

Module-2 Manual Testing Assignment

1. What is Exploratory Testing?

2. Exploratory testing is about learning the product, designing and executing tests, and interpreting the test results, all at the same time. It is a plan-as-you-test approach.
3. In contrast to formal testing (e.g. manual testing), where the purpose is to monitor known risks, the purpose of exploratory testing is to analyze potential risks. This means that exploratory testing starts with a focus on the things you do not know, and the illusions you are holding true without any empirical evidence. The ultimate goal is to explore the unknown to avoid the unexpected happening.
4. In exploratory testing, your next test is always influenced by the result of the last test you did. This implies that you become more exploratory the more you can't tell what test should be run in advance of the test cycle. Finally, exploratory testing is not so much a thing you do, it is much more a way you think.

2. What is traceability matrix?

→ A traceability matrix is a document that details the technical requirements for a given test scenario and its current state. It helps the testing team understand the level of testing that is done for a given product. The traceability process itself is used to review the test cases that were defined for any requirement.

3. 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.

4 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

Module-2 Manual Testing Assignment

5. What is integration testing?

- Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

6. What determines the level of risk?

- As Risk is determined by a combination of Probability and Severity, the main area of the Matrix reveals the Risk Levels. The levels are Low, Medium, High, and Extremely High. To have a low level of risk, we must have a somewhat limited probability and level of severity.

7. What is Alpha testing?

- Alpha testing is the initial phase of validating whether a new product will perform as expected. Alpha tests are carried out early in the development process by internal staff and are followed up with beta tests, in which a sampling of the intended audience actually tries the product out.

8. What is beta Testing ?

- 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.

9. What is component (Unit)Testing?

- 10. Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements. Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.
- 11. Unit tests are automated and are run each time the code is changed to ensure that new code does not break existing functionality. Unit tests are designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system. This allows developers to quickly identify and

Module-2 Manual Testing Assignment

fix any issues early in the development process, improving the overall quality of the software and reducing the time required for later testing.

10. What is functional system testing?

→ Functional testing is basically defined as a type of testing that verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned with the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output, and comparing the actual output with the expected output. This testing focuses on checking the user interface, APIs, database, security, client or server application, and functionality of the Application Under Test. Functional testing can be manual or automated.

11 What is Non-Functional Testing?

- **Non-Functional Testing** is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.
- An excellent example of non-functional test would be to check how many people can simultaneously login into a software.

12 What is GUI Testing?

GUI is the abbreviation of 'Graphical User Interface'. It contains several visual elements, such as buttons, text boxes, menus, checkboxes, images, etc. GUI testing refers to the validating UI functions or features of an application that are visible to the users, and they should comply with business requirements. GUI testing is also known as [UI testing](#). That means 'User Interface testing'. So, you can use both acronyms alternatively.

But why do we need GUI testing? GUI testing aims to ensure that the end-user gets a hassle-free experience.

13 What is Adhoc Testing?

- **Ad hoc Testing** is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly

Module-2 Manual Testing Assignment

and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

14 What is Load Testing?

- In this section, we are going to understand **load testing**, which is the important part of **Performance testing** and used to check the performance of the software by applying some load.

And we also learn about its process, why we need to perform the load testing, the objective of load testing, example, various strategies of load Testing, advantage and disadvantage.

15 what is Stress Testing?

- **Stress Testing** is a software testing technique that determines the robustness of software by testing beyond the limits of normal operation. Stress testing is particularly important for critical software but is used for all types of software. Stress testing emphasizes robustness, availability, and error handling under a heavy load rather than what is correct behavior under normal situations. Stress testing is defined as a type of software testing that verifies the stability and reliability of the system. This test particularly determines the system on its robustness and error handling under extremely heavy load conditions. It even tests beyond the normal operating point and analyses how the system works under extreme conditions. Stress testing is performed to ensure that the system would not crash under crunch situations. Stress testing is also known as **Endurance Testing** or **Torture Testing**

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

Types of White Box Testing

- Path Testing. Path Testing is a white-box testing approach based on a program's control structure. ...
 - Loop Testing. ...
 - Conditional Testing. ...
 - Unit Testing. ...
 - Mutation Testing. ...
 - Integration Testing. ...
 - Penetration Testing. ...
 - Testing based on Memory Perspective
-
- White box testing is a technique that uses a program's internal or source code to design different test cases to check the quality of the program. In this technique, the

Module-2 Manual Testing Assignment

internal structure and implementation of how an application works are known to the tester.

17 What is black box testing? What are the different black box testing techniques?

- ➔ Black box testing is a method for writing test cases that is frequently utilized. It may be beneficial for condensing a large number of potential inputs into a smaller number of more effective ones. It is accomplished by categorizing inputs into classes and assigning a value to each class.

18 Mention what are the categories of defects?•

- ➔ 3 Types of defects every importer needs to know. Quality control professionals typically classify quality defects into three main categories: minor, major and critical. The nature and severity of a defect determines in which of the three categories it belongs.

19 Mention what bigbang testing is?

- ➔ What is Big-Bang Testing? Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system.

20 What is the purpose of exit criteria?

- ➔ Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

21 When should "Regression Testing" be performed?

- ➔ Regression testing is necessary after any feature (or application) enhancement, bug fix, or configuration changes. For example, when developers add a new widget to an application. As more regressions are found in software products, companies are moving towards test automation to perform regression tests.

22 What is 7 key principles? Explain in detail?

1) Exhaustive testing is not possible

Yes! Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application.

Module-2 Manual Testing Assignment

And the million dollar question is, how do you determine this risk?

To answer this let's do an exercise

In your opinion, Which operation is most likely to cause your Operating system to fail?

I am sure most of you would have guessed, Opening 10 different application all at the same time.

So if you were testing this Operating system, you would realize that defects are likely to be found in multi-tasking activity and need to be tested thoroughly which brings us to our next principle [Defect](#) Clustering

2) Defect Clustering

Defect Clustering which states that a small number of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.

By experience, you can identify such risky modules. But this approach has its own problems

If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

3) Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide. Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

To overcome this, the test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects.

Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective. But even after all this sweat & hard work in testing, you can never claim your

Module-2 Manual Testing Assignment

product is bug-free. To drive home this point, let's see this video of the public launch of Windows 98

You think a company like MICROSOFT would not have tested their OS thoroughly & would risk their reputation just to see their OS crashing during its public launch!

4) Testing shows a presence of defects

Hence, testing principle states that – Testing talks about the presence of defects and don't talk about the absence of defects. i.e. Software 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.

But what if, you work extra hard, taking all precautions & make your software product 99% bug-free. And the software does not meet the needs & requirements of the clients.

This leads us to our next principle, which states that- Absence of Error

5) Absence of Error – fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement.

Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user's needs & requirements.

To solve this problem, the next principle of testing states that Early Testing

6) Early Testing

Early Testing – Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

Module-2 Manual Testing Assignment

7) Testing is context dependent

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed software's are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.

22 Difference between QA v/s QC v/s Tester

#	Quality Assurance	Quality Control
1	Aims to prevent defects	Aims to identify and fix defects
2	Is a preventive technique	Is a corrective technique
3	Defines standards and procedures that need to be adhered to in order to meet customer requirements	Ensures that standards are followed while working on the product
4	Helps build processes	Helps implement the existing processes
5	Activities are determined before production work begins and performed while the product is being developed	Activities are performed after the product is developed
6	Is a managerial tool	Is a corrective tool
7	It is the duty of the complete project team	It is only the duty of the testing team
8	Comes under the category of verification.	Comes under the category of validation
9	Is a process oriented exercise	Is a product oriented exercise
10	Prevents the occurrence of issues, bugs or defects in the application	Detects, corrects, and reports the bugs or defects in the application
11	Does not involve executing the program or code	Involves executing the program or code
12	Done before Quality Control	Done only after Quality Assurance
13	Human-based checking of documents or files.	Computer-based execution of program or code
14	Is generally not a time-consuming activity	Is generally a time-consuming activity
15	Makes sure quality team is doing the right things in the right way	Makes sure that whatever the quality team has done is as per the requirement



23 Difference between Smoke and Sanity?

Module-2 Manual Testing Assignment

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

24 Difference between verification and Validation?

- **Verification** is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is static testing.
Verification means **Are we building the product right?**
- **Validation** is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product. Validation is the dynamic testing.
Validation means **Are we building the right product?**

25 Explain types of Performance testing?

- Load Testing. Load testing measures system performance as the workload increases.
...
- Stress Testing. ...
- Spike Testing. ...

Module-2 Manual Testing Assignment

- Endurance Testing. ...
- Scalability Testing. ...
- Volume Testing. ...
- Identify the Testing Environment. ...
- Identify Performance Metrics.

26 What is Error, Defect, Bug and failure?

- In software testing, a bug is the informal name of defects, which means that software or application is not working as per the requirement. When we have some coding error, it leads a program to its breakdown, which is known as **a bug**. The **test engineers** use the terminology **Bug**.
- When the application is not working as per the requirement is known as **defects**. It is specified as the aberration from the **actual and expected result** of the application or software
- The Problem in code leads to errors, which means that a mistake can occur due to the developer's coding error as the developer misunderstood the requirement or the requirement was not defined correctly. The **developers** use the term **error**.

27 Difference between Priority and Severity?

Module-2 Manual Testing Assignment

Parameters	Severity in Testing	Priority in Testing
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameter	Severity is basically a parameter that denotes the total impact of a given defect on any software.	Priority is basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Change of Value	The value of Severity changes continually from time to time.	The value of Priority changes from time to time.
Who Decides the Defect	The testing engineer basically decides a defect's severity level.	The product manager basically decides a defect's priority level.
Types	There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.	There are 3 types of Priorities: High, Medium, and Low.

28 Explain the difference between Functional testing and NonFunctional testing?

Module-2 Manual Testing Assignment

Functional Testing	Non-functional Testing
It verifies the operations and actions of an application.	It verifies the behavior of an application.
It is based on requirements of customer.	It is based on expectations of customer.
It helps to enhance the behavior of the application.	It helps to improve the performance of the 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.	Non-functional testing is based on the performance requirement.



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

Module-2 Manual Testing Assignment

SDLC	STLC
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, 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.

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

Module-2 Manual Testing Assignment

Test Case	Test Script
A test case is a high-level document with instructions on the specific functionality of the software product to be tested.	Test Script is a step-by-step instruction to test each software product's functionality (test case).
A test case is the software development life cycle's 'What to test' component.	Test script is the software development life cycle's 'How to test' component.
Test cases are written in simple English.	Test scripts are written in programming languages like VB Script, Python, Java, etc.
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.	Test Case serves as an outline for writing test scripts.



31 What is priority?•

- ➔ What is Priority in Testing? One can define Priority as a parameter for deciding the order in which one can fix the defect. In this, the defect with a higher priority first needs to get fixed. Priority basically defines the order in which one would resolve any given defect.

32 What is Severity?

- ➔ What is Severity in Testing? One can define Severity as the extent to which any given defect can affect/ impact a particular software. Severity is basically a parameter that denotes the impact of any defect and its implication on a software's functionality.

33 Difference between priority and severity?

What is Priority in Testing? One can define Priority as a parameter for deciding the order in which one can fix the defect. In this, the defect with a

Module-2 Manual Testing Assignment

higher priority first needs to get fixed. Priority basically defines the order in which one would resolve any given defect.

What is Severity in Testing? One can define Severity as the extent to which any given defect can affect/ impact a particular software. Severity is basically a parameter that denotes the impact of any defect and its implication on a software's functionality.

34 What are the different Methodologies in Agile Development Model?•

- ➔ 1) Kanban. ...
- ➔ 2) Scrum. ...
- ➔ 3) Extreme Programming (XP) ...
- ➔ 4) Crystal. ...
- ➔ 5) Dynamic Systems Development Method (DSDM) ...
- ➔ 6) Feature-Driven Development (FDD)