

Module 2 : Manual Testing

Que 1 : What is Software Testing ?

Ans : There are 3 different definition of software testing .

- Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not .
- Testing is executing a system in order to identify any gaps , errors or missing requirements in contrary to actual desire or requirements .
- Software testing is process used to identify the correctness , completeness and quality of developed computer software .

Que 2 : What is Exploratory Testing ?

Ans : Exploratory Testing is an approach to software testing that is often described as simultaneous learning , test design , and execution.

It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of the tests .

Que 3 : What is Traceability Matrix ?

Ans : Traceability Matrix is a table which is used to test the requirements during the software development life cycle .

It can be used for forward tracing or backward .

Each requirements in the RTM document is linked with its associated test case , so that testing can be done as per the mentioned requirements

Que 4 : What is Boundary Value Testing ?

Ans : Boundary value testing/analysis is a methodology for designing test cases that concentrates software testing efforts on cases near the limit of valid ranges .

- Boundary value analysis is a method which refines equivalence partitioning .
- Boundary value analysis generates test cases that highest error better than equivalence partitioning .
- The trick is to concentrate software testing efforts in the extreme ends of the equivalence classes .
- At those points when input value changes from valid to invalid error are most likely to occur .
- Boundary value analysis uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design .

Que 5 : What is Equivalence Partitioning Testing ?

Ans : Equivalence partitioning testing is the process of defining the optimum number of tests by :

- Reviewing documents such as the functional design specification and detailed design specification , and identifying each input condition within a function .
- Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition .

Que 6 : What is Integration Testing ?

Ans : Integration testing is a level of the software testing process where individual units are combined and tested as a group .

The purpose of this level of testing is to expose faults in the interaction between integrated units .

Integration testing tests integration or interfaces between components , interactions to different parts of the system such as opening systems , file systems and hardware or interfaces between systems .

There are 2 types of integration testing .

1.Component Integration testing

2.System integration testing

Que 7 : What determines the level of risk ?

Ans :

Que 8 : What is Alpha Testing ?

Ans : Alpha testing is testing method that performed by the developers at the development site.

Alpha testing is always performs at the time of the acceptance testing When developers test the product and the project to check whether it meets the user requirements or not.

Que 9 : What is Beta Testing ?

Ans : Beta testing is always performed at time when the software product and project are marketed .

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

It is performed by independent testing team .

It is performed in **Real Time Environment** .

Que 10 : What is Component Testing ?

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

Unit testing is first level of testing and it is performs prior to the integration testing .

Unit testing is typically written and rung by software developers to ensure that code meets its design and behaves as intended with debugging tool .

Que 11 : What is Functional System Testing ?

Ans : Functional System Testing is a requirement that specify a function that system or system component must perform .

Que 12 : What is Non-Functional Testing ?

Ans : Non-Functional testing is testing of those requirements that do not relate to the functionality .

Que 13 : What is GUI Testing ?

Ans : GUI Testing Means Graphic User Interface Testing is the process of testing the GUI of the system under test .

GUI testing involves checking the screens with the control like menus , buttons , icons , and all type of bars – tool bar , menu bar, dialogue boxes and windows .

Que 14 : What is Adhoc Testing ?

Ans : Adhoc Testing is an informal type of testing type with an aim to break the system . It does nor follow any test design techniques to create test cases . In fact it does not create test cases together .

This testing is primarily performed if the knowledge of the tester in the system under the test is very high .

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

Que 15 : What is White Box Testing and list the type of White box testing ?

Ans : White box testing is based on the analysis of the internal structure of the component or system .

White box testing is also known as a **Structure Based Testing OR Glass Box testing OR Open Box Testing** , because here Tester requires the knowledge of how the **software is implemented , how it works** .

Type of White Box Testing

1. Control Flow Testing
2. Data Flow Twsting
3. Branch Testing
4. Statement Coverage
5. Decision Coverage
6. Prime Path Testing
7. Path Testing etc.

Que 16 : What is Black Box Testing and list the type of Black box testing ?

Ans : Black Box testing , either functional or non-functional , without reference to the structure of the system .

Black Box testing is also known as a **Specification Based Testing** .

The tester have no knowledge of how the system or component structured inside the box . In Black box testing the tester id only concentrating on what the software does , not how it does it .

Types of Black Box Testing

1. Equivalence Partitioning
2. Boundary value Analysis

3. Decision Table
4. State Transition Testing
5. Use-Case Testing
6. Syntax OR Pattern Testing

Que 17 : Mention What are the categories of defects .

Ans : In Software Testing, Types of Severity of bug/defect can be categorized into four parts :

1. Critical: This defect indicates complete shut-down of the process, nothing can proceed further
2. Major: It is a highly severe defect and collapses the system. However, certain parts of the system remain functional
3. Medium: It causes some undesirable behavior, but the system is still functional
4. Low: It won't cause any major break-down of the system

Que 18 : Mention What Big Bang Testing is ?

Ans : Big Bang Testing is an approach of integration testing where integration of all or major components of the system are tested . The big bang method is very effective for saving time in the integration testing process . It helps amateur developers to find integration related defects earlier than actual integration testing .

Que 19 : What is purpose of the Exit Criteria ?

Ans : 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

Exit Criteria typically measures:

- Thoroughness measures, such as coverage of requirements or of code or risk coverage
- Estimates of defect density or reliability measures. (e.g. how many defects open by category)
- Cost.
- Residual Risks, such as defects not fixed or lack of test coverage in certain areas.
- Schedules - such as those based on time to market.

Que 20 : When should "Regression Testing" be performed ?

Ans : Regression testing is done when the 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.

Regression testing should be carried out:

- when the system is stable and the system or the environment changes
- when testing bug-fix releases as part of the maintenance phase
- It should be applied at all Test Levels

- It should be considered complete when agreed completion criteria for regression testing have been met
- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation.

Que 21 : What is 7 key principles? Explain in detail?

Ans : 7 Key Principle :

1. Testing shows presence of defects
2. Exhaustive testing is impossible
3. Early Testing
4. Defect Clustering
5. The Pesticide Peradox
6. Testing is Context Dependent
7. Absence of Error Fallacy

1. Testing shows presence of defects :

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

2. 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.

- We have learned that we cannot test everything

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.
- 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 .

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' .
- In other words, most defects found during testing are usually confined to a small number of modules .
- Similarly, most operational failures of a system are usually confined to a small number of modules.

5. 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.

6. 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.
- If we build a system and, in doing so, find and fix defects....
- It doesn't make it a good system
- Even after defects have been resolved it may still be unusable .

7. 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. Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing .
- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software .
- 1 to 3 failures per KLOC typical for industrial software.
- 0.01 failures per KLOC for NASA Shuttle code.
- Also different industries impose different testing standards.

Que 22 : Difference between QA v/s QC v/s Tester?

Ans :

No	Quality Assurance	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.

Que 23 : Difference between n Smoke and Sanity Testing ?

Ans :

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

Que 24 : Difference between verification and Validation ?

Ans :

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 fulfills 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	<ul style="list-style-type: none">➤ Reviews➤ Walkthroughs➤ Inspections	<ul style="list-style-type: none">➤ Testing

Que 25 : Explain types of Performance testing.

Ans : Types as below

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

Que 26: What is Error, Defect, Bug and failure?

Ans :

1. A mistake in coding is called error, error found by tester is called defect.
2. Defect accepted by development team then it is called bug,
3. Build does not meet the requirements then it is failure.

- Error: A discrepancy between a computed, observed, or measured value OR condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.
- Failure: The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.
- 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.

- Fault: An incorrect step, process, or data definition in a computer program which causes the program to perform in an unintended or unanticipated manner. See: bug, defect, error, exception.
- Defect: Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

Que 27: Difference between Priority and Severity

Ans :

Que 288 : What is Bug Life Cycle?

Ans :

Que 29 : Explain the difference between Functional testing and Non-Functional Testing ?

Ans :

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	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
Easy to do manual testing	Tough to do manual testing
Types of Functional testing are : <ul style="list-style-type: none">➤ Unit Testing➤ Smoke Testing➤ Sanity Testing➤ Integration Testing➤ White box testing➤ Black Box testing➤ User Acceptance testing➤ Regression Testing	Types of Nonfunctional testing are : <ul style="list-style-type: none">➤ Performance Testing➤ Load Testing➤ Volume Testing➤ Stress Testing➤ Security Testing➤ Installation Testing➤ Penetration Testing➤ Compatibility Testing➤ Migration Testing

Que 30 : To create HLR & TestCase of Insta and Facebook First Page

Que 31 : What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle) ?

Ans :

NO	SDLC	STLC
1.	Software Development Life Cycle	Software Testing Life Cycle
2.	SDLC is a Development Methodology,	whereas STLC is a Testing Methodology
3.	Multiple different phases are combined to form SDLC	Whereas Multiple testing phases or practices combine to form STLC.
4.	SDLC covers the entire software development cycles	STLC covers the entire testing cycles
5.	SDLC is a linear process that ensures you design and construct the proper system	STLC is a technique that allows you to test what you've developed thoroughly.

Que 32 : What is the difference between Test Scenarios, Test Cases, and Test Script?

Ans :

Test Scenarios	Test Cases	Test Script
A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.	Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.	The Test Procedures Specification specifies the sequence of actions for a test, i.e. one or more Test Cases
Test Scenario is 'What to be tested'	Test Case is 'How to be tested'	The Test Script can be manual or automated
The scenarios are derived from use cases.	Test cases are derived (or written) from test scenario.	

Que33: Explain what Test Plan is? What is the information that should be Covered ?

Ans :

- It is a high level document in which how to perform testing is described. The Test Plan document is usually prepared by the Test Lead or Test Manager and the focus of the document is to describe what to test, how to test, when to test and who will do what test.

➤ A test plan will include the following.

1. Introduction to the Test Plan document
2. Assumptions when testing the application

3. List of test cases included in Testing the application
4. List of features to be tested
5. What sort of Approach to use when testing the software
6. List of Deliverables that need to be tested
7. The resources allocated for testing the application
8. Any Risks involved during the testing process
9. A Schedule of tasks and milestones as testing is started

Que34 : What are the different Methodologies in Agile Development Model?

Ans :

1. Iteration
2. Warm up
3. Construction
4. Release End game
5. Production
6. Retirement

Que35 : Explain the difference between Authorization and Authentication in Web testing.?

Ans :

Que36 : What are the common problems faced in Web testing?

Ans :