

# MODULE 2 (Manual Testing)

## Q. What is Exploratory Testing?

A. It is a kind of software testing that aims to optimize and enhance the software program in every viable way. In this kind of testing the tester is free to choose any feasible methodology to check the software. It is usually performed by experienced testers who have a deep understanding of the system under test and can quickly identify defects.

## Q. What is traceability matrix?

A. A Software Testing Traceability Matrix (STM) is a document that links and maps test cases to their respective requirements, ensuring that each requirement has been adequately tested. It serves as a verification tool to confirm that all software requirements, as defined in the requirements specification document, are covered by test scenarios and cases.

The matrix facilitates identifying missing tests, understanding the impact of changes, and ensuring comprehensive test coverage.

## Q. What is Boundary value testing?

A. Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions.

- A boundary value for a valid partition is a valid boundary value.
- A boundary value for an invalid partition is an invalid boundary value.
- For each variable we check-
  1. Minimum value.
  2. Just above the minimum.
  3. Nominal Value.
  4. Just below Max value.
  5. Max value.

**Example:** Consider a system that accepts ages from 18 to 56.

Boundary Value Analysis (Age accepts 18 to 56)		
Invalid (min-1)	Valid (min, min + 1, nominal, max – 1, max)	Invalid (max + 1)
17	18, 19, 37, 55, 56	57

**Q. What is Equivalence partitioning testing?**

**A.** Equivalence Class Partitioning (ECP) is a black-box testing technique that divides input data into groups (or partitions) with similar behaviour. It is used to reduce the number of test cases while maintaining adequate coverage by testing one value from each group instead of testing all possible inputs.

**Q. What is Integration testing?**

**A.** It is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity.

**Q. What determines the level of risk?**

**A.** The risk level is determined by two dimensions: probability and impact. Probability: It measures the likelihood of an event occurring, typically expressed as a percentage or qualitative scale.

**Q. What is Alpha testing?**

**A.** Alpha testing is done to make sure a product is ready to send to potential end-users for beta testing. During Alpha testing, internal testers check the product for bugs and other quality issues. These internal testers include stakeholders, team members etc. Alpha testing occurs before a product launch.

**Q. What is beta testing?**

**A.** Beta testing is an opportunity for real users to use a product in a production environment to uncover any bugs or issues before a general release. Beta testing is the final round of testing before releasing a product to a wide audience.

**Q. What is component testing?**

**A.** It is a software testing technique that verifies the functionality, usability, and behaviour of individual components of an application in isolation to ensure they meet specified requirements. In component testing, each component is tested independently before integration with other modules.

**Q. What is functional system testing?**

**A.** Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

**Q. What is Non-Functional Testing?**

A. Non-functional testing is a type of software testing that verifies non-functional aspects of the product, such as performance, stability, and usability.

**Q. What is GUI Testing?**

A. Graphic User Interface Testing (GUI) testing is the process of ensuring proper functionality of the graphical user interface (GUI) for a specific application. This involves making sure it behaves in accordance with its requirements and works as expected across the range of supported platforms and devices.

**Q. What is Ad hoc testing?**

A. Performing random testing without any plan is known as Ad Hoc Testing. It is also referred to as Random Testing or Monkey Testing. This type of testing doesn't follow any documentation or plan to perform this activity.

**Q. What is white box testing and list the types of white box testing?**

A. White box testing techniques analyse the internal structures the used data structures, internal design, code structure, and the working of the software.

Types of white box testing includes,

- Statement coverage
- Branch coverage
- Condition coverage

**Q. What is black box testing? What are the different black box testing techniques?**

A. Black-box testing is a Type of Software Testing in which the tester is not concerned with the software's internal knowledge or implementation details but rather focuses on validating the functionality based on the provided specifications or requirements.

Blackbox test techniques includes:

- |                            |                             |
|----------------------------|-----------------------------|
| • Equivalence Partitioning | • State Transition Testing  |
| • Boundary Value Analysis  | • Use-case Testing          |
| • Decision Tables          | • Syntex or Pattern Testing |

**Q. Mention what are the categories of defects?**

A. Defects are typically categorized into three main types, Critical defects, Major defects and Minor defects, based on their severity and impact on product performance and safety.

**Q. Mention what big bang testing is?**

A. It is a type of integration testing that combines all the modules or components of a system into a single unit and tests them as a whole.

**Q. What is the purpose of exit criteria?**

A. The purpose of exit criteria is to prevent a task from being considered completed when there are still outstanding parts of the task which have not been finished. Exit criteria are used to report against and to plan when to stop testing.

**Q. When should “Regression Testing” be performed?**

A. Regression testing should be done after the software has been modified and when the environment has been modified.

**Q. What is 7 key principles? Explain in detail?**

A. From test planning to execution and analysis, understanding these principles is vital for delivering robust and reliable software solutions.

**1<sup>st</sup> Principle: Testing shows the Presence of Defects**

- Software testing talks about the presence of defects and doesn't talk about the absence of defects.
- Software testing can ensure that defects are present but it cannot prove that software is defect-free. Testing can reduce the number of defects but not remove all defects.

**2<sup>nd</sup> Principle: Exhaustive Testing is not Possible**

- Exhaustive testing is impossible means the software can never test at every test case.
- It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case.
- If the software will test every testcase then it will take more cost, effort, etc., which is impractical.

**3<sup>rd</sup> Principle: Early Testing**

- To find the defect in the software, early test activity shall be started. The defect detected in the early phases of SDLC will be very less expensive.
- For better performance of software, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.

**4<sup>th</sup> Principle: Defect Clustering**

In a project, a small number of modules can contain most of the defects. The Pareto Principle for software testing states that 80% of software defects come from 20% of modules.

### 5<sup>th</sup> Principle: Pesticide Paradox

Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

### 6<sup>th</sup> Principle: Testing is Context-Dependent

The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing. For example, the testing of the e-commerce site is different from the testing of the Android application.

### 7<sup>th</sup> Principle: Absence of Errors Fallacy

If a built software is 99% bug-free but does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfil all the customer requirements.

### Q. Difference between QA v/s QC v/s Testing.

QA (Quality Assurance)	QC (Quality Checker)	Testing
Focuses on processes and procedures rather than conducting actual testing on system.	Focusing on actual testing by executing software with intend to identify bug/defect through implementation of procedures and process.	Focusing on actual testing.
Process oriented activities.	Product oriented activities.	Product oriented activities.
Do preventive activities.	It is a corrective process.	It is a preventive process.
It is subset of STLC.	It is subset of QA.	It is subset of QC.
Activities which ensure the implementation of process, procedures and standards in context to verification of developed software & intended requirements.	Activities which ensure the verification of developed software with respect to documented requirements.	It includes activities which ensure the identification of bugs/error/defects in the software.

### Q. Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Smoke testing is used to ensure that the build is stable enough for further testing	Sanity testing is used to verify that specific functionality or components are working as expected after making changes or fixing defects.
Smoke testing is performed by either developers or testers.	Sanity testing is normally performed by testers.

Smoke testing is done to measure the stability of the system/product by performing testing.	Sanity testing is done to measure the rationality of the system/product by performing testing.
Smoke testing is used to test all over function of the system/product.	Sanity testing is used in the case of only modified or defect functions of system/products.
Smoke testing is performed by either developers or testers.	Sanity testing is normally performed by testers.

**Q. Difference between verification and Validation.**

**A. Verification** refers to the set of activities that ensure software correctly implements the specific function. **Validation** refers to the set of activities that ensure that the software that has been built is traceable to customer requirements.

Verification is the Static testing. Validation is Dynamic testing. It does not include the execution of the code. It includes the execution of the code.

**Q. What is Error, Defect, Bug and failure?**

**A.** We can say that a mistake made by a programmer during coding is called an error, an error found during the unit testing in the development phase is called a defect, an error found during the testing phase is called a bug and when an error is found at an end user's end is called as the failure.

**Q. Explain the difference between Functional testing and Non-Functional testing.**

Functional Testing	Non-Functional Testing
It is performed using the functional specification provided by the client and verifies the system against the functional requirements.	It verifies the behaviour of an application.
Functional testing is easy to execute manually.	It is hard to execute non-functional testing manually.
It tests what the product does.	It describes how the product does.
Functional testing is based on the business requirement.	Functional testing is based on the business requirement.
Example: Unit Testing Smoke Testing, Integration Testing, Regression Testing, UAT Testing	Example: Performance Testing, Load Testing, Security Testing, Stress Testing

**Q. What is the difference between the SDLC (Software Development Life Cycle) and STLC (Software Testing Life Cycle)?**

<b>SDLC (Software Development Life Cycle)</b>	<b>STLC (Software Testing Life Cycle)</b>
SDLC is mainly related to software development.	STLC is mainly related to software testing.
Besides development other phases like testing is also included.	It focuses only on testing the software.
SDLC involves total six phases or steps.	STLC involves only five phases or steps.
In SDLC, a greater number of members (developers) are required for the whole process.	In STLC, a smaller number of members (testers) are needed.
In SDLC, development team makes the plans and designs based on the requirements.	In STLC, testing team (Test Lead or Test Architect) makes the plans and designs.
Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
It helps in developing good quality software.	It helps in making the software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Post deployment support, enhancement, and update are to be included if necessary.	Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.
Creation of reusable software systems is the end result of SDLC.	A tested software system is the end result of STLC.

**Q. What is the difference between test scenarios, test cases, and test script?**

A. Test case is a document with instructions on testing the specific functionality of an application. Test Script is a program that runs various test data on the functionality of an application. Test scenarios serve as an outline for writing test cases.

**Q. Explain what Test Plan is? What is the information that should be covered.**

A. Test plan is a formal document that outlines the testing activities and strategies for evaluating the quality of a software system.

**It includes** information such as,

- Approach and methodology
- Scope and objectives
- Resources and timelines
- Risks and mitigation plans

**Q. What are the different Methodologies in Agile Development Model?**

A. The different methodologies in Agile Development model includes,

- Kanban.
- Scrum.
- Extreme Programming (XP)
- Feature-Driven Development (FDD)
- Lean.
- Dynamic Systems Development Method (DSDM)
- Crystal.

**Q. Explain the difference between Authorization and Authentication in Web testing.**

<b>Authentication</b>	<b>Authorization</b>
Authentication verifies who the user is.	Authorization determines what resources a user can access.
Authentication works through passwords, one time pins, biometric information, and other information provided or entered by the user.	Authorization works through settings that are implemented and maintained by the organization.
Authentication is the first step of a good identity and access management process.	Authorization always takes place after authentication.
Authentication is visible to and partially changeable by the user.	Authorization isn't visible to or changeable by the user.



Example: By verifying their identity, employees can gain access to a human resources (HR) application that includes their personal pay information, vacation time, and 401K data.	Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization.
---	--

**Q. What are the common problems faced in Web testing?**

A. The common problems faced in web testing includes, browser compatibility, responsiveness, system integration, performance, security and usability.