

# **SOFTWARE TESTING ASSIGNMENT**

## **Module-2(Manual Testing)**

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

- Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits.
- Is not random testing but it is Adhoc testing with purpose of find bugs.
- Is structured and rigorous.
- Is highly teachable and manageable.
- Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently.

### **2. What is 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 analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.
- Boundary value analysis is a method which refines equivalence partitioning.
- Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design.

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

- Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.
- If one value finds a bug, the others probably will too.
- if one doesn't find a bug, the others probably won't either.

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

- Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.
- Integration Testing is a level of the software testing process where individual units are combined and tested as a group.

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

- Risks are of two level of risk
  - Project Risks
  - Product Risk

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

- 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
- It is conducted for the software application and project.
- It is always performed in Virtual Environment
- It comes under the category of both White Box Testing and Black Box Testing.

## **8. 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.
- It is usually conducted for software product.
- It is performed in Real Time Environment.
- It is only a kind of Black Box Testing.

## **9. What is component testing?**

Component(Unit) – A minimal software item that can be tested in isolation.  
It means “A unit is the smallest testable part of software.”

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

Functional System Testing : A requirement that specifies a function that a system or system component must perform

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

Non – functional testing checks the performance reliability, reliability, efficiency, usability, interoperability, maintainability and other non-functional aspect of the software testing.

## **12. 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.

## **13. What is Adhoc testing?**

- Adhoc testing is an informal testing type with an aim to break the system.
- It does not follow any test design techniques to create test cases.
- In fact it does not create test cases altogether!
- This testing is primarily performed if the knowledge of testers in the system under test is very high

## **14. What is load testing?**

Load Testing is to test the system behaviour under normal workload conditions, and it is just testing or simulating with the actual workload.

## **15. What is stress Testing?**

Stress Testing is to test the system behaviour under extreme conditions, and it is carried out till the system failure.

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

White Box Testing: Testing based on an analysis of the internal structure of the component or system.

- The different types of white box testing are:
  1. Statement coverage
  2. Decision coverage
  3. Condition coverage

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

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

- The different types of white box testing are:
  1. Equivalence partitioning
  2. Boundary value analysis
  3. Decision tables
  4. State transition testing
  5. Use-case Testing
  6. Other Black Box Testing
  7. Syntax or Pattern Testing

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

- Database Defects
- Critical Functionality Defects
- Functionality Defects
- Security Defects
- User Interface Defects

## **19. Mention what big bang testing is?**

- In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
- Big Bang testing has the advantage that everything is finished before integration testing starts.

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

The purpose of exit criteria is to define the minimum requirements or conditions that must be met before the testing process can be concluded and the software application can be released, ensuring that the software meets the required quality standards and all necessary testing activities have been completed.

## **21. When should "Regression Testing" be performed?**

Regression Testing: Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in

unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.

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

### **➤ General Testing Principles**

1. Testing shows presence of Defects.
2. Exhaustive Testing is Impossible!
3. Early Testing.
4. Defect Clustering.
5. The Pesticide Paradox.
6. Testing is Context Dependent.
7. Absence of Errors Fallacy.

#### **1. Testing shows presence of Defects.**

Testing can show that defects are present, but cannot prove that there are no defects.

#### **2. Exhaustive Testing is Impossible!**

Testing everything including all combinations of input and preconditions is not possible.

#### **3. 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.

#### **4. Defect Clustering:-**

Small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.

#### **5. 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.

#### **6. Testing is Context Dependent.**

Testing is basically context dependent.

Testing is done differently in different contexts

#### **7. 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.

### 23. Difference between QA v/s QC v/s Tester.

S.N.	Quality Assurance Quality	Quality Control	Testing
1	Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	Activities which ensure the identification of bugs/error/defects in the Software.
2	Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
3	Process oriented activities.	Product oriented activities	Product oriented activities.
4	Preventive activities.	It is a corrective process.	It is a preventive process.
5	It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

## 24. Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine	Sanity Testing is done to check the new functionality / bugs have been fixed
The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing	The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing
This testing is performed by the developers or testers	Sanity testing is usually performed by testers
Smoke testing is usually documented or scripted	Sanity testing is usually not documented and is unscripted
Smoke testing is a subset of Regression testing	Sanity testing is a subset of Acceptance testing
Smoke testing exercises the entire system from end to end	Sanity testing exercises only the particular component of the entire system
Smoke testing is like General Health Check Up	Sanity Testing is like specialized health check up

## 25. Difference between verification and Validation.

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product full fills its intended use when placed in its intended environment.
Question	Are we building the product right?	Are we building the right product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
Activities	Reviews <ul style="list-style-type: none"><li>• Walkthroughs</li><li>• Inspections</li></ul>	• Testing

## 26. Explain types of Performance testing.

### ➤ Types of Performance Testing

1. Load testing
2. Stress testing
3. Endurance testing
4. Spike testing
5. Volume testing
6. Scalability testing

### 1. Load testing

Load testing is a type of testing which involves evaluating the performance of the system under the expected workload.



## **2. Stress testing**

Stress testing is a type of performance testing where we evaluate the applications performance at load much higher than expected load.

## **3. Endurance testing**

Endurance testing is also known as „soak testing“. It is done to determine if the system can sustain the continuous expected load for a long duration.

## **4. Spike testing**

In spike testing, we analyse the behaviour of the system on suddenly increasing the number of users.

## **5. Volume Testing**

The volume testing performed by feeding the application with a high volume of data.

## **6. Scalability testing**

The objective of scalability testing is to determine the software applications effectiveness in “scaling up” to support an increase in user load.

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

**Error:-** a human action that produces an incorrect result.

**Defect:-** A flaw in a component or system that can cause the component or system to be fail to perform its required function

**failure:-** Deviation of the component or system from its expected delivery, service or result

**Bug :-** A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.

## **28. Difference between Priority and Severity.**

**Severity:-** Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

**Priority:-** Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements

## 29. What is Bug Life Cycle?

“A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program’s source code or its design.”

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

Functional Testing	Non-Functional Testing
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first.	Non functional testing should be performed after functional testing.
Manual testing or automation tools can be used for functional testing.	Using tools will be effective for this testing.
Business requirements are the inputs to functional testing.	Performance parameters like speed , scalability are inputs to non-functional testing.
Functional testing describes what the product does.	Non-functional testing describes how good the product works.
Easy to do manual testing.	Tough to do manual testing.
Types of Functional testing are <ul style="list-style-type: none"><li>• Unit Testing</li><li>• Smoke Testing</li><li>• Sanity Testing</li><li>• Integration Testing</li><li>• White box testing</li><li>• Black Box testing</li><li>• User Acceptance testing</li><li>• Regression Testing</li></ul>	Types of Non-functional testing are <ul style="list-style-type: none"><li>• Performance Testing</li><li>• Load Testing</li><li>• Volume Testing</li><li>• Stress Testing</li><li>• Security Testing</li><li>• Installation Testing</li><li>• Penetration Testing</li><li>• Compatibility Testing</li><li>• Migration Testing</li></ul>

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

#### **Test Scenarios:**

- Test scenarios are high-level descriptions of a feature or functionality to be tested.
- Test scenarios are less detailed than test cases and do not go into specific steps or expected results.
- They focus on the big picture of what needs to be tested.
- Test scenarios may cover multiple test cases or even multiple features.
- Test scenarios are created during the requirements analysis phase of the software development life cycle.
- Test scenarios are useful for communicating testing requirements to stakeholders who may not be technical.

#### **Test Cases:**

- Test cases are detailed descriptions of a specific condition or action that verifies a requirement or functionality.
- Test cases are more detailed than test scenarios and include specific input data, expected output, and steps to be performed during testing.
- Test cases are typically focused on a single requirement or functionality.
- Test cases are created during the testing phase of the software development life cycle.
- Test cases are primarily used by testers to execute and validate the system.

#### **Test Scripts:**

- Test scripts are code that automates the execution of test cases or sets of test cases.
- Test scripts are even more detailed than test cases and include specific code to perform actions and verify expected results.
- Test scripts can be used to execute a single test case or a series of test cases.
- Test scripts are created during the testing phase of the software development life cycle.
- Test scripts are primarily used by testers or developers to automate the testing process.

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

A test plan is a document that outlines the entire software testing process, including the testing objectives, scope, approach, resources, schedule, risks, and acceptance criteria. It serves as a roadmap for the testing team and ensures that all aspects of the testing process are covered.

### **33. What is priority?**

Priority is the measure of the importance or urgency assigned to a software issue, indicating how critical it is to the functioning of the software and how quickly it needs to be addressed.

### **34. What is severity?**

Severity is the measure of the impact or seriousness of a software defect on the software's functionality, performance, or user experience.

### **35. Bug categories are...**

- Functional defects
- Performance defects
- Usability defects
- Compatibility defects
- Security defects
- Installation/Configuration defects
- Documentation defects.

### **36. Advantage of Bugzilla.**

Bugzilla is a web-based bug tracking tool that helps software development teams to manage and track bugs efficiently. Some advantages of Bugzilla include easy bug reporting, efficient bug tracking and management, collaboration and communication between team members, customizable workflows, and integration with other tools.

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

There are several methodologies in Agile Development Model, including:

1. **Scrum:** An iterative and incremental framework for managing and completing complex projects.

2. **Kanban:** A visual method for managing work that emphasizes just-in-time delivery and continuous improvement.
3. **Lean:** A methodology that focuses on maximizing customer value while minimizing waste and improving efficiency.
4. **Extreme Programming (XP):** A set of software development practices that emphasizes frequent releases, testing, and customer involvement.
5. **Crystal:** A flexible methodology that adapts to the needs of the project and the team.
6. **Feature-Driven Development (FDD):** A methodology that breaks down the project into small, feature-sized chunks and emphasizes iterative development and testing.
7. **Dynamic Systems Development Method (DSDM):** A framework for delivering projects on time and within budget while maintaining quality.
8. **Adaptive Software Development (ASD):** A methodology that emphasizes collaboration, communication, and feedback to adapt to changing requirements and circumstances.

**38. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?**

<b>Authorization</b>	<b>Authentication</b>
Authorization determines what resources a user can access after authentication	Authentication is the process of verifying the identity of a user or system.
It is the process of verifying whether a user has permission to access the requested resource or not.	It is the process of confirming whether a user is genuine or not.
It involves validating the user's role or privilege and checking if it matches the resource requirements.	It involves the use of credentials like username and password, biometric information, or digital certificates.
It is mainly concerned with granting or denying user access to specific resources based on their role or privilege level.	It is the first line of defense against unauthorized access to a system or resource.
Authorization ensures that only authorized users can access sensitive data and functionalities.	It is the first line of defense against unauthorized access to a system or resource.

## **Common problems faced in Web Testing:**

- ❖ Cross-browser compatibility issues: Web applications need to be tested across different browsers, devices, and operating systems, which can result in compatibility issues that affect the user experience.
- ❖ Security vulnerabilities: Web applications are susceptible to attacks such as cross-site scripting (XSS), SQL injection, and cross-site request forgery (CSRF), which can compromise user data and application functionality.
- ❖ Performance issues: Web applications need to perform well under heavy traffic and load, and may suffer from slow page loading, slow server response times, and other performance issues.
- ❖ Functional defects: Web applications may have defects related to user interface, functionality, and data processing that affect the user experience and application usability.
- ❖ Integration issues: Web applications often need to integrate with other systems and technologies, such as payment gateways, third-party APIs, and databases, which can result in integration issues that need to be tested and resolved.