

# SOFTWARE PROJECT MANAGEMENT

## UNIT-I

### 2 Mark

#### **1. Define software project management.**

Project management software is software used for project planning, scheduling, resource allocation and change management. It allows project managers (PMs), stakeholders and users to control costs and manage budgeting, quality management and documentation and also may be used as an administration system.

Project management software is also used for collaboration and communication between project stakeholders.

#### **2. Define software development process.**

The software development process is a general term describing the over-arching process of developing a software product.

Development processes include the following activities:

- Requirement gathering
- Design
- Implementation
- Testing
- Maintenance

#### **3. What is meant by software model?**

The **software development models** are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals.

There are many development life cycle models that have been developed in order to achieve different required objectives.

The models specify the various stages of the process and the order in which they are carried out.

#### 4. Define Project Activities.

**Project activities planning and scheduling** is the first process group of project time management.

Developing the project implementation schedule is the second group.

#### 5. Define project activity tailoring.

**Project Tailoring** means adapting requirements and specifications of a project to current operational needs of an organization through reviewing, modifying and supplementing the project data.

#### 6. List out management tools.

- Gantt chart
- PERT chart
- Critical path analysis

### 5 mark

#### 1. Explain software development process

The process of software development services in India goes through a series of stages in step wise fashion that almost every developing company follows.

Known as the 'software development life cycle,' these six steps include planning, analysis, design, development & implementation, testing & deployment and maintenance.



- **Planning:** Without the perfect plan, calculating the strengths and weaknesses of the project, development of software is meaningless.
- **Analysis:** This step is about analysing the performance of the software at various stages and making notes on additional requirements. Analysis is very important to proceed further to the next step.

- **Design:** Once the analysis is complete, the step of designing takes over, which is basically building the architecture of the project.
- **Development & Implementation:** The actual task of developing the software starts here with data recording going on in the background.
- **Testing:** The testing stage assesses the software for errors and documents bugs if there are any.
- **Maintenance:** Once the software passes through all the stages without any issues, it is to undergo a maintenance process wherein it will be maintained and upgraded from time to time to adapt to changes.

## 2. Explain Risk Management process.

There are following activities involved in risk management process:

- **Identification** - Make note of all possible risks, which may occur in the project.
- **Categorize** - Categorize known risks into high, medium and low risk intensity as per their possible impact on the project.
- **Manage** - Analyse the probability of occurrence of risks at various phases. Make plan to avoid or face risks. Attempt to minimize their side-effects.
- **Monitor** - Closely monitor the potential risks and their early symptoms. Also monitor the effects of steps taken to mitigate or avoid them.

## 3. Explain Activity tailoring.

**Project Tailoring** means adapting requirements and specifications of a project to current operational needs of an organization through reviewing, modifying and supplementing the project data.

It is a consistent process to ensure that the project is performed correctly according to the organization's operational and business requirements. The process is carried out when the project is finished and there is a need to tailor or fit its product to the business needs.

Tailoring a project is intended to help a management team to ensure that planning, control, governance and use of the project and its deliverables are done in the right way so that the team can continue work towards project deployment and maintenance.

The team needs to be sure that the project approach, strategies, controls, processes, roles, jobs and other content are aligned with organizational goals.

There are four criteria for evaluating a project and tailoring it to operational and business needs. These criteria are:

- An amount of internal effort involved in executing project work, including technical work, planning, implementation.
- An amount of external effort involved in delivering project results, including investments, capital expenditures, third-party consultation costs.
- The total project duration measured in units of time (weeks, months, years).
- Business value delivered by the project.

### **10 mark**

#### **1. Briefly explain the concept of Management tools.**

The risk and uncertainty rises multifold with respect to the size of the project, even when the project is developed according to set methodologies.

There are tools available, which aid for effective project management. A few are described -

- **Gantt Chart**

Gantt charts was devised by Henry Gantt (1917). It represents project schedule with respect to time periods.

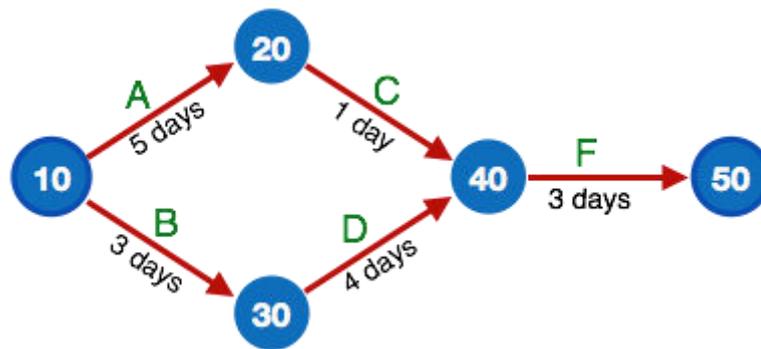
It is a horizontal bar chart with bars representing activities and time scheduled for the project activities.

Weeks	1	2	3	4	5	6	7	8	9	10
<b>Project Activities</b>										
<b>Planning</b>	■	■	■							
<b>Design</b>			■	■	■	■				
<b>Coding</b>						■	■	■	■	
<b>Testing</b>							■	■	■	
<b>Delivery</b>										■

- **PERT Chart**

PERT (Program Evaluation & Review Technique) chart is a tool that depicts project as network diagram.

It is capable of graphically representing main events of project in both parallel and consecutive way. Events, which occur one after another, show dependency of the later event over the previous one.



Events are shown as numbered nodes. They are connected by labeled arrows depicting sequence of tasks in the project.

- **Resource Histogram**

This is a graphical tool that contains bar or chart representing number of resources (usually skilled staff) required over time for a project event (or phase).

Resource Histogram is an effective tool for staff planning and coordination.

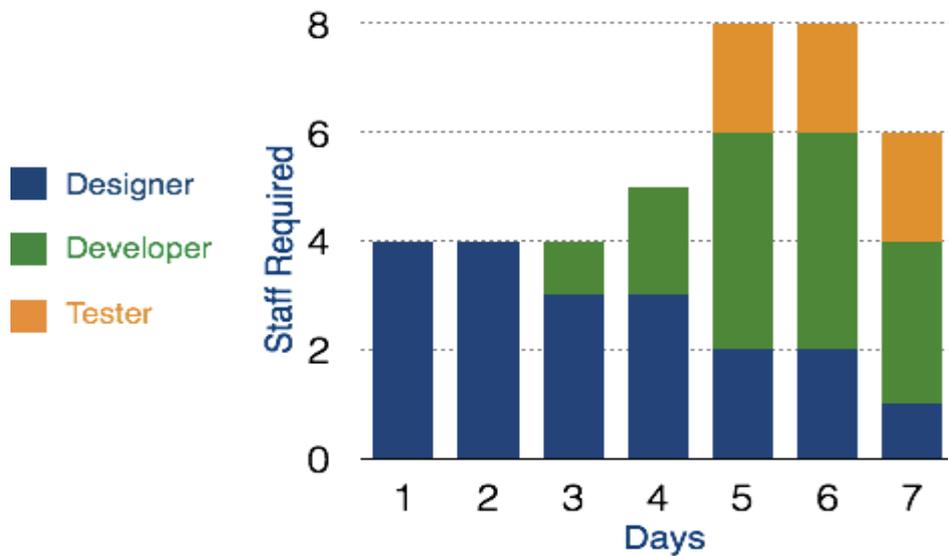
Staff	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
<b>Designer</b>	4	4	3	3	2	2	1
<b>Developer</b>	0	0	1	2	4	4	3
<b>Tester</b>	0	0	0	0	2	2	2
<b>Total</b>	4	4	4	5	8	8	6

- **Critical Path Analysis**

This tool is useful in recognizing interdependent tasks in the project. It also helps to find out the shortest path or critical path to complete the project successfully.

Like PERT diagram, each event is allotted a specific time frame. This tool shows dependency of event assuming an event can proceed to next only if the previous one is completed.

The events are arranged according to their earliest possible start time. Path between start and end node is critical path which cannot be further reduced and all events require to be executed in same order.



## UNIT-II

### 2 Mark

#### **1. Define project planning.**

Project planning is a procedural step in project management, where required documentation is created to ensure successful project completion.

Documentation includes all actions required to define, prepare, integrate and coordinate additional plans.

The project plan clearly defines how the project is executed, monitored, controlled and closed.

#### **2. What is meant by top down approach?**

Top-down design takes the whole software system as one entity and then decomposes it to achieve more than one sub-system or component based on some characteristics.

#### **3. Define bottom up approach.**

The bottom up design model starts with most specific and basic components. It proceeds with composing higher level of components by using basic or lower level components.

#### **4. Define project activities.**

**Project Activities** are the smallest identifiable and measurable pieces of work planned for completion throughout a project.

They aim to accomplish project work by converting available inputs into desired outputs, while consuming allocated resources.

Project activity is often used as an alternative term to project task.

#### **5. Define critical path.**

The technical definition of the **critical path** in a sequence of networked work packages is the path with the least amount of slack.

In practical terms, this path is the sequence of events that if any are delayed, will delay the entire project.

And in even simpler terms, the critical path is the sequence of tasks that will take the longest to complete to deliver the project.

## **6. Define stakeholder.**

A person, group or organization that has interest or concern in an organisation, stakeholders can affect or to be affected by the organization's actions, objectives and policies.

## **7. List out the software design levels.**

- Architectural Design
- High-level Design
- Detailed Design

## **8. Define architectural design.**

The architectural design is the highest abstract version of the system. It identifies the software as a system with many components interacting with each other.

At this level, the designers get the idea of proposed solution domain

## **9. Define high-level design.**

The high-level design breaks the 'single entity-multiple component' concept of architectural design into less-abstracted view of sub-systems and modules and depicts their interaction with each other.

## **10. Define detailed design.**

Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the previous two designs.

It is more detailed towards modules and their implementations.

It defines logical structure of each module and their interfaces to communicate with other modules.

## **5 Mark**

### **1. Explain the concept Project Activities.**

**Project Activities** are the smallest identifiable and measurable pieces of work planned for completion throughout a project.

They aim to accomplish project work by converting available inputs into desired outputs, while consuming allocated resources. Project activity is often used as an alternative term to project task.

Activities are the framework for differentiating between various components of a project.

In project management there is Activity-based Approach that assumes a project can be presented as a series of logically related, prioritized and sequenced activities that can be performed step-by-step or/and in parallel to complete project goals.

Activities of a project are always measurable, time-scaled and intended to produce certain results. They are characterized by the following items:

- Definite duration
- Logic relationships with other activities belonging to the same project
- Resource consumption
- Cost.

## **10 Mark**

### **1. Briefly explain about top-down and bottom-up approach.**

#### **Top Down Design**

Top-down design takes the whole software system as one entity and then decomposes it to achieve more than one sub-system or component based on some characteristics.

Each sub-system or component is then treated as a system and decomposed further. This process keeps on running until the lowest level of system in the top-down hierarchy is achieved.

Top-down design starts with a generalized model of system and keeps on defining the more specific part of it.

When all components are composed the whole system comes into existence.

Top-down design is more suitable when the software solution needs to be designed from scratch and specific details are unknown.

### **Bottom-up Design**

The bottom up design model starts with most specific and basic components.

It proceeds with composing higher level of components by using basic or lower level components. It keeps creating higher level components until the desired system is not evolved as one single component.

With each higher level, the amount of abstraction is increased.

Bottom-up strategy is more suitable when a system needs to be created from some existing system, where the basic primitives can be used in the newer system.

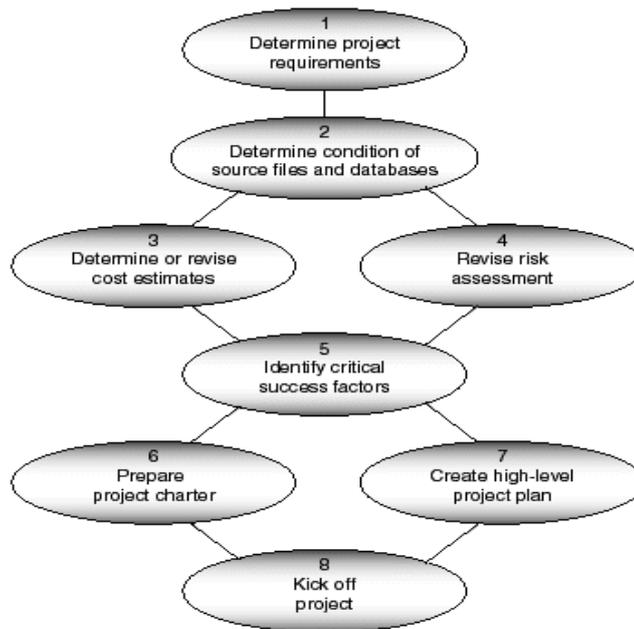
Both, top-down and bottom-up approaches are not practical individually. Instead, a good combination of both is used.

## **2. Discuss about project planning activities.**

- Determine the project requirements
- Determine the condition of the source files and databases
- Determine or revise the cost estimates
- Revise the risk assessment
- Identify critical success factors
- Prepare the project charter
- Create a high-level project plan
- Kick off the project

**Determine the project requirements.** You may have already prepared the objectives for the project and some high-level requirements for the proposed scope during Step 1, Business Case Assessment.

**Determine the condition of the source files and databases.** You can neither complete the project schedule nor commit to a delivery date without a good understanding of the condition of the source files and databases.



**Determine or revise the cost estimates.** Detailed cost estimates must include hardware and network costs as well as purchase prices and annual maintenance fees for tools.

**Revise the risk assessment.** Review and revise the risk assessment performed during Step 1, Business Case Assessment.

**Identify critical success factors.** A critical success factor is a condition that must exist for the project to have a high chance for success.

**Prepare the project charter.** The project charter is similar to a scope agreement, a document of understanding, or a statement of work.

**Create a high-level project plan.** Project plans are usually presented in the form of a Gantt chart that shows activities, tasks, resources, dependencies, and effort mapped out on a calendar.

**Kick off the project.** Once you have planned the project, assigned the resources, and scheduled the training, you are ready to kick off the project.

## UNIT-III

### 2 Mark

1. **Define project tracking.**

**Project Tracking** refers to the **management** of **projects**, which includes but is not limited to measuring and reporting the status of milestones, tasks and activities required in achieving the **pre-defined project** results. ... A **project** is a one-time effort to accomplish an explicit objective by a specific time.

2. **Write down the purpose of scope definition.**

The main purpose of the scope definition is to clearly describe the boundaries of your project.

Clearly describing the boundaries is not enough when it comes to project.

3. **List out project process.**

- Initiating
- Planning
- Executing
- Monitoring and Controlling
- Closing

4. **Define project meeting.**

Project meeting refer to an effective method to distribute information and communicate with the team and stakeholders.

5. **Define quality planning.**

Quality planning is used to determine how to ensure that the project quality objectives will be met.

6. **What is meant by risk assessment?**

Risk assessment is the process of quantifying and evaluating the probability of risk occurrence and risk impact.

## 5Mark

### **1. Explain project process.**

- Project Initiation
- Project Planning
- Project Execution
- Control and Validation

#### **Project Initiation**

Project initiation is the starting point of any project. In this process, all the activities related to winning a project takes place. Usually, the main activity of this phase is the pre-sale.

During the pre-sale period, the service provider proves the eligibility and ability of completing the project to the client and eventually wins the business.

During the requirements gathering activity, all the client requirements are gathered and analysed for implementation.

#### **Project Planning**

Project planning is one of the main project management processes.

If the project management team gets this step wrong, there could be heavy negative consequences during the next phases of the project.

In this process, the project plan is derived in order to address the project requirements such as, requirements scope, budget and timelines.

Once the project plan is derived, then the project schedule is developed.

#### **Project Execution**

During the project execution, there are many reporting activities to be done. The senior management of the company will require daily or weekly status updates on the project progress.

#### **Control and Validation**

During the project life cycle, the project activities should be thoroughly controlled and validated.

The controlling can be mainly done by adhering to the initial protocols such as project plan, quality assurance test plan and communication plan for the project.

### **Closeout and Evaluation**

Once all the project requirements are achieved, it is time to hand over the implemented system and closeout the project.

If the project deliveries are in par with the acceptance criteria defined by the client, the project will be duly accepted and paid by the customer.

Once the project closeout takes place, it is time to evaluate the entire project.

## **2. Write down the steps for project planning.**

**Step 0:** Select the project

**Step 1:** Identify project scope and objectives

**Step 2:** Identify the project infrastructure

**Step 3:** Analyse project characteristics

**Step 4:** Identify project products and activities

**Step 5:** Estimate Effort for each activity

**Step 6:** Identify activity risks

**Step 7:** Allocate resources

**Step 8:** Review/Publicize plan

**Step 9&10:** Execute plan/lower level planning.

## **10 Mark**

### **1. Briefly discuss about project tracking.**

Project Tracking refers to the management of projects, which includes but is not limited to measuring and reporting the status of milestones, tasks and activities required in achieving the pre-defined project results.

Project Tracking can also refer to Project Management software, which automates the tracking of tasks, assignments, events and activities related to the project.

A project is a one-time effort to accomplish an explicit objective by a specific time.

Each project is unique although similar projects may exist. Every project has a definite beginning and a definite ending. Project management is the process of overseeing planning, organizing, scheduling, leading, communicating and controlling of activities to achieve the pre-defined outcome on time and within budget.

One of the key components of Project Management is controlling.

There are various tools and techniques available to manage and track projects.

Project management software is frequently used by companies to assist in managing the initiation, execution, tracking and closing of projects.

By automating many of the project management tasks such as scheduling, time and expense reporting, and charge back reporting, project managers have access to real-time analytics, which ensure the efficient management of resource allocation and utilization.

Automation also assists project managers in consistent project execution, improved resource utilization and reduced resource gaps, operational efficiency and it facilitates communication and decision-making.

## UNIT-IV

### 2 Mark

#### **1. Define product requirements document.**

A **product requirements document (PRD)** is a document containing all the requirements to a certain product. It is written to allow people to understand *what* a product should do.

#### **2. Define the objective of project.**

The project objective describes the project's outcomes: intended and direct, short- and medium-term effects on the target group.

The project objective must lie within the scope of the project, and one must be able to directly attribute the effects to the project.

#### **3. What is meant by software testing?**

Software testing is the process of evaluation a software item to detect differences between given input and expected output.

Also to assess the feature of A software item.

Testing assesses the quality of the product. Software testing is a process that should be done during the development process.

In other words software testing is a verification and validation process.

#### **4. Define verification.**

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase.

In other words, to make sure the product behaves the way we want it to.

#### **5. Define Validation.**

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase.

In other words, to make sure the product is built as per customer requirements.

## **6. Write down the types of software testing.**

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

## **7. What is meant by block box and white box testing?**

### **Blackbox Testing**

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

### **Whitebox Testing**

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

## **8. Define unit testing.**

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing.

## **9. What is meant by system testing?**

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works.

System testing is done with full system implementation and environment. It falls under the class of black box testing.

## **10. Define functional testing.**

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

**11. List out the types of requirements.**



**12. What is meant by software requirement specification?**

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

It is usually signed off at the end of requirements engineering phase.

**13. List out the qualities of SRS.**

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

## 5Mark

### **1. Write a short note on software testing.**

Software testing is the process of evaluation a software item to detect differences between given input and expected output.

Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process.

In other words software testing is a verification and validation process.

#### ***Verification***

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

#### ***Validation***

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

#### ***Basics of software testing***

There are two basics of software testing: black box testing and white box testing.

#### **Black box Testing**

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

#### **White box Testing**

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

### ***Types of testing***

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

## **2. Write a short note on objective of software.**

The project objective describes the project's outcomes: intended and direct, short- and medium-term effects on the target group.

The project objective must lie within the scope of the project, and one must be able to directly attribute the effects to the project.

The project objective is often formulated in terms of the project's utility for the target group: "Better... higher..." It also makes sense to formulate the project objective as a situation to be achieved in the future.

The project objective ought also to describe an outcome, meaning the effect or change that the project is supposed to cause for the target group.

### **A well-formulated project objective**

- Provides a concrete description of the project's effect at the outcome level;
- Was developed in a participatory process;
- Is accepted by the target group and other stakeholders;
- Is clear and concise.

## 10Mark

### **1. Briefly discuss about types of software testing.**

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

#### **Unit Testing**

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing.

#### **Integration Testing**

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation.

It may fall under both white box testing and black box testing.

#### **Functional Testing**

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

## **System Testing**

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

## **Stress Testing**

Stress testing is the testing to evaluate how system behaves under unfavourable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

## **Performance Testing**

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

## **Usability Testing**

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? This falls under the class of black box testing.

## **Acceptance Testing**

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

## **Regression Testing**

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

## **Beta Testing**

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

## **2. Discuss about software requirements.**

Requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

Broadly software requirements should be categorized in two categories:

### **Functional Requirements**

Requirements, which are related to functional aspect of software fall into this category.

They define functions and functionality within and from the software system.

### **EXAMPLES**

- Search option given to user to search from various invoices.
- User should be able to mail any report to management.
- Users can be divided into groups and groups can be given separate rights.
- Should comply business rules and administrative functions.
- Software is developed keeping downward compatibility intact.

### **Non-Functional Requirements**

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

Non-functional requirements include -

- Security
- Logging
- Storage
- Configuration
- Performance
- Cost
- Interoperability
- Flexibility
- Disaster recovery
- Accessibility

Requirements are categorized logically as

- **Must Have** : Software cannot be said operational without them.
- **Should have** : Enhancing the functionality of software.
- **Could have** : Software can still properly function with these requirements.
- **Wish list** : These requirements do not map to any objectives of software.

While developing software, ‘Must have’ must be implemented, ‘Should have’ is a matter of debate with stakeholders and negotiation, whereas ‘could have’ and ‘wish list’ can be kept for software updates.

## UNIT V

### 5Mark

#### **1. Define software Quality.**

Quality software is reasonably bug or defect free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable.

#### **2. What is meant by software quality assurance?**

Software quality assurance (SQA) is a process that ensures that developed software meets and complies with defined or standardized quality specifications.

SQA is an on-going process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures.

#### **3. Define software review.**

A **software review** is "A process or meeting during which a software product is examined by a project personnel, managers, users, customers, user representatives, or other interested parties for comment or approval".

#### **4. List out the key aspects of quality.**

- Good design – looks and style
- Good functionality – it does the job well
- Reliable – acceptable level of breakdowns or failure
- Consistency
- Durable – lasts as long as it should
- Good after sales service
- Value for money

#### **5. List out types of software reviews.**

Formal technical review

Informal technical review

## **6. Define formal technical review.**

Technical review should be done by the team of members. The document, which is going to be reviewed, who has prepared and reviewers should sit together and do the review of that document. It is called Peer Review.

If it is a technical document, It can be called as formal Technical review, I guess. It varies depends on the company policy.

## **7. What are non-functional requirements?**

The non-functional requirements of a software product are: reliability, usability, efficiency, delivery time, software development environment, security requirements, standards to be followed etc.

## **8. Define software reliability.**

It is the probability that software will work without failure for a specified period of time in a specified environment.

Reliability of software is measured in terms of Mean Time Between Failure (MTBF). For eg if MTBF = 10000 hours for an average software, then it should not fail for 10000 hours of continuous operation.

## **5Mark**

### **1. Explain the key aspects of software quality.**

Key aspects of quality for the customer include:

- Good design – looks and style
- Good functionality – it does the job well
- Reliable – acceptable level of breakdowns or failure
- Consistency
- Durable – lasts as long as it should
- Good after sales service
- Value for money

### **Good design – looks and style:**

It is very important to have a good design. The application or product should meet all the requirement specifications and at the same time it should be user friendly.

### **Good functionality – it does the job well:**

Along with the good looks of the application or the product it's very important that the functionality should be intact.

### **Reliable – acceptable level of breakdowns or failure:**

After we have tested for all the features and their functionalities it also very important that the application or product should be reliable.

### **Consistency:**

The software should have consistency across the application or product. Single software can be multi-dimensional. It is very important that all the different dimensions should behave in a consistent manner.

### **Durable – lasts as long as it should:**

The software should be durable. The software product or application should continue to behave in the same way without any functional breaks.

### **Good after sales service:**

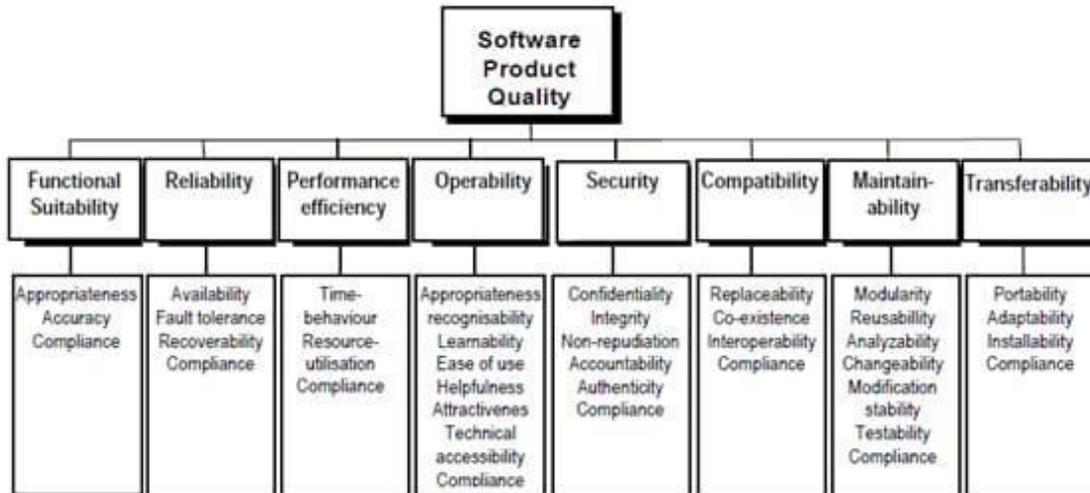
Once the product is shipped to the customers then maintenance comes into the picture. It is very important to provide good sales services to keep the customers happy and satisfied.

### **Value for money:**

It's always important to deliver the product to the customers which have value for money.

## 10Mark

### 1. Explain the concept software product quality.



- **Functional Suitability** (functional appropriateness) – Does the function facilitate the completion of the user’s task(s) and objectives?
- **Performance Efficiency** (time behavior) – Does the printer function respond within three seconds?
- **Compatibility** (interoperability) – Can the user print over a variety of networks and printers and on computers with different operating systems
- **Usability** (learnability) – Can the user figure out how to print or will it take a rocket scientist?
- **Reliability** (recoverability) – When the printer is unplugged in the middle of printing a task, is the user notified?
- **Security** (non-repudiation) – Is there a record that the printer printed the file successfully?
- **Maintainability** (testability) – Can test criteria be specified for the print function?
- **Portability** (adaptability) – Can the software automatically adapt to new printer models, or an update in printer driver software?

## 2. Explain the concept software reviews.

### 1. Walkthrough:

- It is not a formal process
- It is led by the authors
- Author guide the participants through the document according to his or her thought process to achieve a common understanding and to gather feedback.
- Useful for the people if they are not from the software discipline, who are not used to or cannot easily understand software development process.
- Is especially useful for higher level documents like requirement specification, etc.

The goals of a walkthrough:

- To present the documents both within and outside the software discipline in order to gather the information regarding the topic under documentation.
- To explain or do the knowledge transfer and evaluate the contents of the document
- To achieve a common understanding and to gather feedback.
- To examine and discuss the validity of the proposed solutions

### 2. Technical review:

- It is less formal review
- It is led by the trained moderator but can also be led by a technical expert
- It is often performed as a peer review without management participation
- Defects are found by the experts (such as architects, designers, key users) who focus on the content of the document.
- In practice, technical reviews vary from quite informal to very formal

The goals of the technical review are:

- To ensure that an early stage the technical concepts are used correctly
- To access the value of technical concepts and alternatives in the product
- To have consistency in the use and representation of technical concepts

- To inform participants about the technical content of the document

### 3. Inspection:

- It is the most formal review type
- It is led by the trained moderators
- During inspection the documents are prepared and checked thoroughly by the reviewers before the meeting
- It involves peers to examine the product
- A separate preparation is carried out during which the product is examined and the defects are found
- The defects found are documented in a logging list or issue log
- A formal follow-up is carried out by the moderator applying exit criteria

The goals of inspection are:

- It helps the author to improve the quality of the document under inspection
- It removes defects efficiently and as early as possible
- It improve product quality
- It create common understanding by exchanging information
- It learn from defects found and prevent the occurrence of similar defects