SQA group's upcoming paperwork
- Quantity of comments given to the software development team

Observes documented processes for recording and responding to software project-related changes. Document any infractions and send a report up the chain of command. Maintaining positive working relationships with other groups contributing to a project's progress is essential in the workplace. If the SQA team's communication with the development team is poor, the project will suffer as a direct result.

Concern at Organisational Level: The goal of quality management is to produce consistently high-quality software by building a set of organisational processes and standards. Quality management at the process level is carrying out quality processes as specified in documentation and checking to see if they were carried out as intended. Establishing a quality plan is a primary focus of quality management at the project level. Quality objectives and required procedures and criteria should be laid forth in the project's quality plan.

Concern at Process Level: Applying quality processes that have been established by management and checking to make sure they have been followed are essential parts of quality management.

Concern at Project Level: Establishing a quality plan is a primary focus of quality management. Quality objectives and required procedures and criteria should be laid forth in the project's quality plan.

Benefits of SQA Activities: Software quality assurance (SQA) operations are essential to the efficient completion of software development projects. They help get top-notch software out the door that satisfies customers and the bottom line.

- The quality of software is enhanced via SQA.
- Software quality is a strong suit for SQA.
- Good applications are money and time savers.
- SQA increases dependability.
- With SQA, you can go for much longer without doing any upkeep.
- Profitability of businesses is improved by using high-quality commercial software facilitates a better method for developing software

SELF-ASSESSMENT QUESTIONS - 1

1. Software quality assurance (SQA) is an umbrella activity that is applied throughout the lifecycle of the project across all phases. (True / False)
2. ______________ is an SQA activity that assures standards and procedures defined for the project are followed properly and follows the compliance as stated.
3. ______________ is an SQA activity that ensures that appropriate steps to carry out the process are being followed.

3. SOFTWARE QUALITIES

Just like the requirements gathered for a software product being developed, for the SQA, the qualities that need to be maintained during the course of the project should be determined first. 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

Reliability gives an indication on the susceptibility of the product to Failure. For the hardware product reliability is described in terms of Mean-Time-To-Failure. For Software product also similar terminology can be applied; although the nature of failure here is different and also the accuracy with which hardware reliability can be predicted is not yet available for software product.

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

Similar to the reliability, the maintainability can be attributed to MTTR (Mean-Time-To-Repair). Here also the MTTR is applied accurately for hardware products, the software counterpart cannot be predicted that accurately. However the maintainability gives insight about attributes that make the software maintenance comparatively easier. It focuses on factors like code modularity, reusability, readability etc. to achieve better maintainability.

Definition: Software maintainability

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

3) **Transportability**

Transportability or portability corresponds to the ease at which the software can be deployed across heterogeneous systems with minimum or no changes.

4) **Interoperability**

Interoperability gives the measure of ability of software to be able exchange information with external interfaces.

5) **Efficiency**

Efficiency indicates the quality of the software to produce maximum performance utilizing minimum hardware resources.

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. During the inception of the project the qualities to be paid importance should also be determined. Different projects might focus on different qualities with higher intent. The SQA procedures and monitoring should be accordingly tuned. Maximizing some qualities may cause others to be decreased (for example high efficiency might require code to be developed in assembly language which might reduce the portability of the code), hence the care should be taken to accommodate the project goals and proceed with SQA activities accordingly.

SELF-ASSESSMENT QUESTIONS - 2

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. _____________ gives the measure of ability of software to be able exchange information with external interfaces.

4. STANDARDS

For the evolution of software development methodology, establishing set of standards and procedures is very important. Standards set aside the deterministic points using which the software products can be compared with each other. Similarly, the procedures set aside the criteria using which software development and controlling processes can be compared. Hence the standards and procedures define what are basically guiding factors for software development; the SQA will ensure that the standards and procedures are present in the current project and are adequate for the needs of the project. Independent bodies have developed standards using which the process adherence, process monitoring, auditing can be done in unambiguous manner. We will discuss the ISO, CMM standards here.

<u>Benefits of using Standards:</u>

Standards provide an impartial and authoritative foundation for organisations and consumers all around the world to interact and conduct business because they give specific descriptions and terminology.

- software mistakes and bugs decreased
- Saving time and effort by reducing software errors
- Less money spent on creating and maintaining software
- Improved consistency in software operation
- Improved relationships with clients
- More content programmers

<u>Software Quality Standards:</u>

- ANSI: American National Standards Institute. Does not itself make standards but approves them
- EIA: Electronic Industries Association (Ex. EIA/IS¬632 Systems Engineering)
- IEEE: Institute of Electrical and Electronics Engineers Computer Society Software Engineering Standards Committee (Ex. IEEE Std 1228-1994 Standard for Software Safety Plans)

- ISO: International Organisation for Standardisation (Ex. ISO/IEC 2382-7:1989 Vocabulary)
- SEI : SEI stands for 'Software Engineering Institute'. CMM stands for 'Capability Maturity Model', developed by the SEI. It's a model of 5 levels of organizational 'Maturity' that determine effectiveness in delivering quality software

Certification of standards: To enable a software development company to demonstrate its capacity to consistently provide software products or maintenance services of a specified quality. The main goals of any certification programme are security, reliability, and the bare minimum of customer satisfaction.Organisations increase earnings and decrease losses when they adhere to quality standards.

4.1 ISO – Standards for software organization

ISO stands for International Organization for Standardization that controls and develops the standards for product development. ISO 9000 standards are for quality management systems. These standards are developed to help the organization to meet the customer requirements and satisfy the quality constraints. ISO 9000 standards are adopted by more than 100 countries worldwide. ISO 9001 provides the stringent requirements that the organizations will have to meet in order to be certified by third party through assessment.

It's important to remember that ISO 9000 certification demonstrates just that processes are followed as stated and does not guarantee product quality.

Outside of the software industry, many production and manufacturing firms can benefit from the guidelines outlined in the ISO 9001, 9002, and 9003 quality standards.

The software development industry typically uses 9001 since it is the most complete standard.

Everything from planning through last touches like documentation, production, testing, and servicing is included.

Professionals who are implementing the ISO 9000 standard provide comments to standing technical committees and advisory groups, who then make adjustments to the standard as needed.

ISO 9000-3 suggests that software development companies implement the quality management practises outlined in ISO 9001.

The ANSI/ASQ Q9000 series is the American equivalent of the worldwide ISO 9000 standards.

An independent auditor evaluates a company to determine whether or not it meets the criteria for ISO 9001 certification; the certification lasts for three years before it must be renewed.

ISO 900-3 Planning: Using ISO 9000-3 is now DEPRECATED. ISO 90003 has since replaced it.ISO drafted the quality standard ISO 9000-3:1997 to aid businesses in adapting the earlier standard ISO 9001:1994 for use with software.Organisations that create, deploy, and maintain software applications should implement ISO 9000-3.

The 1997 revision, known as ISO 9000-3, is an updated version of the 1994 standard ISO 9001.ISO has taken the content from ISO 9001, copied and pasted it into ISO 9000-3, and added a few lines of additional text that are specific to software. To be certified as ISO 9000-3 compliant, businesses must fulfil the following requirements:

- Build up your company's quality assurance system
- Set up the company's SQA system
- submit to audits for certification

To meet them, you'll need to carefully map out how you'll put in place the systems and gather the materials for the work that will lead to certification.

Obtaining ISO 9000-3 accreditation for software development and maintenance requires management to create a strategy for doing so.Initial steps involve doing an in-house audit of the present SQA setup and methodology. The certification procedure should launch from this in-house survey.The absence of necessary processes and the inadequacy of currently implemented ones as compared to the standards for SQA.

- Discrepancies between employees' current SQA expertise and what's needed to do their jobs effectively
- Discrepancies in the recording of construction and upkeep
- Dissimilarities in system capabilities and implementation of software configuration systems
- Discrepancies in the management procedures necessary for monitoring project development
- Disconnect between the SQA unit's structure and its abilities

Following the conclusion of the aforementioned evaluation, the strategy for acquiring certification can be developed. Such a list must have:

A schedule of events detailing what needs to be done and when. The time and money expected to be spent on each task and the Capabilities of the organisation:

(a) SQA employees and senior software engineers; (b) SQA professionals hired from outside the company.

It's important to get every last element right in the strategy you're making

The company should create a SQA management system according to ISO 9000-3 standards. The following are some things to consider adding: Creation of a thorough SQA process document and guide.

- Building out the rest of SQA's supporting structures, including educational opportunities for employees,
- Procedures for both preventing problems and fixing them.
- Services for managing configurations,
- Quality assurance documentation and record keeping.
- Creating a system to monitor the development of a project.

If there are any areas in which an organisation is not meeting requirements, it must do so before moving on to the next phase of certification.

The company's SQA processes and quality manual will be evaluated. The review establishes and guarantees fullness and accuracy. Audits designed to ensure that the company is adhering to the standards set out in its quality manual and SQA practises

4.2 CMM (Capability Maturity Model)

CMM (Capability Maturity Model) is a process improvement methodology developed by ***Software Engineering Institute (SEI)*** at Carnegie Mellon University, USA. CMM is developed for increasing the productivity in the software development. It provides set of best practices, **procedures, and standard practices** to address the important aspects of the software development. The CMM addresses areas like the productivity, cost control, performance, predictability etc. The goal of the CMM is to define the characteristics of mature and capable processes in a way that can be measured across the other organizations in the industry. Any organization will grow in terms of maturity from one level to other level as it adapts to the

SEI- CMM Principles: Using the model, a company can assess its progress and plan for the improvements necessary to advance to the subsequent capability level in the capability curve.

The following ideas and principles form the basis of CMM evaluation:

- The five-stage capability maturity model is the tool for better software development.
- The "what" but not the "how" are defined by process areas.
- Continuous improvement is emphasised.
- The model can be used by a wide variety of implementation groups thanks to the generic approach because:
- Any life cycle model can be used with it.
- Any programming language, software development tool, or design process can be used.
- It does not mandate a particular format for documentation.

SEI-CMM Evolution:

The framework was overhauled to accommodate numerous specialised maturity models for capabilities. In the years following its initial release in 1993, the SEI's Software Development and Maintenance Capability Maturity Model (SW-CMM) grew in scope and complexity. The following iterations have been created:

Focusing on system engineering procedures in relation to product-centric customer needs, the System Engineering Capability Maturity Model (SE-CMM)T-CMM was designed to help secure and confidential software systems that require higher levels of software quality assurance. The System Security Engineering Capability Maturity Model (SSE-CMM) is a framework for the development of safe products.

SEI CMM Levels:

CMM assess on five levels

- Optimized
- Managed
- Defined
- Repeatable
- Initial

Any organization will grow in terms of maturity from one level to other level as it adapts to the processes as defined in the standards of CMM. Figure 10.1 shows various levels in CMM.

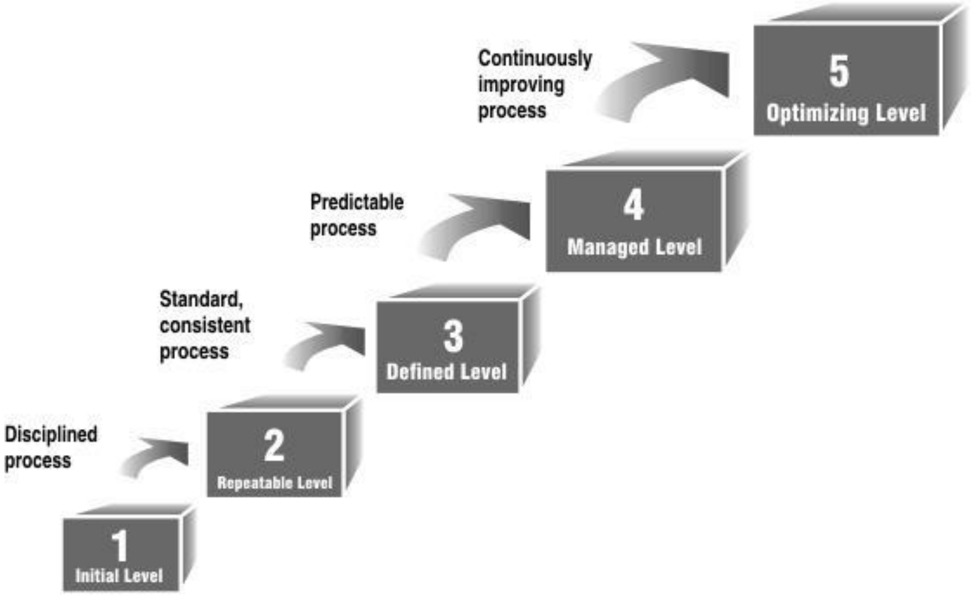

Fig. 10.1: The CMM model defines five levels of organizational maturity

1) **Initial level:** This is the first level of CMM which forms the basis for comparing with next higher levels of maturity. In this type of the organization, the result of the project depends on the resources and not on the practices. The success of the project depends on the personal approach taken by the project manager, his experience, and decisions. This means the project success is on the team which executed the project, and can be repeated only if the same set of resources is available. If the resources leave the company, work will be severely affected. Accordingly the development process will be limited to coding and testing.
2) **Repeatable level:** In this level basic project management technologies are applied in the company. The project planning is applied and the standards and quality concepts are introduced and team tries to use these standards for their project implementation. At critical times though, team tends to roll back to initial level concentrating only on the code and testing.
3) **Defined level:** At this next level, the main target is to decrease the dependency of the company on the resources in the organization. At this level processes for software development and maintenance are introduced to the organization and the same will be documented and followed in practice for the projects. The quality team is also set up and quality team helps in transitioning to the organization to adapt to these new

standards. Appropriate training will be provided by the quality department to help the organization to achieve this level. As the organization works on attaining this maturity level, the tendency of rolling back to previous level also decrease and the organization dependency on the resources is also avoided.

4) **Managed level:** At this level of maturity, quantitative analysis is applied for both software processes and the project performance as a whole for all the projects in the organization. Project management standards are improved based on the quantitative indices obtained across the organization. Also mature communication channel is set across the organization that leads to continuous process improvements based on the variations in indices and observations in the projects.
5) **Optimizing level:** At this highest level of maturity, the organization seeks continuous improvement for existing processes as well as evaluates the innovative approaches on newly introduced systems. As stated earlier organization seeks permanent improvement in its processes and tries to anticipate the possible errors and defects to achieve overall goal of decreasing the cost of software development.

Advantages of SEI –CMM:

- The number of certifications is unlimited.
- It makes progress and development possible.
- Used in addition to other measures of quality
- It immediately draws attention to any problems.
- The CMM assigns input priorities to tasks.
- A SWOT analysis is conducted.

4.3 Comparison between ISO 9001 & SEI-CMM

Both SEI-CMM and ISO 9001 targets to improve on software development practices by defining set of procedures and standards. Hence we can expect some correlation between the two standards. The level at which details are covered differ between the two standards. 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 between the two standards.

ISO defines the clauses while the CMM defines KPA's (Key process area) in its representation of the standards. 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 main deviation, between these two standards 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.

Similarly, the biggest similarity between CMM and ISO 9001 is the importance given to the documentation. 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 according to some experts the way ISO emphasizes on continuous improvement in the processes.

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 latest version of CMMI ver 1.2 was released in August 2006. There are 3 sub categories of CMMI in the new version. They are:

- CMMI Development
- CMMI Services
- CMMI Acquisition.

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 - 3

7. Standards set aside the deterministic points using which the software products can be compared with each other. (True / False)
8. SEI-CMM stands for ______________.
9. ANSI stands for ______________.

5. SUMMARY

Let's recapitulate important points:

- Software quality is an umbrella activity that is applied throughout the lifecycle of the project across all phases. It not only involves testing of the code that is developed, but also includes activities such as technical reviews, documentation, and compliance etc.
- 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.

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?

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. an approach that determines the technique need to be adopted for managing the quality in the project
   b. software engineering methodologies and tools
   c. technical reviews that need to be applied throughout the software processes
   d. testing strategy for the project
   e. documentation mechanism and effective ways of controlling the changes to the documents
   f. procedure for software compliance when needed to be obtained

g. metrics gathering and reporting mechanisms. (Refer Section 2 for detail)

2. Some commonly used software qualities are reliability, maintainability, transportability, interoperability, testability, usability, reusability, traceability, sustainability, and efficiency. (Refer Section 3)
3. ISO stands for International Organization for Standardization that controls and develops the standards for product development. ISO 9000 standards are for quality management systems. These standards are developed to help the organization to meet the customer requirements and satisfy the quality constraints. ISO 9000 standards are adopted by more than 100 countries worldwide. (Refer Sub-section 4.1)
4. CMM (Capability Maturity Model) is a process improvement methodology developed by Software Engineering Institute (SEI) at Carnegie Mellon University, USA. CMM is developed for increasing the productivity in the software development. It provides set of best practices, procedures, and standard practices to address the important aspects of the software development. The CMM addresses areas like the productivity, cost control, performance, predictability etc. The goal of the CMM is to define the characteristics of mature and capable processes in a way that can be measured across the other organizations in the industry. Any organization will grow in terms of maturity from one level to other level as it adapts to the processes as defined in the standards of CMM. (Refer Sub-section 4.2)
5. The main deviation, between these two standards 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. Similarly the biggest similarity between CMM and ISO 9001 is the importance given to the documentation. 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 4.3)

6. The other most popular standards apart from ISO and CMM are ANSI, CMMI and IEEE. (Refer Section 5)