★ What is Software ?

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

★ What is Quality?

Quality means “Performance upon expectations” and “fit for functions.”

★ What is Software quality ?

Software Quality describes the desirable attributes of software products.

★ What is Software Quality Assurance ?

Software Quality Assurance (SQA) is simply a way to assure quality in the software.

Software Quality Assurance (SQA) is the practice of making sure software works well by checking and improving its development process.

★ IEEE Definition of "Software Quality" ?

The degree to which a system, component, or process meets specified requirements.

The degree to which a system, component, or process meets customer or user needs or expectations.

★ IEEE Definition of "Software Quality Assurance" ?

A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.

A set of activities designed to evaluate the process by which the products are developed or manufactured. Contrast with quality control.

★ What is Software Testing ?

Testing is the process of executing a program with the intent of finding errors.

★ Testing and Debugging

Testing is the process of finding defects in software.
Identifying errors in a program code and then fixing them up are known as debugging.
Debugging is the process that helps fix identified defects.

★ What is Software Review ?

A software review is a process where a software product is examined by team members, users, or stakeholders to provide feedback or approval.

Types of Reviews;

1. **Informal Review**

   Informal reviews take place between two or three people. The review conference is scheduled at their convenience.

   ex- walkthrough

- ➔ This meeting is generally scheduled during the free time of the team members.
- ➔ There is no planning for the meeting.
- ➔ If any errors occur, they are not corrected in the informal reviews.
- ➔ There is no guidance from the team.
- ➔ This review is less effective compared to the formal review.

2. **Formal Reviews**

Formal reviews take place among a team of three to five members. In the formal review, the members discuss the software model.

ex- Inspection

1. Fagan Inspections
2. Glib And Graham Inspection

- ➔ This meeting is scheduled beforehand. This gives the team members time to prepare.
- ➔ This meeting consists of a professional team that identifies and corrects errors in the model.
- ➔ This meeting does not exceed two hours.

Three Types of Review in Software Testing

1. **Software Peer Reviews**

Peer review is the process of evaluating the technical content and quality of a work product, typically conducted by the author and other developers.
Different types of Software Peer Reviews;

- ➔ Code Review
- ➔ Pair Programming
- ➔ Inspection
- ➔ Walkthrough
- ➔ Technical Review

2. **Software Management Reviews**

The management representatives are responsible for this type of review. The status of the work is evaluated and the decision by the activities of the software.

This review is very important in making a decision regarding software.

3. **Software Audit Reviews**

These are conducted by the personnel outside of the software project. They evaluate the software with specifications, standards, and other criteria.

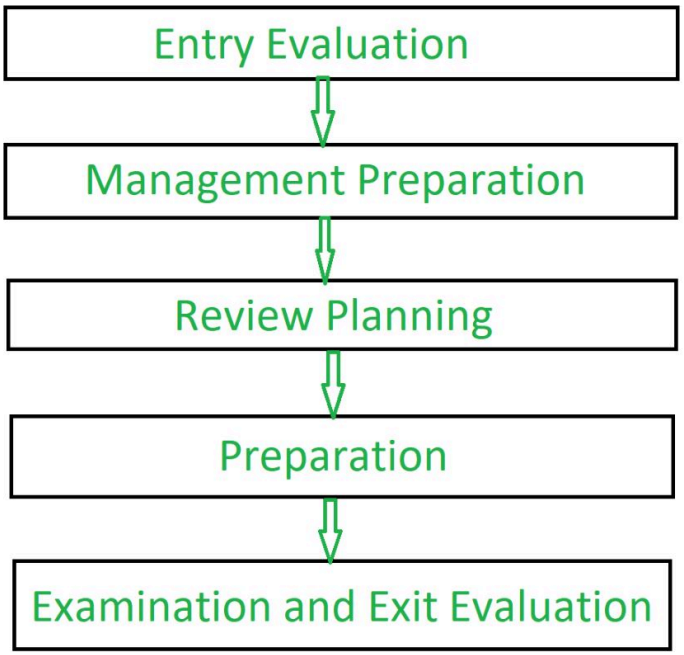
★ **Difference between Inspection and Walkthrough**

Inspection	Walkthrough
It is formal	It is informal
Initiated by project team	Initiated by author
A group of relevant persons from different departments participate in the inspection	Usually team members of the same project take participation in the walkthrough. Author himself acts walkthrough leader
Checklist is used to find faults	No checklist is used in the walkthrough
Inspection processes include overview, preparation, inspection, and rework and follow up.	Walkthrough process includes overview, little or no preparation, little or no preparation examination (actual walkthrough meeting), and rework and follow up.
Formalized procedure in each step.	No formalized procedure in the steps.

★ **The objective of the software review is**

1. To improve the productivity of the development team.
2. To make the testing process time and cost-effective.
3. To make the final software with less defects.
4. To eliminate the inadequacies.

★ **Process of Software Review**



Entry Evaluation: Confirm documentation and readiness to determine software availability.

Management Preparation: Assign roles, gather resources, and brief management.

Review Planning: Set goals, invite participants, and schedule the meeting.

Preparation: Distribute resources and give reviewers time to prepare.

Examination and Exit Evaluation: Examine results, record concerns, plan remediation, and evaluate effectiveness.

★ Advantages of Software Review

1. Reviews are cost-effective.
2. It can be used to train technical authors.
3. They can also reduce testing time and cost.
4. Earlier inspection also reduces the maintenance cost of software.
5. Reviews can improve software development productivity and reduce development timescales.

★ Software Verification Vs Validation

➔ **Verification is**

Verification typically involves reviews and meetings to evaluate documents, plans, code, requirements, and specifications.

Two Types of Verification

1. **Dynamic verification,**
   also known as Test or Experimentation - This is good for finding bugs
2. **Static verification,**
   also known as Analysis - This is useful for proving correctness of a program although it may result in false positives

➔ **Validation is**

Validation typically involves actual testing and takes place after verifications are completed.

Validation is the process of examining whether or not the software satisfies the user requirements.

★ What is Static Testing ?

Static Testing is a software testing technique that involves reviewing and analyzing

There are two basic types of static testing.

1. **People-based -** People based techniques are generally known as “reviews" There are different ways in which reviews can be performed.
2. **Tool-based -**The tool-based techniques examine source code and are known as "static analysis"

★ Why Static Testing?

Static Testing is a valuable technique because it helps detect errors and defects in software before it is tested.

if you review code during Static Testing, you may spot issues like
coding errors,
inconsistent naming conventions

★ Here are some major features that are subject to static testing

1. Requirements & Specifications Review
2. Structural Design Analysis
3. Code Review
4. Database Analysis
5. Documentation Review
6. User Manual Review

★ Static Testing Techniques

1. **Review**

   Reviews are essential in static testing, allowing testers to identify defects and issues in documentation, such as requirements and design.

   They are crucial for detecting sources of failure early on.

   In static testing, reviews can be divided into four different parts, which are as follows:
   - ➔ Informal reviews
   - ➔ Walkthroughs
   - ➔ Technical/peer review
   - ➔ Inspections

2. **Static Analysis**

   Static Testing technique is static analysis, which is used to contain the assessment of the code quality, which is established by developers.

   The static analysis will also help us to identify the below errors:

   - ➔ Dead code
   - ➔ Unused variables
   - ➔ Endless loops
   - ➔ Incorrect syntax
   - ➔ Variable with undefined value

★ Benefits of Static Testing

1. Early defect detection
2. Improved code quality
3. Reduced costs and time
4. Prevention of common issues
5. Improved collaboration

★ Type of Static Testing

1. Manual Static Testing

Manual static testing methods are techniques used to identify errors and defects in software code and documentation without actually executing the code

1. Inspections
2. Walkthroughs
3. Informal reviews
4. Technical reviews
5. Audits

2. Automated Static Testing

This method leverages specialized tools to analyze source code, documentation, and other artifacts without execution.

Tools dissect the code's structure, logic, and flow, identifying potential errors like:

1. Syntactic errors
2. Logic errors
3. Security vulnerabilities
4. Code quality issues
5. Coding standard violations
6. Code Reviews & Lint Checks
7. Formal Methods

★ Tools Used for Static Testing

1. Checkstyle
2. SourceMeter
3. Soot
4. Lint
5. SonarQube
6. PDM
7. Findings

★ What is Tested in Static Testing?

Code Quality

- ➔ Syntax errors
- ➔ Coding standards
- ➔ Potential bugs
- ➔ Performance bottlenecks
- ➔ Security vulnerabilities

Documentation Quality

- ➔ Requirements completeness and consistency
- ➔ Design document correctness
- ➔ Test plan completeness and feasibility

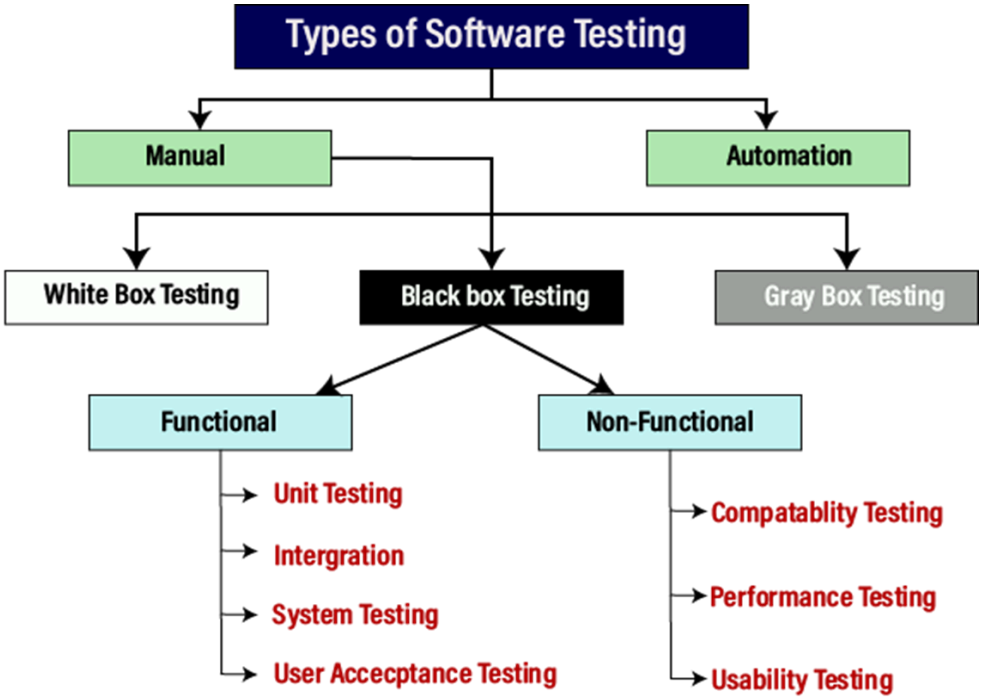
★ Advantages of Software Review

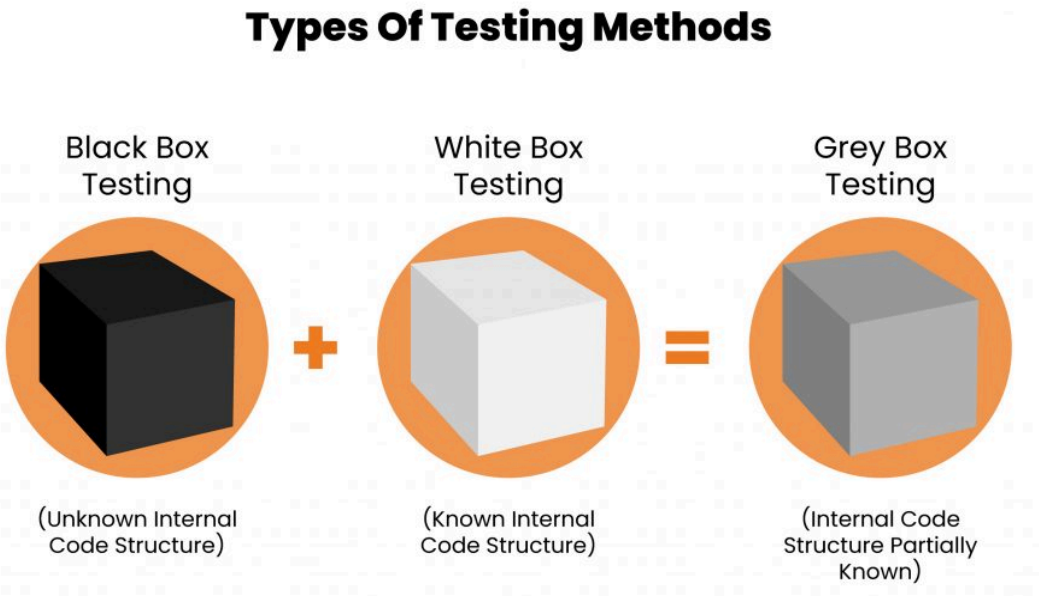
1. Reduced maintenance costs.
2. Reduced testing time and cost.
3. It can be used to train technical authors.
4. Improved productivity and shorter development timescales.
5. Reviews are cost-effective.

★ What is Software Testing ?

Software Testing is a method to assess the functionality of the software program. The process checks whether the actual software matches the expected requirements and ensures the software is bug-free.

★ Type of Software Testing





1. Black Box Testing

Tester tests the application software by applying different inputs and comparing the output with expected results.
It is also known as Behavioural Testing.

Ex - Appium, Selenium, Microsoft Coded UI, Applitools, HP QTP.

Advantages of Black Box Testing

- ➔ No need for extensive functional knowledge or programming skills.
- ➔ Efficient for testing larger systems.
- ➔ Executed from the user's or client's perspective.
- ➔ Test cases are easily reproducible.
- ➔ Identifies ambiguities and contradictions in functional specifications.

Disadvantages of Black Box Testing

- ➔ There is a possibility of repeating the same tests while implementing the testing process.
- ➔ Complex inputs make test case execution difficult.
- ➔ Test failure reasons can be hard to detect.
- ➔ Some programs in the application are not tested.
- ➔ Does not reveal errors in control structures.
- ➔ Difficult to implement test cases without clear functional specifications.
- ➔ Testing with a large sample space is time-consuming and exhaustive.

2. White Box Testing

White box testing is a verification technique where software engineers can use to examine.if the code works as expected or not.

Ex- Wireshark,PyUnit,Nunit,Nmap

Techniques in White Box Testing

- ➔ Statement Coverage
- ➔ Branch Coverage
- ➔ Condition Coverage
- ➔ Multiple Condition Coverage
- ➔ Basic Path Testing
- ➔ Loop Testing

Advantages of Whitebox Testing

- ➔ Thorough Testing
- ➔ Code Optimization
- ➔ Early Detection of Defects
- ➔ Integration with SDLC
- ➔ Detection of Complex Defects
- ➔ Comprehensive Test Cases

Disadvantages of White box Testing

- ➔ Programming Knowledge and Source Code Access
- ➔ Overemphasis on Internal Workings
- ➔ Bias in Testing
- ➔ Test Case Overhead
- ➔ Dependency on Tester Expertise
- ➔ Inability to Detect Missing Functionalities
- ➔ Increased Production Errors

3. **Gray Box Testing**

It is a software testing methodology to test the software with partial knowledge of the code in the application. Gray box testing is a combination Black and White box testing.

★ What is a Test Case ?

A test case is a set of actions performed on a system to verify that it meets software requirements and functions correctly.

★ Why do we need test cases ?

1. To require consistency in the test case execution
2. To make sure a better test coverage
3. It depends on the process rather than on a person
4. To avoid training for every new test engineer on the product

★ Types of Test Cases

1. Positive Test Cases
2. Negative Test Cases

★ Elements of a Test Case

Followings are the elements of test cases:

- ➔ Test Case ID
- ➔ Test Scenario
- ➔ Test Case Description
- ➔ Test Steps
- ➔ Prerequisite
- ➔ Test Data
- ➔ Expected Result
- ➔ Actual Result
- ➔ Environment Information
- ➔ Comments

★ Advantages of Test Case Design Techniques

1. Test Design finds Faults in the early stage.
2. Faults found early are cheaper to fix.
3. Early test design techniques can prevent faults from entering into the system.
4. Change requests can be reduced considerably through test design.
5. Early test design helps to build quality and stops fault multiplication.

★ Automation Testing

Automation testing is the process of converting manual test cases into test scripts using automation tools.

This testing is a testing procedure done with aid of automated testing tools.

★ Why automation testing?

- ➔ Automated testing saves money and time.
- ➔ Speed of your test execution.
- ➔ Testing improves accuracy.
- ➔ Automated tests help developers and testers.
- ➔ Vastly increase your test coverage.
- ➔ Automation tests do what manual tests cannot.

★ Difference Between Manual Testing and Automated Testing

Manual Testing	Automated Testing
The test cases are executed by the human tester.	The test cases are executed by the software tools.
Manual testing is time-consuming.	Automation testing is faster than manual testing.
Manual testing is less reliable.	Automation testing is more reliable.
No need for programming knowledge.	Programming knowledge is a must.
Manual testing doesn't use frameworks.	Automation testing uses frameworks like Data Drive, Keyword, etc.

★ Different type of software test that can be automated

- ➔ Unit Testing.
- ➔ Integration Testing.
- ➔ Functional Testing.
- ➔ Keyword Testing.
- ➔ Regression Testing.
- ➔ Data – Driven Testing.
- ➔ Black-box testing.
- ➔ Smoke Testing.

★ Automation Testing Tools.

- ➔ Ranorex
- ➔ Sahi pro
- ➔ Apache JMeter
- ➔ Cypress
- ➔ Galen
- ➔ Karate – DSL
- ➔ Robot Framework
- ➔ Selenium

★ The Dark side of automation testing

- ➔ Automating the wrong things.
- ➔ The tools you picked can't solve your problems.
- ➔ The tools you selected aren't a good fit for your testers.
- ➔ You failed to add up the total cost of tool ownership.
- ➔ You chose a tool just because it's open source.
- ➔ You didn't foster a testing culture that embraces automation.

★ Performance testing and reporting with - Apache JMeter

JMeter is a software that can perform load test, performance test, regression test ,stress test etc.

★ The following protocols are supported by JMeter,

- ➔ Web - HTTP, HTTPS
- ➔ Web services - SOAP / REST
- ➔ FTP services
- ➔ Database via JDBC drivers
- ➔ Directory - LDAP
- ➔ Message-oriented middleware (MOM) via JMS
- ➔ Mail - SMTP(S), POP3(S) and IMAP(S)
- ➔ Native commands or shell scripts
- ➔ TCP

★ What can we do with JMeter ?

- ➔ Using command line parameters in JMeter for load testing.
- ➔ Generate Random variables in JMeter
- ➔ Database load testing with JMeter
- ➔ Run a stress test in JMeter
- ➔ Test SOAP services with JMeter

This short note making resources are,google search engine,chat gpt and lecture slides

WhatsApp group (click on this/new syllabus notes only) Credit - Kanishka viraj

➔ Load testing video streaming with JMeter
➔ Load test TCP protocol services with JMeter

★ How JMeter Works?

JMeter simulates a group of users sending requests to a target server, and returns statistics that show the performance/functionality of the target server/application via tables, graphs, etc.

★ Thread Group elements are the beginning points of your test plan.

➔ Set the number of threads
➔ Set the ramp-up period
➔ Set the number of times to execute the test

★ JMeter has two types of Controllers:

1. **Samplers**
   Samplers allow JMeter to send specific types of requests to a server. They simulate a user request for a page from the target server.

2. **Logic Controllers**
   Logic Controllers control the order of processing of Samplers in a Thread. Logic controllers can change the order of a request coming from any of their child elements.

★ Difference between Black Box Testing and White Box Testing

Black Box Testing	White Box Testing
Testing without knowledge of the internal structure or code	Testing with knowledge of the internal structure or code.
Implementation of code is not needed for black box testing.	Code implementation is necessary for white box testing.
Black Box Testing is mostly done by software testers.	White Box Testing is mostly done by software developers.
No knowledge of programming is required.	It is mandatory to have knowledge of programming.
Black Box Testing is least time consuming.	White Box Testing is most time consuming.
Types of Black Box Testing: • Functional Testing • Non-functional testing • Regression Testing	Types of White Box Testing: • Path Testing • Loop Testing • Condition testing

★ What is System Integration Testing (SIT) ?

System Integration Testing is a QA process that is leveraged to ensure the compatibility of two or more systems.

It helps to ensure that the systems are working together correctly and that any interactions are appropriate and safe.

★ **Benefits of System Integration Testing**

- ➔ Improved quality
- ➔ Increases efficiency
- ➔ Improved scalability

★ **System Integration Testing Techniques**

1. Top down approach
2. Bottom up approach
3. Bigbang approach
4. Incremental integration testing

★ **Objective of System Integration Testing**

- ➔ To meet software with user requirements.
- ➔ To maximize the memory usage.
- ➔ To control data flow within the system.
- ➔ To test the control flow of the system.
- ➔ To find the errors and bugs in the system.
- ➔ To minimize time consumption for the testing process.

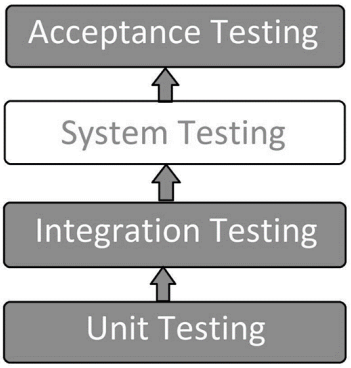
★ **There are three major states of SIT.**

1. Data state within the integration layer
2. Data state within the database layer
3. Data state within the application layer

★ **What is System Testing ?**

System testing assesses the complete software solution's functionality and performance.

System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution.



Types of System Testing

1. Regression Testing
2. Load Testing
3. Functional Testing
4. Recovery Testing
5. Migration Testing
6. Usability Testing
7. Software and Hardware Testing
8. Performance Testing

This short note making resources are,google search engine,chat gpt and lecture slides

WhatsApp group (click on this/new syllabus notes only) Credit - Kanishka viraj

9. Load Testing
10. Stress Testing
11. Scalability Testing

Tools used for System Testing

1. JMeter
2. Selenium
3. HP Quality Center/ALM
4. IBM Rational Quality Manager
5. Appium
6. LoadRunner
7. Apache JServ
8. SoapUI

★ Advantages of System Testing

1. Verifies the overall functionality of the system.
2. Improves system reliability and quality.
3. Enhances the overall performance of the system.
4. Increases user confidence and reduces risks.
5. Improves the system's maintainability and scalability.

★ Disadvantages of System Testing

1. Can be time-consuming and expensive.
2. Can be impacted by changes made during development.
3. Requires specialized skills and expertise.
4. Requires proper planning, coordination, and execution.
5. Limited visibility into the internal workings of the system.

★ What is Risk ?

Risk is a potential event, hazard, threat, or situation whose occurrence causes an adverse effect.

Type of Risks

1. **Project risks**

   ➔ Organizational issues (e.g., delays in work products deliveries, inaccurate estimates, cost-cutting)
   ➔ People issues (e.g., insufficient skills, conflicts, communication problems, shortage of staff)
   ➔ Technical issues (e.g., scope creep, poor tool support)
   ➔ Supplier issues (e.g., third-party delivery failure, bankruptcy of the supporting company)

2. **Product risks**

   ➔ Missing or wrong functionality
   ➔ incorrect calculations,
   ➔ Runtime errors
   ➔ Poor architecture
   ➔ Inefficient algorithms

➔ Response time
➔ Poor user experience
➔ Security

Product risk analysis aims to identify potential risks associated with a product to guide testing efforts and minimize overall risk.

Stakeholders can identify risks by using various techniques and tools,
E.g:
Brainstorming, workshops, interviews, or cause-effect diagrams.

★ What is Risk Management?

Risk Management is about identifying, assessing, and addressing potential threats to an organization's finances, assets, and operations.

The main goal of risk management is to predict possible risks and find solutions to deal with them successfully.

★ Risk Management Process

1. Risk Identification.
2. Risk Assessment.
3. Risk Planning.
4. Risk Monitoring.

★ Benefits of Risk Management

1. Helps protect against potential losses.
2. Improves decision-making by considering risks.
3. Reduces unexpected expenses.
4. Ensures adherence to laws and regulations.
5. Builds resilience against unexpected challenges.
6. Safeguards company reputation.

★ Limitation of Risk Management

1. Too much focus on risk can lead to missed opportunities.
2. Some risks are hard to predict or quantify.
3. Implementing risk management can be expensive.
4. Managing risks can take a lot of time and resources.
5. Risk models can be overly complex and hard to understand.
6. Having risk controls might make people feel too safe.
7. Relies on accurate human judgment and can be prone to mistakes.