

# 6

## Quality Assurance

### Syllabus

Software Quality Assurance Activities, Software Qualities, Software Quality Standards - ISO Standards for Software Organization, Capability Maturity Model (CMM), Comparison between ISO 9001 & SEI CMM, Other Standards.

### Contents

- 6.1 Software Quality Assurance
- 6.2 Software Quality Assurance Activities
- 6.3 Benefits of Software Quality Assurance (SQA)
- 6.4 Software Qualities
- 6.5 Software Quality Standards-ISO Standards for Software Organization
- 6.6 Capability Maturity Model (CMM)
- 6.7 Comparison between ISO 9000 and SEI-CMM
- 6.8 Other Standards



## 6.1 Software Quality Assurance

### Quality

- Quality is extremely hard to define, and it is simply stated: "Fit for use or purpose." It is all about meeting the needs and expectations of customers with respect to functionality, design, reliability, durability, & price of the product.

### Assurance

- Assurance is nothing but a positive declaration on a product or service, which gives confidence. It is certainty of a product or a service, which it will work well. It provides a guarantee that the product will work without any problems as per the expectations or requirements.

### Software Quality Assurance (SQA)

- Software Quality Assurance (SQA) is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standards. These defined standards could be one or a combination of any like ISO 9000, CMMI model, ISO15504, etc.
- SQA incorporates all software development processes starting from defining requirements to coding until release. Its prime goal is to ensure quality.

### Software Quality Assurance have :

1. A quality management approach
2. Formal technical reviews
3. Multi testing strategy
4. Effective software engineering technology
5. Measurement and reporting mechanism

## 6.2 Software Quality Assurance Activities

Major Software Quality Assurance Activities are :

### 1) Creating an SQA Management Plan

- The foremost activity includes laying down a proper plan regarding how the SQA will be carried out in your project. Along with what SQA approach you are going to follow, what engineering activities will be carried out, and it also includes ensuring that you have a right talent mix in your team.



## ❑ 2) Setting the Checkpoints

- The SQA team sets up different checkpoints according to which it evaluates the quality of the project activities at each checkpoint/project stage. This ensures regular quality inspection and working as per the schedule.

## ❑ 3) Apply Software Engineering Techniques

- Applying some software engineering techniques aids a software designer in achieving high-quality specification. For gathering information, a designer may use techniques such as interviews and FAST (Functional Analysis System Technique).
- Later, based on the information gathered, the software designer can prepare the project estimation using techniques like WBS (work breakdown structure), SLOC (source line of codes), and FP (functional point) estimation.

## ❑ 4) Executing Formal Technical Reviews

- An FTR is done to evaluate the quality and design of the prototype.
- In this process, a meeting is conducted with the technical staff to discuss regarding the actual quality requirements of the software and the design quality of the prototype. This activity helps in detecting errors in the early phase of SDLC and reduces rework effort in the later phases.

## ❑ 5) Having a Multi - Testing Strategy

- By multi-testing strategy, we mean that one should not rely on any single testing approach, instead, multiple types of testing should be performed so that the software product can be tested well from all angles to ensure better quality.

## ❑ 6) Enforcing Process Adherence

- This activity insists the need for process adherence during the software development process. The development process should also stick to the defined procedures.
- **This activity is a blend of two sub-activities which are explained below in detail :**
  - (i) **Product Evaluation :** This activity confirms that the software product is meeting the requirements that were discovered in the project management plan. It ensures that the set standards for the project are followed correctly.
  - (ii) **Process Monitoring :** This activity verifies if the correct steps were taken during software development. This is done by matching the actually taken steps against the documented steps.



**□ 7) Controlling Change**

- Use a mix of manual procedures and automated tools to have a mechanism for change control.
- By validating the change requests, evaluating the nature of change and controlling the change effect, it is ensured that the software quality is maintained during the development and maintenance phases.

**□ 8) Measure Change Impact**

- If any defect is reported by the QA team, then the concerned team fixes the defect.
- After this, the QA team should determine the impact of the change which is brought by this defect fix. They need to test not only if the change has fixed the defect, but also if the change is compatible with the whole project.
- For this purpose, we use software quality metrics which allows managers and developers to observe the activities and proposed changes from the beginning till the end of SDLC and initiate corrective action wherever required.

**□ 9) Performing SQA Audits**

- It Inspects the entire actual SDLC process followed by comparing it against the established process.
- It also checks whatever reported by the team in the status reports were actually performed or not. This activity also exposes any non-compliance issues.

**□ 10) Maintaining Records and Reports**

- It is crucial to keep the necessary documentation related to SQA and share the required SQA information with the stakeholders. The test results, audit results, review reports, change requests documentation, etc. should be kept for future reference.

**□ 11) Manage Good Relations**

- In fact, it is very important to maintain harmony between the QA and the development team.
- We often hear that testers and developers often feel superior to each other. This should be avoided as it can affect the overall project quality.



### 6.3 Benefits of Software Quality Assurance (SQA)

1. SQA produce high quality software.
2. High quality application saves time and cost.
3. SQA is beneficial for better reliability.
4. SQA is beneficial in the condition of no maintenance for long time.
5. High quality commercial software increase market share of company.
6. Improving the process of creating software.
7. Improves the quality of the software.

### 6.4 Software Qualities

- Software Quality is defined as the conformance to explicitly state functional and performance requirements, explicitly documented development standards, and inherent characteristics that are expected of all professionally developed software.

#### Software quality attributes

- Each attribute can be used to measure product performance. These attributes can be used for Quality assurance as well as Quality control.

1. **Functionality** - A set of attributes that bear on the existence of a set of functions and their specified properties. The functions satisfy stated or implied needs.

- Suitability
- Accuracy
- Interoperability
- Security
- Functionality Compliance

2. **Reliability** - A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.

- Maturity
- Fault Tolerance
- Recoverability
- Reliability Compliance



3. **Usability** - A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.
  - Understandability
  - Learnability
  - Operability
  - Attractiveness
  - Usability Compliance
4. **Efficiency** - A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used under stated conditions.
  - Time Behavior
  - Resource Utilization
  - Efficiency Compliance
5. **Maintainability** - A set of attributes that bear on the effort needed to make specified modifications.
  - Analyzability
  - Changeability
  - Stability
  - Testability
  - Maintainability Compliance
6. **Portability** - A set of attributes that bear on the ability of software to be transferred from one environment to another.
  - Adaptability
  - Installability
  - Coexistence
  - Replaceability
  - Portability Compliance
- The quality attributes in the areas of *functionality*, *reliability*, *usability*, *efficiency*, and *portability* address system properties that can be implemented like normal functional requirements.

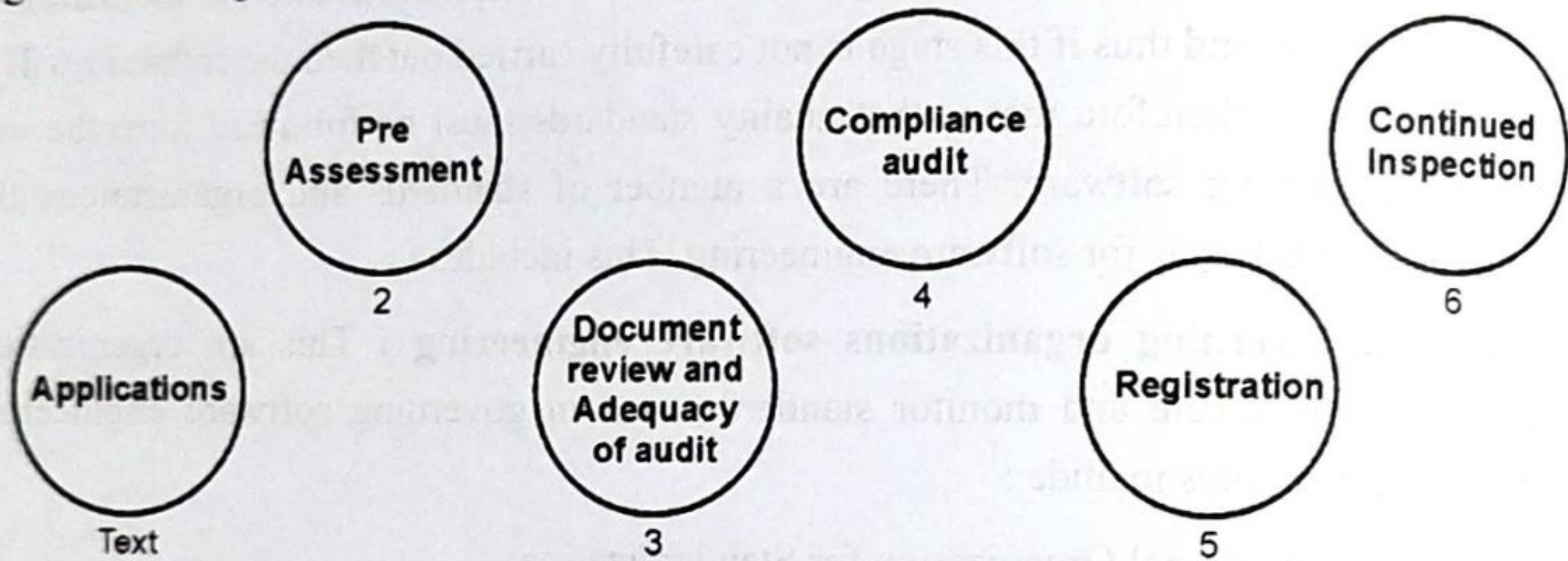


## 6.5 Software Quality Standards - ISO Standards for Software Organization

- Software engineering is a vast field, and the products created by software engineers must be of upmost quality. For this quality to be achieved, correct measures and grading standards must be followed, so as to achieve excellent products. There are many different quality standards put in place.
- This is the fact that the requirement stage in software development create the foundation of measuring quality, and thus if this stage is not carefully carried out then the software will be short of quality. It therefore means that quality standards must be followed from the very first step of creating software. There are a number of standards and organizations that govern quality standards for software engineering. This includes :
  - **Standard governing organizations software engineering** : This are organizations established to create and monitor standards used in governing software engineering. Some organizations include :
    1. **ISO** : International Organization for Standardization
    2. **SEI** : Software Engineering Institute (Carnegie-Mellon University)
    3. **IEEE** : Institute of Electrical and Electronics Engineers Computer Society Software Engineering Standards
    4. **ANSI** : American National Standards Institute
    5. **EIA** : Electronic Industries Association
  - **Quality standards for software engineering** : These are standard written down, that must be followed by software engineers and engineering firms to achieve quality in their products. Below are some major standards in the software engineering industry :
    1. Capability Maturity Model (CMM)
    2. ISO 9000 family : This includes the :
      - ISO 9000
      - ISO 9001
      - ISO 9126
    3. IEEE STD 1061 - 1998 Standard
    4. FAA-STD-018 and - 018a
    5. ANSI/IEEE std 730 - 1998



- The **International Standards Organization (ISO)** is a standard which serves as a for contract between independent parties. It specifies guidelines for development of **quality system**.
- Quality system of an organization means the various activities related to its products or services. Standard of ISO addresses to both aspects i.e. operational and organizational aspects which includes responsibilities, reporting etc. An ISO 9000 standard contains set of guidelines of production process without considering product itself.



**Fig. 6.5.1 : ISO 9000 Certification**

#### • Why ISO Certification required by Software Industry ?

There are several reasons why software industry must get an ISO certification. Some of reasons are as follows :

- This certification has become a standards for international bidding.
- It helps in designing high-quality repeatable software products.
- It emphasis need for proper documentation.
- It facilitates development of optimal processes and totally quality measurements.

#### □ Features of ISO 9000 Requirements

- **Document control** - All documents concerned with the development of a software product should be properly managed and controlled.
- **Planning** - Proper plans should be prepared and monitored.
- **Review** - For effectiveness and correctness all important documents across all phases should be independently checked and reviewed .
- **Testing** - The product should be tested against specification.
- **Organizational aspects** - Various organizational aspects should be addressed e.g., management reporting of the quality team.



### ❑ Advantages of ISO 9000 Certification

Some of the advantages of the ISO 9000 certification process are following :

- Business ISO-9000 certification forces a corporation to specialize in "how they are doing business". Each procedure and work instruction must be documented and thus becomes a springboard for continuous improvement.
- Employees morale is increased as they're asked to require control of their processes and document their work processes.
- Better products and services result from continuous improvement process.
- Increased employee participation, involvement, awareness and systematic employee training are reduced problems.

### ❑ Shortcomings of ISO 9000 Certification

Some of the shortcoming of the ISO 9000 certification process are following :

- ISO 9000 does not give any guideline for defining an appropriate process and does not give guarantee for high quality process.
- ISO 9000 certification process have no international accreditation agency exists.

## 6.6 Capability Maturity Model (CMM)

- It has long been accepted that continuous process improvement is based on many small evolutionary steps rather than larger revolutionary innovations. The Capability Maturity Model (CMM) provides a framework for organising these evolutionary steps into five maturity levels that lay successive foundations for continuous process improvement.
- This methodology is at the heart of most management systems which are designed to improve the quality of the development and delivery of all products and services.

### ❑ The Five Maturity Levels

- The five maturity levels define a scale for measuring the maturity of an organisation's software process and for evaluating the capability of these processes. They also help an organisation prioritise its improvement efforts.
- *A maturity level is a well-defined evolutionary plateau toward achieving a mature software process.*
- Each maturity level comprises a set of process goals that, when satisfied, stabilise an important component of the process. Achieving each level of maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organisation.



• The five Software Capability Maturity levels have been defined as :

1. **Initial** : The software process is characterised as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.
2. **Repeatable** : Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
3. **Defined** : The software process for both management and engineering activities is documented, standardised, and integrated into all processes for the organisation. All projects use an approved version of the organisation's standard software process for developing and maintaining software.
4. **Managed** : Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
5. **Optimising** : Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

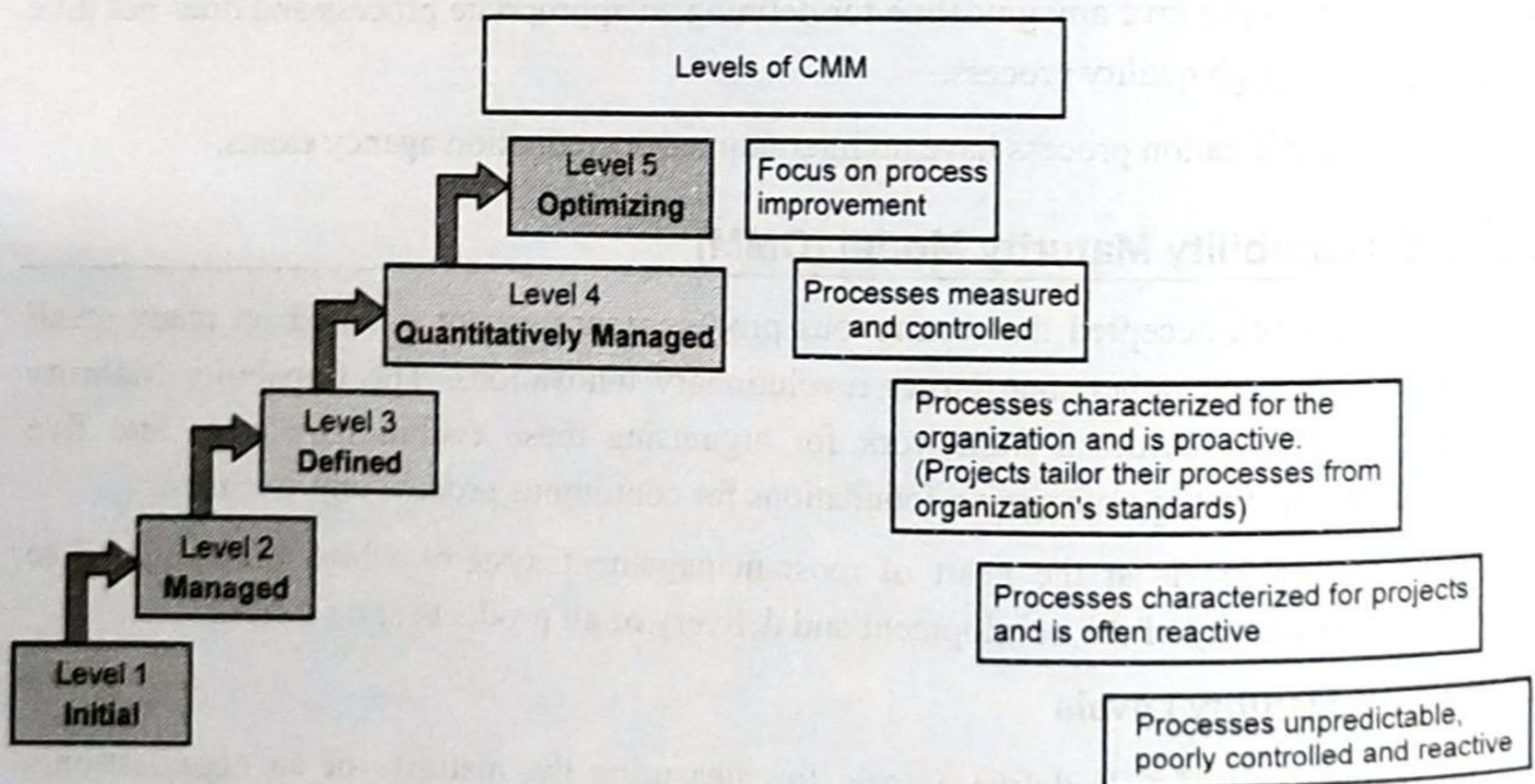


Fig. 6.6.1

## 6.7 Comparison between ISO 9000 and SEI - CMM

### ISO 9000

- ISO9000 is an international standard of quality management and quality assurance. It certifies the companies that they are documenting the quality system elements which are needed to run a efficient and quality system.



## □ SEI-CMM

- SEI (Software Engineering Institute) - Capability Maturity Model (CMM) is specifically for software organizations to certify them at which level, they are following and maintaining the quality standards.

- Following are the important differences between ISO9000 and SEI-CMM.

Sr. No.	Key	ISO9000	SEI-CMM.
1	Definition	ISO9000 is an international standard of quality management and quality assurance. It certifies the companies that they are documenting the quality system elements which are needed to run a efficient and quality system.	SEI-CMM is specifically for software organizations to certify them at which level, they are following and maintaining the quality standards.
2	Focus	Focus of ISO9000 is on customer supplier relationship, and to reduce the customer's risk.	Focus of SEI-CMM is to improve the processes to deliver a quality software product to the customer.
3	Target Industry	ISO9000 is used by manufacturing industries.	SEI-CMM is used by software industry.
4	Recognition	ISO9000 is univesally accepted accross lots of countries.	SEI-CMM is mostly used in USA.
5	Guidelines	ISO9000 guides about concepts, priciples and safeguards to be in place in a workplace.	SEI-CMM specifies what is to be followed at what level of maturity.
6	Levels	ISO9000 has one acceptance level.	SEI-CMM has five acceptance levels.
7	Validity	ISO9000 certificate is valid for three years.	SEI-CMM certificate is valid for three years as well.
8	Level	ISO9000 has no levels.	SEI-CMM has five levels, Initial, Repeatable, Defined, Managed and Optimized.
9	Focus	ISO9000 focuses on following a set of standards so that firms delivery are successful everytime.	SEI-CMM focuses on improving the processes.



## 6.8 Other Standards

- **Quality standards for software engineering :** These are standard written down, that must be followed by software engineers and engineering firms to achieve quality in their products. Below are some major standards in the software engineering industry :
  1. **Capability Maturity Model (CMM) :** Written by the software engineering institute, this model describes good practices for planning, engineering and managing software development. This practice allows software engineering organizations to create quality software while minimizing budget and time factors. CMM allows software engineers to judge and compare its processes to stated practices in the industry. Thus improving software processes and quality.
  2. **ISO 9000 family :** These are standards for quality management and quality assurance for any business. It applies to process of creating, and controlling of products and services an organization provides, by giving an organized course for activities to be carried out to ensure that customer's requirements are made (ISOQAR, 2010). This includes the :
    - ISO 9000
    - ISO 9001
    - ISO 9126
  3. **IEEE STD 1061-1998 standard :** This standard was created as a way to establish quality requirements, and provides a well defined metrics for identifying, implementing, analyzing and validating the processes of developing a quality software product. The standard provides metrics for quality for the entire software engineering life cycle (IEEE Std 1061, 1998).
  4. **FAA-STD-018 and -018a :** This are software quality standards created by the Federal Aviation Authority (FAA-STD-018a, 1987).
  5. **ANSI/IEEE std 730 - 1998 :** This standard was created to provide organization a framework to prepare content for a software quality assurance plan, by providing the most minimum and uniform acceptable requirements possible (ANSI/IEEE std 730,1998).
- **Quality is the key to any successful software project,** thus software engineers and engineering organizations must work toward achieving it. It is therefore necessary to have quality standards in place that help achieve this goal. These standards do not govern the product but rather govern the process of creating the product to achieve its maximum quality.

□□□