

# **Module–2**

## **Manual Testing**

### **1. What is Exploratory Testing?**

- Exploratory testing is a software testing method that involves designing and executing tests on the fly, without using pre-defined test cases. It's a combination of informal and scripted testing, where the tester decides what to test on the moment

### **2. What is Integration testing?**

- Integration testing, also known as I&T (integration and testing), is a software testing phase that verifies how components of a software application work together.

### **3. What is Function 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

### **4. What is Non-Function system testing?**

- Non-Functional Testing is a type of software testing that focuses on evaluating the non-functional aspects of a system, such as its performance, usability, reliability, and security. It complements functional testing by assessing how well the system operates rather than just whether it operates correctly

## **5. What is GUI system testing?**

- Graphical User Interface (GUI) testing is the process of testing the system's GUI. 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
- During GUI Testing Test Engineers validates user interface of the application as following aspects:
  - Look & Feel
  - Easy to use
  - Navigations & shortcut keys
- Gui Objects:
  - Window, Dialog Box, Push Button, Radio Button, Radio Group, Tool bar, Edit Box, Text Box, Check Box, List Box, Drop down Box, Combo Box, Tab, Tree view, progress bar, Table, Scroll bar Etc.

## **6. What is Load testing?**

- Load Testing is a type of performance testing that evaluates a system's behaviour under expected workload conditions. It aims to determine how the system responds when multiple users access it concurrently.

## **7. What is Stress testing?**

- Stress Testing is a non-functional testing technique that evaluates a system's behaviour under extreme or peak load conditions. It's designed to push the system beyond its normal operating limits to identify its breaking point and assess its stability and reliability under stress.

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

- White Box Testing is a software testing method where the tester has access to the internal structure and workings of the application. This knowledge is used to design test cases that can verify the correctness of the software at the code level.
- White box testing is primarily performed by software developers who have a deep understanding of the code and its internal structure.
- **Types of White Box Testing:**
  - I. **Statement Coverage:** Ensures that every line of code is executed at least once during testing.
  - II. **Branch Coverage:** Ensures that every branch of a conditional statement is executed at least once.
  - III. **Path Coverage:** Ensures that every possible path through the code is executed at least once.
  - IV. **Condition Coverage:** Ensures that both true and false conditions of each condition in the code are tested.
  - V. **Decision Coverage:** Ensures that all possible outcomes of each decision point in the code are tested.

## 9. What is black box testing? What are the different black box testing techniques?

- **Black Box Testing** is a software testing method where the tester examines the functionality of an application without peering into its internal structures or workings. It's like treating the software as a "black box" – you know what goes in (input) and what comes out (output), but not how it works inside.
- Black box testing is done by **software testers**, who are professionals specifically trained in testing methodologies. They design and execute test cases based on the software's requirements and specifications.

## 10. Difference between QA v/s QC v/s Tester QA(Quality assurance)

QA(Quality assurance)	QC(Quality Control)	Tester
A broader process that focuses on preventing defects by establishing and implementing processes and standards throughout the entire software development lifecycle.	A narrower process that focuses on identifying and correcting defects in the final product.	A specific role within the QC process responsible for executing tests, analyzing results, and reporting defects.
QA establishes the quality framework.	QC verifies the product against that framework.	Testers are the individuals who perform the testing activities within the QC process.
<b>QA</b> would establish the overall quality framework, such as coding standards, testing methodologies, and risk management processes.	<b>QC</b> would focus on the actual testing activities, such as functional, performance, and security testing, to identify and address defects.	<b>Testers</b> would be the individuals responsible for designing and executing specific test cases, logging bugs, and verifying bug fixes.

## 11. Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Smoke Testing is done to assure that the acute functionalities of program is working fine.	Sanity testing is done to check the bugs have been fixed after the build.
Smoke testing is also called subset of acceptance testing.	Sanity testing is also called subset of regression testing.
Smoke testing is documented	Sanity testing isn't documented.
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.

Smoke testing is used to test all over function of the system/product.

Smoke testing can be performed either manually or by using automation tools.

Smoke testing is performed when new product is built.

Smoke Testing firstly performs on the initial build. Smoke testing is done first.

In the smoke testing process, the software build could be stable or unstable.

For every new build release smoke testing is carried out.

Sanity testing is done to measure the rationality of the system/product by performing testing.

Sanity testing is used in the case of only modified or defect functions of system/products.

Sanity testing is commonly executed manually, not by using any automation approach.

Sanity testing is conducted after the completion of regression testing.

Sanity Testing is done on stable builds or for the introduced new features in the software.

During sanity testing, the software build is comparatively stable.

Sanity testing is carried out when in-depth testing is not possible because of short time.

## **12. Explain types of Performance testing:**

### **1. Load testing**

Load Testing is simulating 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.

### **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.

### **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.

#### **4. Soak testing**

Soak Testing is a type of load testing that tests the system's ability to handle a sustained load over a prolonged period. It helps identify any issues that may occur after prolonged usage of the system.

#### **5. Scalability testing**

In Scalability, 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.

### **13. What is Error, Defect, Bug and failure?**

#### **1. Error**

- **Definition:** A mistake, misconception, or misunderstanding on the part of the developer. It can arise from incorrect assumptions, misinterpretations of requirements, or simply human error during the coding process
- **Example:** Developer made a mistake in coding

#### **2. Defect**

- **Definition:** A deviation from the expected behavior of the software. It's a flaw in the software that causes it to behave incorrectly
- **Example:** When software have to send same output but didn't get output and program stop execute from that point

#### **3. Bug**

- **Definition:** An informal term often used interchangeably with "defect." It generally refers to any unexpected behavior or problem in the software
- **Example:** when user enters wrong password but system log in from that it is bug

#### **4. Failure**

- **Definition:** The inability of the software to perform its intended function. It's the observable consequence of a defect

- **Example:** If a critical bug prevents the software from starting or processing data correctly, it results in a system failure

#### 14. Difference between Priority and Severity:

Features	Severity	Priority
<b>Definition</b>	Severity is a parameter to denote the impact of a particular defect on the software.	Priority is a parameter to decide the order in which defects should be fixed.
<b>Purpose</b>	Severity means how severe the defect is affecting the functionality.	Priority means how fast the defect has to be fixed.
<b>Relation</b>	Severity is related to the quality standard.	Priority is related to scheduling to resolve the problem.
<b>Categories</b>	Severity is divided into 4 categories: <ol style="list-style-type: none"> <li>1. Critical</li> <li>2. Major</li> <li>3. Medium</li> <li>4. Low</li> </ol>	Priority is divided into 3 categories: <ol style="list-style-type: none"> <li>1. Low</li> <li>2. Medium</li> <li>3. High</li> </ol>
<b>Who decides defects?</b>	The testing engineer decides the severity level of the defect.	The product manager decides the priorities of defects.
<b>Value</b>	Its value is objective.	Its value is subjective.
<b>Value change</b>	Its value doesn't change from time to time.	Its value changes from time to time.
<b>Association</b>	It is associated with functionality or standards.	It is associated with scheduling.

<b>Based On</b>	It is based on the technical aspect of the product.	It is based on the customer's requirements.
-----------------	-----------------------------------------------------	---------------------------------------------

### 15. Explain the difference between Functional testing and Non-Functional testing

#### Functional Testing

It verifies the operations and actions of an application.

It is based on requirements of customer.

It helps to enhance the behaviour of the application.

Functional testing is easy to execute manually.

It tests what the product does.

Functional testing is based on the business requirement.

Example: Unit, Smoke, Integration, Regression

#### Non-functional Testing

It verifies the behaviour of an application.

It is based on expectations of customer.

It helps to improve the performance of the application.

It is hard to execute non-functional testing manually.

It describes how the product does.

Non-functional testing is based on the performance requirement.

Example: Load, Stress, Performance, Scalability

### 16. What is priority?

- Priority comes from the word prior, which means to come before something else. A priority is the concern, interest or desire that comes before all others

### 17. What is severity?

- The quality or state of being severe: the condition of being very bad, serious, unpleasant, or harsh. the severity of the climate. the severity



of the punishment. Medication can help shorten the illness and lessen its severity

### 18. Difference between priority and severity

Severity	Priority
Severity is a parameter to denote the impact of a particular defect on the software.	Priority is a parameter to decide the order in which defects should be fixed.
Severity means how severe the defect is affecting the functionality.	Priority means how fast the defect has to be fixed.
Severity is related to the quality standard.	Priority is related to scheduling to resolve the problem.
The testing engineer decides the severity level of the defect.	The product manager decides the priorities of defects.
It is based on the technical aspect of the product.	It is based on the customer's requirements.

### 19. Write a scenario of only WhatsApp chat messages

- Verify that the user can set a chat wallpaper.
- Verify that the user sets privacy settings like turning on/off last seen, online status, read receipts, etc.
- Verify that the user can update notification settings like – notification sound, on/off, and show preview for both group and individual chats.
- Verify that the user can take the complete chat backup of his chats.
- Verify that the user can update the phone number that is used by the WhatsApp application.
- Verify that the user can disable/delete his Whatsapp account.

- Verify that the user can check data usage by images, audio, video, and documents in WhatsApp chats.

## **20. Write a Scenario of Pen**

- Verify the outer body material of the pen. Check if it is metallic, plastic, or any other material specified in the requirement specifications.
- Check the colour of the outer body of the pen. It should be as per the specifications.
- Verify that the brand name and/or logo of the company creating the pen should be clearly visible. • Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen.
- Verify that the user is able to write clearly over different types of papers.
- Verify if the pen is with a cap or without a cap.
- Verify the colour of the ink on the pen.
- Verify that the user is able to refill the pen with all the supported ink types.

## **21. Write a scenario of Microwave Owen:**

- Verify that the oven's material is optimal for its use as an oven and as per the specification.
- Verify that the oven heats the food at the desired temperature properly.
- Verify that the oven heats food at the desired temperature within a specified time duration.
- Verify the ovens functioning with the maximum attainable temperature.
- Verify the ovens functioning with minimum attainable temperature.
- Verify that the oven's plate rotation speed is optimal and not too high to spill the food kept over it.
- Verify that the oven's door gets closed properly.
- Verify that the oven's door opens smoothly.

- Verify that the power cord of the oven is long enough
- Verify that the chair is stable enough to take an average human load.
- Check the material used in making the chair-wood, plastic etc.
- Check if the chair's leg are level to the floor.
- Check the usability of the chair as an office chair, normal household chair.
- Check if there is back support in the chair.
- Check if there is support for hands in the chair.
- Check the height of the chair's seat from floor.
- Verify the type of door of the lift is as per the specification.
- Verify the type of metal used in the lift interior and exterior.
- Verify the capacity of the lift in terms of the total weight.
- Verify the buttons in the lift to close and open the door and numbers as per the number of floors.

## **22. Write a scenario of chair**

- Verify that the chair is stable enough to take an average human load.
- Check the material used in making the chair-wood, plastic etc.
- Check if the chair's leg are level to the floor.
- Check the usability of the chair as an office chair or normal household chair.
- Check if there is back support in the chair.
- Check if there is support for hands.
- Check the height of the chair's seat from floor.

## **23. Write a Scenario of Lift (Elevator)**

- Verify that the lift moves to the particular floor as the button of the floor is clicked.
- Verify that the lift stops when the up/down buttons on a particular floor are pressed.
- Verify if there is an emergency button to contact officials in case of any mishap.
- Verify the performance of the floor – the time taken to go to a floor.
- Verify the time duration for which the door remains open by default.
- Verify if the lift interior is having proper air ventilation.

- Verify lighting and fan in the lift.
- Verify that at no point the lift door should open while in motion
- Verify that in case of power loss, there should be a backup mechanism to safely get into a floor or a backup power supply.
- Verify that in case the multiple floor number button is clicked, the lift should stop on each floor.
- Verify that in case of capacity limit is reached users are prompted with a warning alert- audio/visual.