

- Scope
- Objectives
- Phases
- Responsibilities
- Matrices



Date :  
Page No. :

\* How do you differentiate Quality management with Testing in software engineering.

### 1. Scope

- Quality management in software engineering encompasses the entire software development life cycle (SDLC). It involves establishing and maintaining processes, standards, and practices that promote quality throughout the software development process.

### • Testing In Software Engineering.

Testing is a specific subset of quality assurance activities within software engineering. It involves the systematic process of evaluating a software application or system to identify defects, verify that it meets specified requirements, and assess its overall quality.

### 2. Objectives

- The main objective of quality management in software engineering is to establish a culture of quality within the development organization. It aims to prevent defects by improving processes, managing risks, and ensuring the entire development team is committed to delivering high-quality software.

- The primary objective of testing is to find and report defects or issues in the software product.

Testing verifies that the software functions correctly and conforms to the defined requirements.

### 3. Phases

Quality management activities occur throughout the entire SDLC, from project initiation to maintenance and support. It includes process definition, training, risk management, and continuous process improvement.

- Testing is typically concentrated in specific phases of the SDLC, such as unit testing, integrated testing, system testing and each testing phase has its objectives and scope.

### 4. Responsibilities

- Quality management is a shared responsibility across the development organization. It involves roles like quality managers, process owners, project managers.
- Testing is primarily the responsibility of dedicated testing teams or individuals who design test cases, create tests, report defects.



## Metrics :

- Quality management uses a wide range of metrics, and improve the overall quality of the development process.

- Testing metrics typically include metrics related to test coverage, defect detection rate and test case effectiveness.

\* What is Risk management. Discuss various project and product & business risks.

- Risk management is a systematic process of identifying, assessing, prioritizing and mitigating risks to minimize their impact on a project, product or business.

## Project Risks

- 1) Scope Creep.
- 2) Resource constraints.
- 3) Schedule delays.
- 4) Budget Overruns.
- 5) Quality issues.
- 6) Technical risks.

- The risk that project requirements will continuously change and expand, leading to scope creep, which can impact project timelines and budgets.

- The risk of not having enough resources to complete the project on time and within budget.
- Risks such as technical challenges, supplier delays or resource shortages can lead to project schedule delays.
- The risk of exceeding the allocated budget due to unexpected costs, or inaccurate cost estimation.
- The risk of delivering a product that doesn't meet quality standards or customer expectations.
- Uncertainties related to technology, the potential for software bugs, hardware failures or compatibility issues.

#### → Product risks:

- Market acceptance
- Technological obsolescence
- Supply chain disruptions
- Competitive pressure
- Intellectual Property (IP) issues.

#### → Business risks:

- Economic factors
- Market volatility
- Financial risks
- Operational risks
- Cybersecurity threats.

3)

Explain Lehman's law in Detail.

Lehman's law of software evolution, formulated by Dr. Meir M. Lehman, are a set of observations and principles that describe how software systems evolve over time.

These laws help us understand the dynamics of software systems and how they change as they are maintained & modified.

There are eight laws in total.

1. E1 - Continuing change
2. E2 - Increasing complexity
3. E3 - Self-regulation
4. E4 - Conservation of organizational stability
5. E5 - Conservation of familiarity
6. E6 - Continuing growth
7. E7 - Declining quality
8. E8 - Feedback system.

4) \*

Characteristics of legacy System.

- Legacy systems are older software or hardware systems that have been in use for an extended period and are often considered outdated or less compatible with modern technology and business requirements.



1. Outdated Technology
2. Limited or No vendor support
3. Customization & Complexity
4. Incompatibility
5. Performance & Scalability
6. Security risks.
7. High maintenance cost.

3) \* Socio-technical system are complex in its design . justify :

- Socio-technical systems are indeed complex in their design. and this complexity arises from the intricate interplay between social and technical components within these systems.

1. Human Interaction
2. Social Dynamics
3. Technical components
4. Adoption and Evolution
5. Feedback loops
6. Non-linear relationships
7. Scale and scope.



6) Write a short note on ~~don't do it~~ ~~software development~~.

a) Test driven development.

Test Driven Development (TDD) is a software development methodology that emphasizes writing test before writing the actual code or functionality.

1. write a Test :

Begin by writing a test case that defines the desired behaviour of a specific piece of code or a ~~feature~~ feature, this test should fail initially because the code hasn't fulfill the ~~test~~ test doesn't exist yet.

2. Write the code

Develop the minimum amount ~~one~~ of code necessary to make the failing test pass. The code is often written incrementally to meet the test requirements.

3. Run the test:

Execute the test to ensure that it now passes, if it passes, it means code meets the specified requirements.

4. Refactor

After a test passes, you can refactor the code to improve its design or efficiency while keeping the test suit as safety net.

b. System procurement:

System procurement refers to the process of acquiring software or hardware systems from external vendors or suppliers to meet the needs of an organization.

1. Needs assessment

The organization identifies its requirements & assesses its needs for new hardware or software system.

2. Market research

Research is conducted to identify potential vendors & solutions in the market place that can fulfill the organization's requirements.

3. Vendor Selection

The organization selects a vendor based on the evaluation results.

4. Contract Negotiation

Contract negotiation takes place to define terms, conditions, pricing, delivery schedules and support agreements.

c. Test suit

A test suit is ~~referred~~ a collection of test cases or test scripts that are grouped together for the purpose of testing a specific software component.

1. Test organization
2. Efficiency.
3. Reusability
- b. Reporting
5. Automation.

2) Concept of program evolution dynamics with focus on changing requirements.

A. The concept of program evolution dynamics, particularly with a focus on changing requirements, highlights how software systems undergo changes and adaptions over time in response to evolving user needs.

#### Scenario

Imagine a product called "Taskmaster" which is a task management application used by individuals and teams to track & manage their tasks & projects.

TaskMaster has been in use for several years & it evolved to ~~not~~ meet changing requirements.

Phase 1. Initial requirements.

Phase 2. User feedback & needs.

Phase 3. Continuous Evolution.

Phase 4. Longterm Sustainability.



### Topdown

- it estimates starts with a high level overview of the project or task.
- less accurate.
- More speed.
- need to prioritize projects.

### Bottom up.

- it estimates involves breaking down the project into smaller detailed components or tasks.
- more accurate.
- less speed.
- used for detailed project planning.