

(1) What is Exploratory Testing?

- ~ In this testing, first application under the test is explored, flow of application is understood, then Test cases are designed and test cases are executed.
- ~ This testing is performed when a requirement is missing.
- ~ This is also an informal testing process.
- ~ It involves test design and control and notes are taken.
- ~ Progress is also tracked.
- ~ It helps to study the product and argument the document and also research the bug.

(2) What is a traceability matrix?

- ~ Test conditions should be able to be linked back to their sources in the test basis, this is known as Traceability.
- ~ Traceability can be horizontal through all the test documentation for a given test level or it can be vertical through the layers of development documentation.
- ~ A software process should help you keep the virtual table up-to-date.
- ~ Traceability Matrix (also known as Requirement Traceability Matrix - RTM) is a table which is used to trace the requirements during the Software development life Cycle. It can be used for forward tracing (i.e. from Requirements to Design or Coding) or backward (i.e. from Coding to Requirements). There are many user defined templates for RTM.

(3) What is Boundary value testing?

- ~ This is one of the software testing technique in which the test cases are designed to include values at the boundary.
- ~ If the input data is used within the boundary value limits, then it is said to be Positive Testing.
- ~ If the input data is picked outside the boundary value limits, then it is said to be Negative Testing.
- ~ A system can accept the numbers from 0 to 10 numeric values. All other numbers are invalid values. Under this technique, boundary values 0 , 10 and -10 will be tested.

(4) What is Equivalence partitioning testing?

- ~ This is a software testing technique which divides the input data into many partitions.
- ~ Values from each partition must be tested at least once. Partitions with valid values are used for Positive Testing.
- ~ While, partitions with invalid values are used for negative testing.
- ~ Numeric values Zero to ten can be divided into two (or three) partitions. In our case we have two partitions -10 to -1 and 0 to 10. Sample values (5 and -5) can be taken from each part to test the scenario.

(5) What is 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.
- ~ There are 2 levels of Integration Testing
 - ~ Component Integration Testing
 - ~ System Integration Testing

(6) What determines the level of risk?

- ~ A factor that could result in future negative consequences; usually expressed as impact and likelihood.
- ~ A Risk could be any future event with a negative consequence .You need to identify the risks associated with your project.
- ~ Risks are of two types
 - ~ Project Risk
 - ~ Product Risk
- ~ Example of Project risk is a Senior Team Member leaving the project abruptly.
- ~ Example of product risks would be Flight Reservation system not installed in test environment.

(7) 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 always performed by the developers at the software development site.
- ~ Sometimes it is also performed by the Independent Testing Team.
- ~ Alpha Testing is not open to the market and public.
- ~ It is conducted for software applications and projects.
- ~ It is always performed in a Virtual Environment.
- ~ It is always performed within the organisation.
- ~ It is a form of Acceptance Testing.

(8) What is beta testing?

- ~ Final testing before releasing an application for commercial purpose. It is typically done by end-users or others.
- ~ It is always performed by the customers at their own site.
- ~ It is not performed by the Independent Testing Team.
- ~ Beta Testing is always open to the market and public.
- ~ It is usually conducted for software products.
- ~ It is performed in a Real Time Environment.
- ~ It is also the form of Acceptance Testing.
- ~ It is only a kind of Black Box Testing.
- ~ Beta Testing is always performed at the time when software products and projects are Marketed.
- ~ Beta testing can be considered “pre-release” testing.

(9) What is component testing?

- ~ A minimal software item that can be tested in isolation. It means A unit is the smallest testable part of software.
- ~ The testing of individual software components.
- ~ Unit Testing is a level of the software testing process where individual units/components 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.
- ~ Sometimes known as Unit Testing, Module Testing or Program Testing.
- ~ Unit tests find problems early in the development cycle.
- ~ Unit testing is performed by using the White Box Testing method.

(10) What is functional system testing?

- ~ A requirement that specifies a function that a system or system components must perform.
- ~ There is two types of techniques
 - (a) Requirement Based Functional Testing
 - ~ Test procedures and cases derived from:
 - ~ detailed user requirements
 - ~ system requirements functional specification
 - ~ User documentation/instructions
 - ~ high level System design
 - (b) Process Based Testing
 - ~ Test procedures and cases derived from:
 - ~ Expected user profiles
 - ~ Business scenarios
 - ~ Use cases

(11) 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 (not just Non Functional Systems Testing).
- ~ It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.

(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 – toolbar, 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.
- ~ Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of the application.
- ~ 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” is the most likely source of errors.
- ~ Error guessing is a technique where experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

(14) What is load testing?

- ~ 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.

(15) What is stress testing?

- ~ Stress testing is to test the system behaviour under extreme conditions and is carried out till the system failure.
- ~ Stress testing determines the breaking point of the system to reveal the maximum point after which it breaks.

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

- ~ White box testing is testing based on an analysis of the internal structure of the component or system.
- ~ Structure-based testing technique is also known as 'white-box' or 'glass-box' testing technique because here the testers require knowledge of how the software is implemented, how it works.
- ~ List the types of white box testing
 - ~ Unit Testing
 - ~ Static and Dynamic Analysis
 - ~ Statement
 - ~ Branch
 - ~ Path Coverage
 - ~ Security Testing
 - ~ Loop and Conditional Testing
 - ~ Mutation and Integration Testing
 - ~ Penetration Testing
 - ~ Memory Perspective Testing

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

- ~ A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality. It is performed by QA teams.
- ~ Specification-based testing technique is also known as 'black-box' or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
- ~ The testers have no knowledge of how the system or component is structured inside the box.
- ~ There are four specification-based or black-box technique:
 - ~ Equivalence partitioning
 - ~ Boundary value analysis
 - ~ Decision tables
 - ~ State transition testing

(18) 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.

(19) What is the purpose of exit criteria?

- ~ 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 (e.g. hand over from System Test to UAT)

(20) When should "Regression Testing" be performed?

- ~ Change in requirements and code is modified according to the requirement
- ~ New feature is added to the software
- ~ Defect fixing
- ~ Performance issue fix

(21) What are 7 key principles? Explain in detail?

~ **Testing shows presence of Defects:**

- ~ Testing can show that defects are present, but cannot prove that there are no defects.
- ~ 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.
- ~ We test to find Faults

~ **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.

~ **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.

~ **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.

~ **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.
- ~ 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.
- ~ Testing identifies bugs, and programmers respond to fix them.
- ~ As bugs are eliminated by the programmers, the software improves.

~ **Testing is Context Dependent:**

- ~ Testing is basically 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!

~ **Absence of Errors Fallacy:**

- ~ If the system built is unusable and does not fulfil the user's needs and expectations then Finding and fixing defects does not help.
- ~ Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations.

(22) Difference between QA v/s QC v/s Tester?

| S.N. | Quality Assurance | Quality Control | Tester |
|------|---|--|---|
| 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 intent to identify bugs/defects through implementation of procedures and processes. | 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. |

(23) Difference between Smoke and Sanity?

| S.N. | Smoke Testing | Sanity Testing |
|------|--|---|
| 1. | Smoke Testing is performed to ascertain that the critical functionalities of the program are working fine. | Sanity Testing is done to check the new functionality / bugs have been fixed. |
| 2. | 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. |
| 3. | This testing is performed by the developers or testers. | Sanity testing is usually performed by testers. |
| 4. | Smoke testing is usually documented or scripted. | Sanity testing is usually not documented and is unscripted. |

| | | |
|----|--|--|
| 5. | Smoke testing is a subset of Regression testing. | Sanity testing is a subset of Acceptance testing. |
| 6. | Smoke testing exercises the entire system from end to end. | Sanity testing exercises only the particular component of the entire system. |
| 7. | Smoke testing is like a General Health check up. | Sanity Testing is like a specialised health check up. |

(24) Difference between verification and Validation?

| S.N. | Verification | Validation |
|------|--|--|
| 1. | 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. |
| 2. | 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 fulfils its intended use when placed in its intended environment. |
| 3. | Are we building the product right? | Are we building the right product? |
| 4. | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| 5. | Reviews, Walkthroughs, Inspections | Testing |

(25) Explain types of Performance testing?

- ~ Various types of performance testing is follows:
- ~ **Load testing:**
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.
- ~ **Stress testing:**
Stress testing is to test the system behaviour under extreme conditions and is carried out till the system failure.
- ~ **Volume testing:**
Testing which confirms that any values that may become large over time (such as accumulated counts, logs, and data files), can be accommodated by the program and will not cause the program to stop working or degrade its operation in any manner. It is usually conducted by the performance engineer.
- ~ **Scalability testing:**
Part of the battery of non-functional tests which tests a software application for measuring its capability to scale up - be it the user load supported, the number of transactions, the data volume etc. It is conducted by the performance engineer.

(26) What is Error, Defect, Bug and failure?

- ~ A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is a failure.

(27) Explain the difference between Functional testing and NonFunctional testing?

| S.N. | Functional Testing | Non-Functional Testing |
|------|--|---|
| 1. | 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. |
| 2. | Functional testing is executed first. | Non functional testing should be performed after functional testing. |
| 3. | Manual testing or automation tools can be used for functional testing. | Using tools will be effective for this testing. |
| 4. | Business requirements are the inputs to functional testing. | Performance parameters like speed, scalability are inputs to non-functional testing. |
| 5. | Functional testing describes what the product does. | Nonfunctional testing describes how good the product works. |
| 6. | Easy to do manual testing. | Tough to do manual testing. |
| 7. | Types of Functional testing are: Unit Testing Smoke Testing Sanity Testing Integration Testing White box testing Black Box testing | Types of Nonfunctional testing are: Performance Testing Load Testing Volume Testing Stress Testing Security Testing Penetration Testing Compatibility Testing Migration Testing |

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

| S.N. | SDLC | STLC |
|------|--|---|
| 1. | SDLC is mainly related to software development. | STLC is mainly related to software testing. |
| 2. | Besides development other phases like testing is also included. | It focuses only on testing the software. |
| 3. | SDLC involves a total of six phases or steps. | STLC involves only five phases or steps. |
| 4. | In SDLC, more members (developers) are required for the whole process. | In STLC, less number of members (testers) are needed. |

| | | |
|-----|--|---|
| 5. | In SDLC, the development team makes the plans and designs based on the requirements. | In STLC, the testing team(Test Lead or Test Architect) makes the plans and designs. |
| 6. | Goal of SDLC is to complete successful development of software. | Goal of STLC is to complete successful testing of software. |
| 7. | It helps in developing good quality software. | It helps in making the software defects free. |
| 8. | SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |
| 9. | Post deployment support , enhancement , and update are to be included if necessary. | Regression tests are run by the QA team to check deployed maintenance code and maintain test cases and automated scripts. |
| 10. | Creation of reusable software systems is the end result of SDLC. | A tested software system is the end result of STLC. |

(29) What is the difference between test scenarios, test cases, and test scripts?

- ~ A test scenario is any functionality that a software testing company can examine. It is also called a Test Condition or Test Possibility. A test case is a document that lists the steps a QA engineer needs to execute. A test script is a short program written in a programming language.

(30) Explain what the Test Plan is? What is the information that should be covered?

- ~ A test plan is essentially a project management plan, which should always include the major components of a normal plan for project management purposes, including an overview, scope, methodology, resources, schedule, risks, and communication plan.
- ~ This includes defining test objectives, test approach, test tools, test environment, test schedules and team responsibilities and composition. However, before the right test approach and other planning details can be defined, a larger view of the organisational and project objectives must be defined first.

(31) What are the different Methodologies in Agile Development Model?

- ~ There are various methodologies present in agile testing and those are listed below:

- ~ **Scrum**

- ~ **eXtreme Programming**

Below listed methodologies are used less frequently

- ~ **Dynamic System Development Method (DSDM)**

This is an Iterative and incremental approach that emphasises on the continuous user involvement.

- ~ **Test Driven Development (TDD)**

This is a technique which has short iterations where new test cases covering the desired improvement or new functionality are written first.

- ~ **Feature Driven Development**

This is an iterative and incremental software development process and this can aim depending on the features.

- ~ **XBreed**

Agile enterprise previously known as Xbreed .It is an agile way of managing, architecting and monitoring the enterprise.

~ **Crystal**

Crystal is an adaptive technique mainly used for software development methodologies.

(32) When to use Usability Testing?

- ~ Aesthetics and design are important. How well a product looks usually determines how well It works.
- ~ There are many software applications / websites, which miserably fail, once launched, due to following reasons –
 - ~ Where do I click next?
 - ~ Which page needs to be navigated?
 - ~ Which Icon or Jargon represents what?
 - ~ Error messages are not consistent or effectively displayed
 - ~ Session time is not sufficient.
- ~ Usability Testing identifies usability errors in the system early in development cycle and can save a product from failure.

(33) What is the procedure for 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.

