

## **What is software testing?**

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

## **What is Exploratory Testing?**

In exploratory testing tester focuses more on how the software actually works, testers do minimum planning and maximum execution of the software by which they get in depth idea about the software functionality, once the tester starts getting insight into the software he can make decisions to what to test next. Exploratory testing is mostly used if the requirements are incomplete and time to release the software is less.

## **What is Boundary value testing?**

Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions. The behavior at the edge of the equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.

It checks for the input values near the boundary that have a higher chance of error. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition.

## **What is traceability matrix?**

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship.

## **What is Equivalence partitioning testing?**

Equivalence Partitioning Method is also known as Equivalence class partitioning (ECP). It is a software testing technique or black-box testing that divides input domain into classes of data, and with the help of these classes of data, test cases can be derived. An ideal test case identifies class of error that might require many arbitrary test cases to be executed before general error is observed.

In equivalence partitioning, equivalence classes are evaluated for given input conditions. Whenever any input is given, then type of input condition is checked, then for this input conditions, Equivalence class represents or describes set of valid or invalid states.

## **What is Integration testing?**

Integration testing -- also known as integration and testing (I&T) -- is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity. However, these modules may be coded by different programmers.

## **What determines the level of risk?**

The probability of any unwanted incident is defined as Risk. In Software Testing, risk analysis is the process of identifying the risks in applications or software that you built and prioritizing them to test. After that, the process of assigning the level of risk is done. The categorization of the risks takes place, hence, the impact of the risk is calculated.

## **What is Alpha testing?**

Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the user acceptance testings. This is referred to as alpha testing only because it is done early on, near the end of the development of the software. Alpha testing is commonly performed by homestead software engineers or quality assurance staff. It is the last testing stage before the software is released into the real world.

## **What is beta testing?**

Beta testing is a type of user acceptance testing where the product team gives a nearly finished product to a group of target users to evaluate product performance in the real world.

## **What is component testing?**

Component testing, also known as program or module testing, is done after unit testing. In this type of testing those test objects can be tested independently as a component without integrating with other components e.g. modules, classes, objects, and programs. This testing is done by the development team.

## **What is functional system testing?**

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.

## **What is Non-Functional Testing?**

Non-functional testing is a type of software testing to test non-functional parameters such as reliability, load test, performance and accountability of the software. The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters. The parameters of non-functional testing are never tested before the functional testing.

## **What is GUI Testing?**

GUI Testing is a software testing type that checks the Graphical User Interface of the Software. The purpose of Graphical User Interface (GUI) Testing is to ensure the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc.

## **What is Adhoc testing?**

Adhoc testing is a term that refers to testing that is done on the fly. It is a form of unstructured testing technique, as described above, in which no systematic strategy is made before the testing process begins. As a result, no requirement specification or test case preparation and design is done prior to testing.

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

White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

- Path testing
- Loop testing
- Condition testing
- Testing based on the memory perspective
- Test performance of the program

## **What is black box testing? What are the different black box testing techniques?**

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test.

- Equivalent Partitioning
- Boundary value analysis
- Decision transition testing
- Use case testing
- State transition testing

## **Mention what are the categories of defects?**

- Arithmetic Defects
- Logical Defects
- Syntax Defects

- Multithreading Defects
- Interface Defects
- Performance Defects

## **Mention what bigbang testing is?**

Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

## **What is the purpose of exit criteria?**

Purpose of exit criteria is to define when we STOP testing either at the:

- End of all testing – i.e. product Go Live
- End of phase of testing (e.g. hand over from System Test to UAT)

## **What is 7 key principles? Explain in detail?**

- Testing shows presence of Defects
- Exhaustive Testing is Impossible!
- Early Testing
- Defect Clustering
- The Pesticide Paradox
- Testing is Context Dependent
- Absence of Errors Fallacy

### Testing shows presence of Defects

- Testing can show that defects are present, but cannot prove that there are no defects.
- Testing reduces the probability of undiscovered defects remaining in the software but,
- even if no defects are found, it is not a proof of correctness.
- We test to find Faults
- As we find more defects, the probability of undiscovered defects remaining in a system
- reduces.

### Exhaustive Testing is Impossible

- Testing everything including all combinations of inputs and preconditions is not
- possible.
- So, instead of doing the exhaustive testing we can use risks and priorities to focus testing
- efforts.
- For example: In an application in one screen there are 15 input fields, each having 5
- possible values, then to test all the valid combinations you would need 30 517 578 125
- (515) tests.
- This is very unlikely that the project timescales would allow for this number of tests.

- So, accessing and managing risk is one of the most important activities and reason for testing in any project.
- We have learned that we cannot test everything (i.e. all combinations of inputs and preconditions).
- That is we must Prioritise our testing effort using a Risk Based Approach.

#### Early Testing

- Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
- Testing activities should start as early as possible in the development life cycle
- These activities should be focused on defined objectives – outlined in the Test Strategy
- Remember from our Definition of Testing, that Testing doesn't start once the code has been written!

#### Defect Clustering

- A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
- Defects are not evenly spread in a system.
- In other words, most defects found during testing are usually confined to a small number of modules

#### The Pesticide Paradox

- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
- To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

#### Absence of Errors Fallacy

- If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.

### **Difference between QA v/s QC v/s Tester**

Quality Assurance	Quality Tester
Aim to prevent defect	Aim to identify and fix defect
Is a prevent technique	Is a corrective technique
It is the duty of complete project team	It is the duty of testing team
This is managerial tool	This is corrective tool

### **Difference between Smoke and Sanity?**

Smoke Testing	Sanity Testing
Smoke testing exercises the <b>entire system</b> from end to end.	Sanity testing exercises only the <b>particular component</b> of the entire system.
The main objective of the testing is to verify the <b>stability</b> of the system.	The main objective of the testing is to verify the <b>rationality</b> of the system.
Smoke Testing is performed to ascertain that	Sanity testing is done at random to verify

the <b>critical functionalities</b> of the program are <b>working fine</b> .	that <b>each functionality</b> is <b>working as expected</b> .
------------------------------------------------------------------------------	----------------------------------------------------------------

## Difference between verification and Validation

Validation is the process of checking whether the specification captures the customer's requirements, while verification is the process of checking that the software meets specifications.

## Explain types of Performance testing.

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

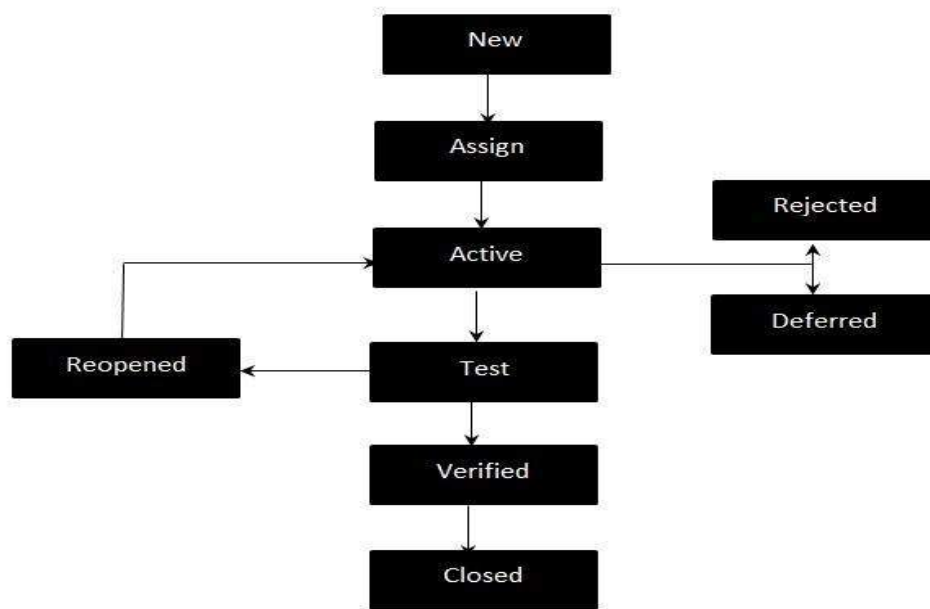
## What is Error, Defect, Bug and failure?

A mistake in coding is called Error, error found by tester is called Defect, defect accepted by development team then it is called Bug, build does not meet the requirements then it is Failure.

## Difference between Priority and Severity

Priority	Severity
Priority is a parameter to decide the order in which defects should be fixed.	Severity is a parameter to denote the impact of a particular defect on the software.
Priority means how fast defect has to be fixed.	Severity means how severe defect is affecting the functionality.
Priority is related to scheduling to resolve the problem.	Severity is related to the quality standard.
Product manager decides the priorities of defects.	Testing engineer decides the severity level of the defect.
Its value is subjective.	Its value is objective.
Its value changes from time to time.	Its value doesn't change from time to time.
Priority is of 3 types: Low, Medium, and High.	Severity is of 5 types: Critical, Major, Moderate, Minor, and Cosmetic.

## What is Bug Life Cycle?



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

SDLC	STLC
What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?	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, more number of members (developers) are required for the whole process.	In STLC, less 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.

## What is the difference between test scenarios, test cases, and test script?

Test Scenario	Test Case	Test Script
Is any functionality that can be tested.	Is a set of actions executed to verify particular features or functionality.	Is a set of instructions to test an app automatically.
Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirement Specification (SRS).	Is mostly derived from test scenarios.	Is mostly derived from test cases.
Helps test the end-to-end functionality in an Agile way.	Helps in exhaustive testing of an app.	Helps to test specific things repeatedly.
Is more focused on what to test.	Is focused on what to test and how to test.	Is focused on the expected result.
Takes less time and fewer resources to create.	Requires more resources and time.	Requires less time for testing but more resources for scripts creating and updating.
Includes an end-to-end functionality to be tested.	Includes test steps, data, expected results for testing.	Includes different commands to develop a script.
The main task is to check the full functionality of a software application.	The main task is to verify compliance with the applicable standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.
Allows quickly assessing the testing scope.	Allows detecting errors and defects.	Allows carrying out an automatic execution of test cases.

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

A Test Plan refers to a detailed document that catalogs the test strategy, objectives, schedule, estimations, deadlines, and the resources required for completing that particular project. Think of it as a blueprint for running the tests needed to ensure the software is working properly – controlled by test managers.

A Test Plan refers to a detailed document that catalogs the test strategy, objectives, schedule, estimations, deadlines, and the resources required for completing that particular project.

## Explain the difference between Authorization and Authentication in Web testing.

### Authentication

Authentication (AuthN) is a process that verifies that someone or something is who they say they are. Technology systems typically use some form of authentication to secure access to an application or its data.

### Authorization



Authorization is the security process that determines a user or service's level of access. In technology, we use authorization to give users or services permission to access some data or perform a particular action.

## What are the common problems faced in Web testing?

- Integration
- Interoperability
- Security
- Performance
- Usability

## What are the different Methodologies in Agile Development Model?

Agile methodology is a “step by step” dynamic focused on short-term visibility but never losing the long-term product goal.

### Scrum

Scrum is, undoubtedly, the most used of the many frameworks underpinning Agile methodology. **Scrum is characterised by cycles or stages of development, known as sprints**, and by the maximisation of development time for a software product towards a goal, the Product Goal. This Product Goal is a larger value objective, in which sprints bring the scrum team product a step closer.

### Kanban

The word **Kanban** is of Japanese origin and its meaning is linked to the concept of “just in time”. In practice, the Kanban method is organised on a board or table (Kanban board), divided into columns, showing every flow within the software production project. As the development evolves, the information contained in the table changes, and whenever a new task comes into play, a new “card” is created.