

## Module–2(Manual Testing)

### 1. What is Exploratory Testing?

Exploratory testing is a concurrent process where Test design, execution and logging happen simultaneously

Exploratory Testing is a type of software testing in which the tester is free to select any possible methodology to test the software.

It is an unscripted approach to software testing.

In exploratory testing, software developers use their learning, knowledge, skills, and abilities to test the software developed by themselves

Testing is often not recorded

### 2. What is traceability matrix?

Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability.

Traceability matrix maps all the requirements with the test cases. By using this document one can verify test cases cover all functionality of the application as per the requirements of the customer.

Types of Traceability Matrix

Forward Traceability – Mapping of Requirements to Test cases

Backward Traceability – Mapping of Test Cases to Requirements

Bi-Directional Traceability - A Good Traceability matrix is the References from test cases to basis documentation and vice versa.

### 3. What is Boundary value testing?

Boundary value testing involves evaluating data based on its boundary values, or its two opposite ends, such as its minimums and maximums.

Boundary value analysis generates test cases that highlight errors better than equivalence partitioning.

The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.

At those points when input values change from valid to invalid errors are most likely to occur.

#### 4. What is Integration testing?

System Integration Testing is testing between the 'System' and 'Acceptance' phases.

The objective of System Integration Testing is to provide confidence that the system or application is able to interoperate successfully with other specified software systems and does not have an adverse effect on other systems that may also be present in the live environment, or vice versa

#### 5. What determines the level of risk?

The risk level is determined by two dimensions: probability and impact.

Probability: it measures the likelihood of event occurring. Expressed as a percentage. It answers the question: "how likely is it to happen"

Impact: risk, by default, brings a negative impact to any project. It assesses the severity of event. It answers the question: "what would be the extent of its effect?"

Risk level calculation:

Probability(20%) x impact(high) = risk level (high)

#### 6. What is Alpha testing?

Alpha Testing is always performed at the time of Acceptance Testing when developers test the product and project to check whether it meets the user requirements or not.

It is considered as the User Acceptance Testing (UAT) which is done at developer's area.

It is always performed by the developers at the software development site.

Sometimes it is also performed by Independent Testing Team.

Alpha Testing is not open to the market and public and It is always performed in Virtual Environment.

#### 7. What is beta testing?

It is always performed by the customers at their own site.

It is not performed by Independent Testing Team.

Beta Testing is always open to the market and public. and performed in Real Time Environment.

It is always performed outside the organization.

It is also the form of Acceptance Testing.

Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.

## 8. What is component(unit) testing?

A unit is the smallest testable part of software.

Component Testing is the testing of individual software components.

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

Unit testing is the first level of testing and is performed prior to Integration Testing.

Component can be tested in isolation – stubs/drivers may be employed.

Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

Unit tests find problems early in the development cycle.

Unit testing is performed by using the White Box Testing method.

## 9. What is functional system testing?

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications.

A Requirement may exist as a text document

There are two types of Test Approach

Requirement Based Functional Testing

Process Based Testing

## 10. What is Non-Functional Testing?

Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability

May be performed at all Test levels

Non-functional testing includes, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.

The term non-functional testing describes the tests required to measure characteristics of systems and software that can be quantified on a varying scale, such as response times for performance testing.

To address this issue, performance testing is carried out to check & fine tune system response times. The goal of performance testing is to reduce response time to an acceptable level

Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

## 11. What is GUI Testing?

Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

Parameters to be checked in GUI testing

Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.

Check you can execute the intended functionality of the application using the GUI

Check Error Messages are displayed correctly

Check for Clear demarcation of different sections on screen.

Check Font used in application is readable.

Check the alignment of the text is proper.

Check the Color of the font and warning messages is aesthetically pleasing

Check that the images have good clarity.

Check that the images are properly aligned.

Check the positioning of GUI elements for different screen resolution.

## 12. What is Adhoc(error guessing) testing?

Adhoc testing is an informal testing type with an aim to break the system.

It does not create test cases altogether.

Testers randomly test the application without any test cases or any business requirement document.

Main aim of this testing is to find defects by random checking.

Adhoc testing can be achieved with the testing technique called Error Guessing.

Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.

The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

It also saves a lot of time because of the assumptions and guessing made by the experienced testers to find out the defects which otherwise won't be able to find.

## 13. What is load testing?

Load testing - Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

This testing usually identifies

The maximum operating capacity of an application

Determine whether current infrastructure is sufficient to run the application

Sustainability of application with respect to peak user load

Number of concurrent users that an application can support, and scalability to allow more users to access it.

It is a type of non-functional testing. Load testing is commonly used for the Client/Server, Web based applications – both Intranet and Internet.

Load testing gives confidence in the system & its reliability and performance.

Load Testing helps identify the bottlenecks in the system under heavy user stress scenarios before they happen in a production environment.

Load testing gives excellent protection against poor performance and accommodates complementary strategies for performance management and monitoring of a production environment.

Load testing will determine whether system needs to be fine-tuned or modification of hardware and software is required to improve performance.

#### 14. What is stress Testing?

System is given stress beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

Stress testing is used to test the stability & reliability of the system.

Verifying the application behavior under customer expected configuration and more than the customer expected load

Involves testing an application under extreme workloads to see how it handles high traffic or data processing.

The objective is to identify breaking point of an application

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

This testing technique takes into account the internal structure of system or component. Complete access to the object's source code is needed for white box-testing. This is known as 'white box' testing because tester gets the internal working of the code.

Unit testing and some part of integration testing fall under white box testing category.

## White box Testing Techniques:

**Statement/segment Coverage:** This ensures at least one execution of each statement in the code

**Decision/branch Coverage:** This relates to testing of all branches of the code in the condition statements. Decision coverage is more effective than statement coverage. 100% decision coverage ensures 100% statement coverage but not vice versa.

## Condition Coverage:

This is closely related to decision coverage but has better sensitivity to the control flow.

Condition coverage reports the true or false outcome of each condition.

Condition coverage measures the conditions independently of each other.

16. What is black box testing? What are the different black box testing techniques?

Testing, either functional or non-functional, without reference to the internal structure of the component or system.

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.

What a system does, rather than HOW it does it.

Typically used at System Test phase

Types of black box testing

Equivalence partitioning

Boundary value analysis

Decision tables

State transition testing

Use-case Testing

17. Mention what bigbang testing is?

In Big Bang integration testing all components or modules are integrated simultaneously, after which everything is tested as a whole.

Big Bang testing has the advantage that everything is finished before integration testing starts.

The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.

Here all components are integrated together at **once**, and then tested.

18. What is the purpose of exit criteria?

Exit criteria is used to determine when testing at any stage is complete. The set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed.

Exit criteria may be defined in terms of:

Thoroughness – i.e. coverage or requirements

cost or time constraints

percentage of tests run without incident

number of faults remaining

19. When should "Regression Testing" be performed?

The purpose of regression testing is to confirm that a recent program or code change has not adversely affected existing features.

Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

When new functionality is added to the system and the code has been modified to absorb and integrate that functionality with the existing code.

When some defect has been identified in the software and the code is debugged to fix it.

When the code is modified to optimize its working.

20. What are 7 key principles? Explain in detail?



## 1. Testing shows presence of Defects

You test software to identify problems so you can fix them before you deploy the software to production environments. However, this process doesn't mean that there aren't any bugs in the product. It just means that there may be bugs, but you didn't find them.

## 2. Exhaustive Testing is Impossible!

The truth is that you can't test everything, i.e., every combination of preconditions and inputs. And if you try to do so you'll waste time and money, but it won't affect the overall quality of the software.

What you need to do is assess risk and plan your tests around these risks so you can be sure you're testing the key functions.

## 3. Early Testing

When it comes to the software development lifecycle, testing early is the key to identifying any defects in the requirements or design phase as soon as possible. It's much easier and less expensive to fix bugs in the early stages of testing than at the end of the software lifecycle

## 4. 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 They are 'clustered'

most operational failures of a system are usually confined to a small number of modules

## 5. The Pesticide Paradox

if you continuously run the same tests, eventually they'll fail to find new defects, even though they'll probably confirm the software is working.

Consequently, you must continue to review your tests as well as add to your scenarios or modify them to help prevent this pesticide paradox.

## 6. Testing is Context Dependent

Testing is done differently in different contexts.

Different kinds of sites are tested differently.

For example, Safety – critical software is tested differently from an e-commerce site.

## 7. Absence of Errors Fallacy

Even after defects have been resolved it may still be unusable if it does not fulfil the users' needs and expectations

## 21. Explain types of Performance testing.

### Load testing

Load testing simulates a real-world load on the system to see how it performs under stress. It helps identify bottlenecks and determine the maximum number of users or transactions the system can handle. It checks the product's ability to perform under anticipated user loads. The objective is to identify performance congestion before the software product is launched in the market.

### 2. Stress testing

Stress testing is a type of load testing that tests the system's ability to handle a high load above normal usage levels. It helps identify the breaking point of the system and any potential issues that may occur under heavy load conditions. It involves testing a product under extreme workloads to see whether it handles high traffic or not. The objective is to identify the breaking point of a software product.

### 3. Spike testing

Spike testing is a type of load testing that tests the system's ability to handle sudden spikes in traffic. It helps identify any issues that may occur when the system is suddenly hit with a high number of requests. It tests the product's reaction to sudden large spikes in the load generated by users.

### 5. Endurance testing

Endurance testing is similar to soak testing, but it focuses on the long-term behavior of the system under a constant load. It is performed to ensure the software can handle the expected load over a long period.

### 6. Volume testing

In Volume testing, a large number of data is saved in a database and the overall software system's behavior is observed. The objective is to check the product's performance under varying database volumes.

### 7. Scalability testing

In Scalability testing, the software application's effectiveness is determined by scaling up to support an increase in user load. It helps in planning capacity additions to your software system.

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

An Error is a mistake made in the code due to which compilation or execution fails,

A Defect is a deviation between the actual and expected output

A bug refers to defects which means

that the software product or the application is not working as per the adhered requirements set

Failure is the accumulation of several defects that ultimately lead to Software failure and results in the loss of information in critical modules thereby making the system unresponsive.

A Fault is a state that causes the software to fail and therefore it does not achieve its necessary function.

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

The test plan provides a detailed understanding of the workflow and functions of the system. It documents how each of the features will be tested to find out if the system works according to its design, find bugs, and determine its actual limitations.

Information about resources, test environment, test limitations and the testing schedule are covered.

24. When to use Usability Testing?

Conducting usability tests before any design decisions are made helps us identify the most important user pain points. By observing how users behave, we can uncover latent needs that people don't articulate during interviews or surveys.

25. What is the procedure for GUI Testing

## MANUAL BASED TESTING

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

## RECORD AND REPLAY

GUI testing can be done using automation tools. This is done in 2 parts. During Record, test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test. Example of such tools - QTP.

## MODEL BASED TESTING

A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements.