# Unit-V

## Q.Explain the concept of software quality.

## Software Quality Concept

There is no one universal definition of software quality. This is because of the complexity caused by the three or more participants affected by the quality of software, namely, customer, developer and stakeholders. The issue is whose views, expectations and aspirations are to be considered supreme. The majority hold that customer satisfaction should be the goal for measuring software quality. The customer may be satisfied though with software, the quality of which cannot be considered the best by other standards.

**The software quality definition is based on the following:**

- Customer focus and customer satisfaction
- Functional and performance requirement
- Ease of learning, use and maintainability
- Adherence to development standards

Customer satisfaction largely depends on meeting functional and performance requirements and ease of operations. Adhering to development standards ensures to a great extent the achievement of these goals.

Software quality is defined as the quality that ensures customer satisfaction by offering all the customer deliverables on performance, standards and ease of operations. The definition is applicable for software as well as for a generic software product.

**Q.Explain the role of SQA group.**
**Q.What are the objectives of software quality assurance group.**

## Software Quality Assurance Group (SQA)

Software quality assurance is a planned effort to ensure that a software product fulfills these criteria and has additional attributes specific to the project, e.g., portability, efficiency, reusability, and flexibility. It is the collection of activities and functions used to monitor and control a software project so that specific objectives are achieved with the desired level of confidence. It is not the sole responsibility of the software quality assurance group but is determined by the consensus of the project manager, project leader, project personnel, and users.

A formal definition of software quality assurance is that is ‘the systematic activities providing evidence of the fitness for use of the total software product.” Software quality assurance is achieved through the use of established guidelines for quality control to ensure the Integrity and prolonged life of software. The relationships between quality assurance, quality control, the auditing function, and software testing are often confused.

Quality assurance is the set of support activities needed to provide adequate confidence that processes are established and continuously improved in order to products that meet specifications and are fit for use. Quality control is the process by which product quality is compared with applicable standards and the action taken when nonconformance is detected. Auditing is the inspection/ assessment activity that verifies compliance with plans, policies, and procedures.

## Q.Explain the various quality control activities.

## <u>SQA Activities</u>

SQA is ensured through a Quality Management System (QMS), QMS is made of several components; it is a system integrated in the bigger system of software development, which comprises project, process and product management systems.
The Software Engineering Institute (SEI) recommends a set of activities, which, when implemented effectively, assures the designed quality. These activities include:

- Quality assurance planning
- Data gathering on key quality defining parameters
- Data analysis and reporting
- Quality control mechanisms

**Components of Quality Assurance**
Most software quality assurance activities can be categorized into software testing, i.e.,
1] verification and validation,
2] software configuration management, and
3] quality control.

**Q.Explain software quality assurance standard in detail.**

**Q.Explain and compare process and product quality in detail.**

**Quality management**:

**Quality Management** is a much broader field that ensures the required level of quality is achieved in software product. You can create a standard quality management approach for your organization. It has four main sub processes: **1] quality assurance, 2] quality planning, 3] quality control and 4] quality improvement.**

**1] Quality Assurance Standards**

Software Quality Assurance, QA, is a planned and systematic way of creating an environment to assure that the software product being developed meets the quality requirements.

**QA refers to the implementation of well-defined standard practices and methods.** It is a pro-active quality process. This process is controlled and determined at managerial level**. Quality assurance focuses on the process checklists, process standards, project audits, methodology and procedures for development.**

It is a preventive process that aims at establishing the correct methodology and standards to provide a quality environment to the product being developed.

**2] Quality planning**

In this sub process, quality assurance plan is created for a particular project. In the quality assurance plan, organizational standards are selected which are applicable to a project. It should also involve the **plan for quality control.**

**Quality assurance planning details out what QC activities are performed, when the QC activities are performed and who will perform those.** It also contains details of resource required, tools and techniques to be used for performing quality control.

**3] Quality Control**

Quality Control, QC, is the set of activities **that control the quality of product being developed by identifying any bugs that might be present.**

**Quality control process is a subset that falls under the quality assurance.** It is a corrective process. The task of actual testing is performed to find out and identify the bugs present in the product. The bugs are raised to the developers, who then try to fix them.

After fixes, the product is verified again such that the functionalities and features are working as required**. QC process assures that that the product being developed is of the required quality. Examples of quality control activities include inspection, deliverable peer reviews and the software testing process.**

### 4] Quality Improvement

Quality improvement is a formal approach to analyze the feedback received from the quality control team. In this process efforts are put systematically to identify any room of improvements in the existing standards and procedures. The target is to improve the process that establishes the standards of quality in the organization.

**Note: -** “The primary difference between quality assurance and quality control is that the quality assurance activities are conducted during the software development. Quality control activities are performed after the software has been developed.”

### Product Quality

**Product quality means to incorporate features that have a capacity to meet consumer needs (wants) and gives customer satisfaction by improving products (goods) and making them free from any deficiencies or defects.** There are various important aspects which define a product quality like:

1. Storage
2. Quality of Design
3. Quality of Conformance
4. Reliability
5. Safety

### Process Quality

**Process Quality is defined as all the steps used in the manufacturing the final product.** Its focus is on all activities and steps used to achieve a maximum acceptance regardless of its final product.

So, there is a big difference between product quality and process quality i.e. even when you follow the process with all quality features it may or may not produce a useful product, on the other hand product quality comprises of all the features required to make it acceptable to your customer and hence solve all the pain points.

-----------------------------------------------------------------------------------------------

## 'Software Maintenance'

Software maintenance is a part of Software Development Life Cycle. Its main purpose is to modify and update software application after delivery to correct faults and to improve performance. Software is a model of the real world.
The **Maintenance** Phase occurs once the system is operational. It includes implementation of changes that **software** might undergo over a period of time, or implementation of new requirements after the **software** is deployed at the customer location.

**Why Software Maintenance Is Necessary**?
**Maintaining** a system is equally **important** as Web Application Development. It keeps solutions healthy to deal with changing technical and business environment. ... It introduces technical advancements almost every day that improve solution efficiency to streamline business operations.

**Types of Software Maintenance**
There are four types of maintenance, namely, 1] corrective, 2] adaptive, 3] perfective, and 4] preventive
Corrective maintenance is concerned with fixing errors that are observed when the software is in use.
Adaptive maintenance is concerned with the change in the software that takes place to make the software adaptable to new environment such as to run the software on a new operating system.
Perfective maintenance is concerned with the change in the software that occurs while adding new functionalities in the software.
Preventive maintenance involves implementing changes to prevent the occurrence of errors. The distribution of types of maintenance by type and by percentage of time consumed.

**1] Corrective maintenance** deals with the repair of faults or defects found in day-today system functions. A defect can result due to errors in software design, logic and coding. Design errors occur when changes made to the software are incorrect, incomplete, wrongly communicated, or the change request is misunderstood. Logical errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow, or incomplete test of data. All these errors, referred to as residual errors, prevent the software from conforming to its agreed specifications. Note that the need for corrective maintenance is usually initiated by bug reports drawn by the users.

**2] Adaptive Maintenance**

Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns, and government policies have a significant impact on the software system.

For instance, a government policy to use a single 'European currency' will have a significant effect on the software system. An acceptance of this change will require banks in various member countries to make significant changes in their software systems to accommodate this currency. Adaptive maintenance accounts for 25% of all the maintenance activities.

**3] Perfective Maintenance**

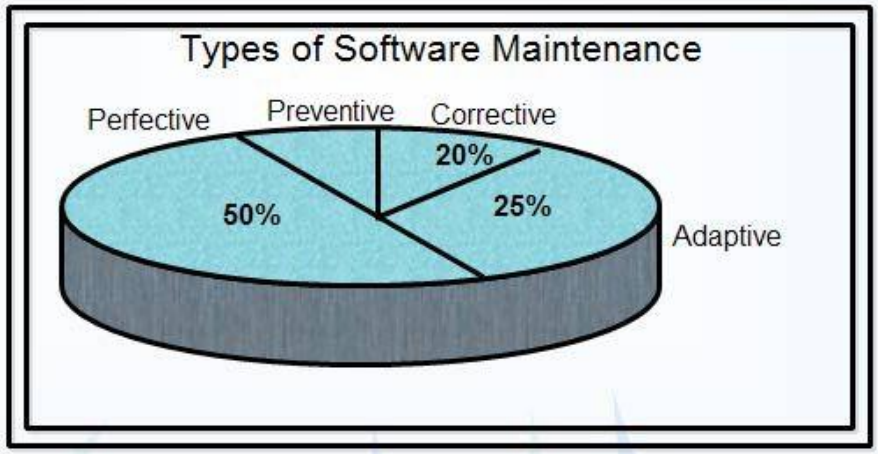
Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

Examples of perfective maintenance include modifying the payroll program to incorporate a new union settlement and adding a new report in the sales analysis system. Perfective maintenance accounts for 50%, that is, the largest of all the maintenance activities.

**4] Preventive Maintenance**

Preventive maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understandability and increasing software maintainability. It comprises documentation updating, code optimization, and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

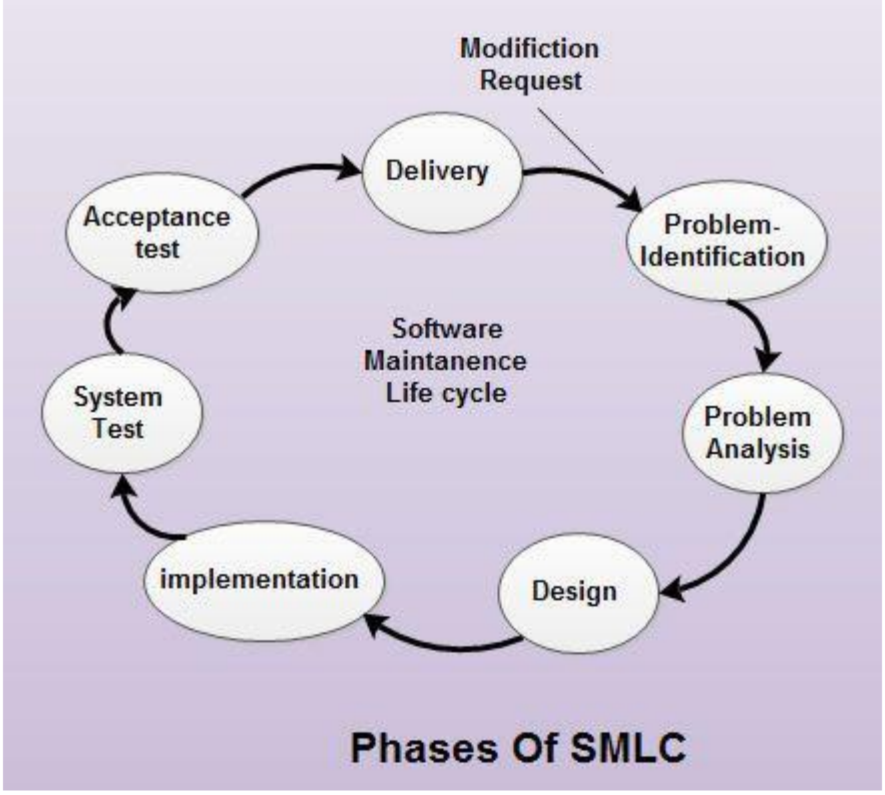
Preventive maintenance is limited to the maintenance organization only and no external requests are acquired for this type of maintenance. Preventive maintenance accounts for only 5% of all the maintenance activities.



## Q.Explain software maintenance life cycle in detail.

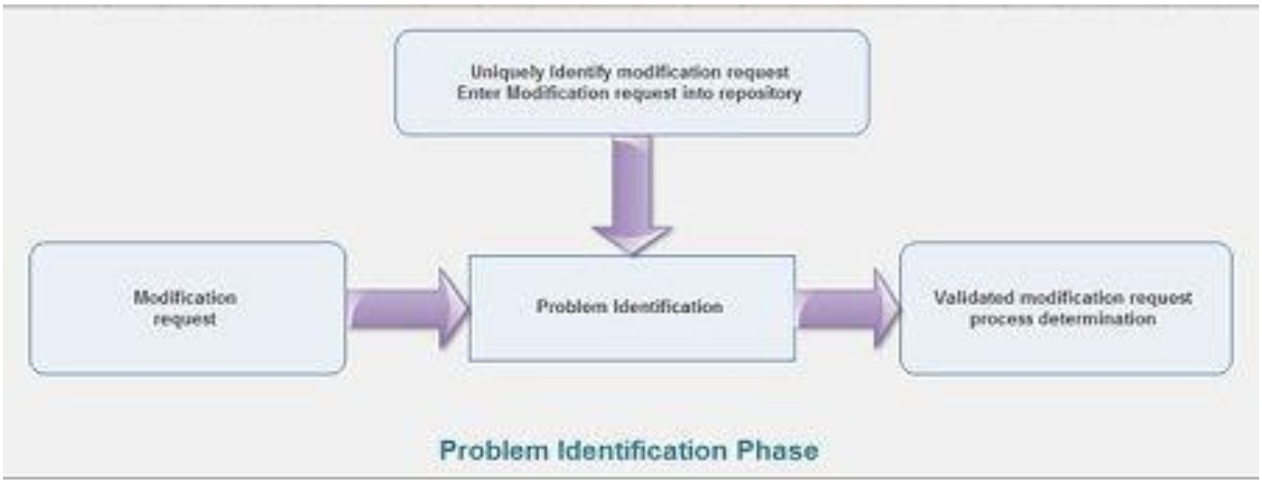
## Software Maintenance Life Cycle

Changes are implemented in the software system by following a software maintenance process, which is known as Software Maintenance Life Cycle (SMLC).This life cycle comprises seven phases, namely, problem identification, analysis, design, implementation, system testing, acceptance testing, and delivery phase.



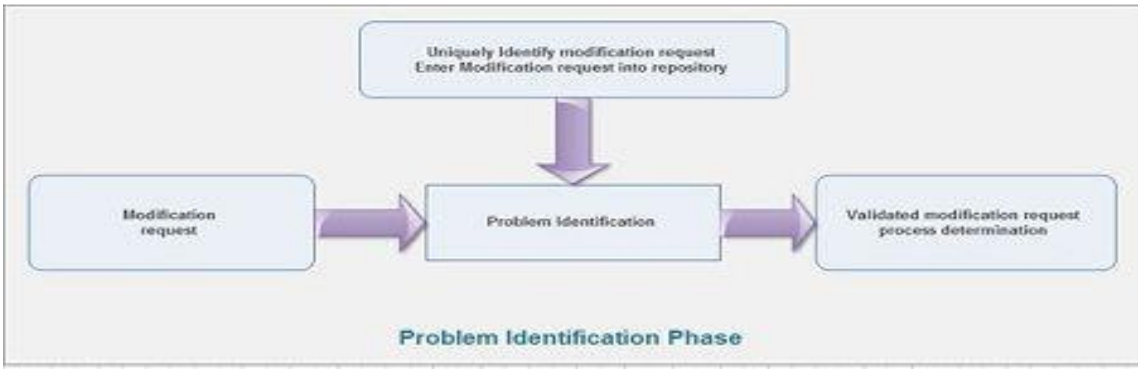
### 1] Problem Identification Phase

In this phase, the requests for modifications in the software are identified and assigned an identification number. Each Modification Request (MR) is then assessed to determine to which type of maintenance activity (corrective, adaptive, perfective, and preventive) the MR belongs. After classification, each MR is assigned with a priority to determine the order in which it is to be processed.


Uniquely Identify modification request
Enter Modification request into repository
Modification request
Problem Identification
Validated modification request process determination
Problem Identification Phase

**2] Problem Analysis Phase**

In this phase, the feasibility and scope of each validated modification request are determined and a plan is prepared to incorporate the changes in the software. The input attribute comprises validated modification request, initial estimate of resources, project documentation, and repository information.


Uniquely Identify modification request
Enter Modification request into repository
Modification request
Problem Identification
Validated modification request process determination
Problem Identification Phase

**3] Design Phase**

In this phase, the modifications to be made in the software are designed. The input attribute comprises outputs produced by analysis phase (detailed analysis), project and system documentation, software's source code, and databases.

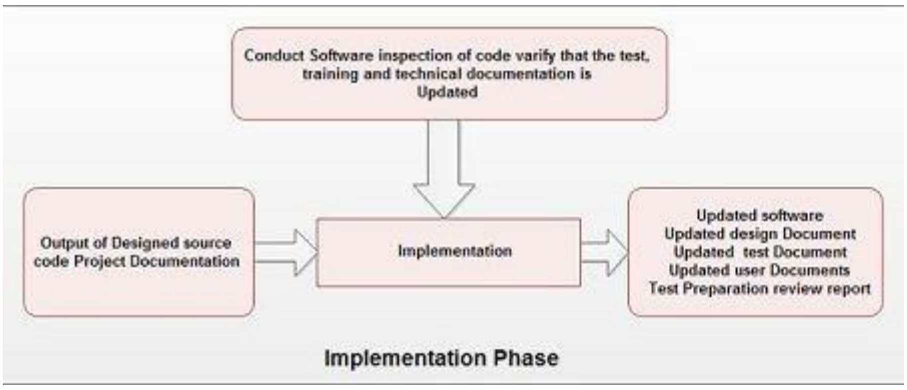
The process attribute for design comprises the following steps.

1. Identifying the affected software modules
2. Modifying software module documentation (like data-flow diagrams and program design language)
3. Developing test cases for the new design including safety and security issues.
4. Creating regression tests

5. Documenting the updated requirements
6. Revising the list of modifications.

**4] Implementation Phase**

In this phase, the actual modifications in the software code are made, new features that support the specification of present software are added, and the modified software is installed. The input attribute comprises the source code, the output of design phase, and the modified system and project documentation.


Conduct Software inspection of code varify that the test, training and technical documentation is Updated
Output of Designed source code Project Documentation
Implementation
Updated software
Updated design Document
Updated test Document
Updated user Documents
Test Preparation review report
Implementation Phase

The process attribute comprises the following subprocesses.

1. **Coding and unit testing:** The user requirements as identified in the design phase are implemented in the form of software code. After the software code is written, unit testing on individual modules is performed.

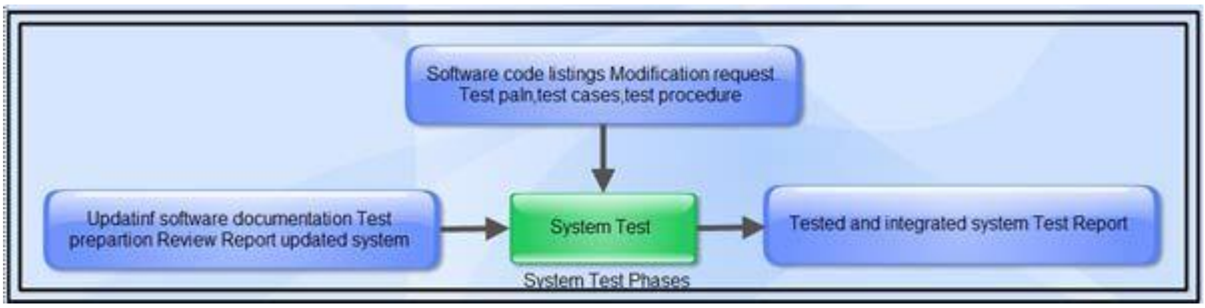
2. **Integration:** When the software code is written and all the modules are tested, the modified software modules are integrated with the existing system. After integration of the software system, integration test and regression test are performed. These tests are conducted to identify the impact on the functionality, performance, usability, and security of the modified software system. Impacts that lead to serious errors such as failure of software system are recorded so that they can be removed after identifying and understanding the source of error.

3. **Risk analysis and review:** In the implementation phase, the risk analysis and review are carried out periodically rather than on the completion of this phase. Risks that commonly occur are risk for data failure, provision of backup, and so on.

4. **Test preparation review:** This review is conducted to evaluate the review for system test in accordance with IEEE standards.

### 5] System Test Phase

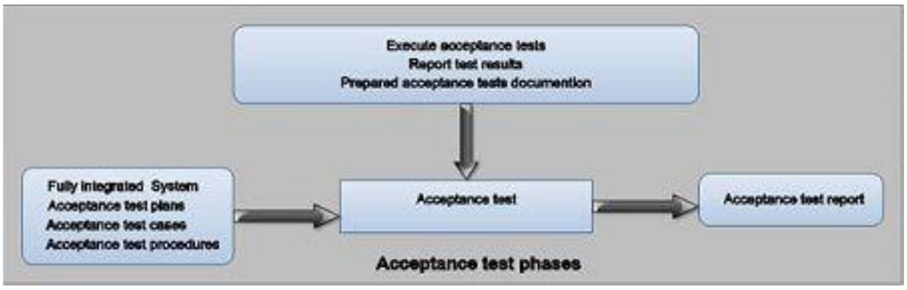
In this phase, the regression testing (a type of system testing) is performed on the modified system to ensure that no new faults are introduced in the software as a result of the maintenance activity. The input attribute comprises the updated software documentation, test preparation review report, and the updated system.


Software code listings Modification request
Test paln,test cases,test procedure
Updatinf software documentation Test preparion Review Report updated system
System Test
Tested and integrated system Test Report

System Test Phases

### 6] Acceptance Test Phase

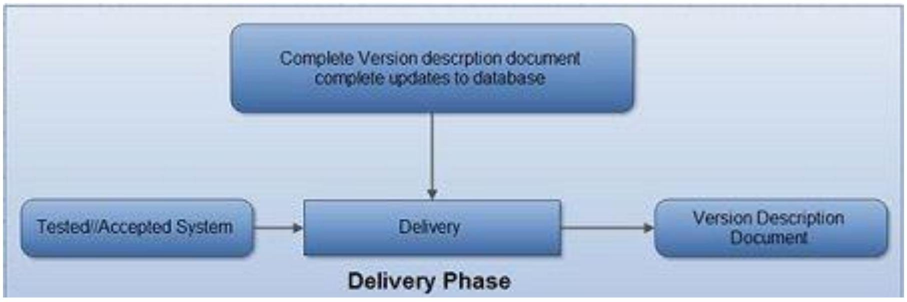
In this phase, acceptance testing is performed on the fully integrated system by the user or by a third party specified by the user. The objective is to detect errors and verify that the software features are according to the requirements stated in the modification request. The input attribute comprises the fully integrated system, acceptance test plans, acceptance test cases, and acceptance test procedures.



Acceptance test phases

### 7] Delivery Phase

In this phase, the modified (or new) software system is delivered to the user. In addition, users are provided with a proper documentation consisting of manuals and help files that describe the operation of the software along with its hardware specifications. The input attribute comprises a fully tested and accepted version of the system.


Complete Version descrption document complete updates to database
Tested//Accepted System
Delivery
Version Description Document
Delivery Phase