

# Manual Testing

## Interview Question and Answer Freshers

### Q1 What is Software Testing?

Software testing is a process of executing a program or application with the intent of finding a software bug.

### Q1.1 What do you understand by software testing?

Software testing is a validation process that confirms that a system works as per the business requirements. It qualifies a system on various aspects such as usability, accuracy, completeness, efficiency, etc. ANSI/IEEE 1059 is the global standard that defines the basic principles of testing.

### Q2 Why Software Testing is necessary?

Software Testing is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong, Humans make mistakes all the time.

- Software testing is really required to point out the **defects** and **errors** that were made during the **development phases**.
- It's essential since it makes sure that the customer finds the organization reliable and their satisfaction in the application is maintained.
- It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.
- Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
- Testing is required for an effective performance of software application or product.

- It's important to ensure that the application should not result into any **failures** because it can be very expensive in the future or in the later stages of the development.
- It's required to stay in business.

### **Q3 Software Testing objectives and purpose?**

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

### **Q4 What are the two main categories of software testing?**

Software testing is a huge domain but it can be broadly categorized into two areas such as :

- **Manual Testing** – This is the oldest type of software testing where the testers manually execute test cases without using any test automation tools. It means the software application is tested manually by QA testers.
- **Automation Testing** – This is the process of using the assistance of tools, scripts, and software to perform test cases by repeating pre-defined actions. Test Automation focuses on replacing manual human activity with systems or devices that enhance efficiency.

### **Q5 Why do we say testing is context dependent?**

Testing is very much dependent upon context, as the test 'subject' will require a specific type of approach; for example, an application designed for use in the cruise industry will be quite different to one in the insurance industry.

## **Q6 What different types of manual testing are there?**

Different types of manual testing are;

- Black Box Testing
- White Box Testing
- Unit Testing
- System Testing
- Integration Testing
- Acceptance Testing

## **Q7 What is Black Box Testing?**

**Black box testing** involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

**Black box testing** is a powerful testing technique because it exercises a system end-to-end. Just like end-users “don’t care” how a system is coded or architected, and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Along the way, a black box test evaluates all relevant subsystems, including UI/UX, web server or application server, database, dependencies, and integrated systems.

## **Q8 What is White Box Testing?**

**White Box Testing** is a testing technique in which software’s internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

## **Q9 What is Unit Testing?**

**Unit testing** involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.

A unit is a single testable part of a software system and tested during the development phase of the application software.

The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.

Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as **Unit testing** or **components testing**.

## **Q11 What is Integration Testing?**

**Integration Testing** is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

Integration Testing focuses on checking data communication amongst these modules.

## **Q12 What is Acceptance Testing?**

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience.

## **Q13 What are the Black Box Testing Techniques?**

- Boundary Value Analysis
- Equivalence partitioning
- State Transition Testing
- Decision Table Testing
- Graph-Based Testing
- Error Guessing Technique

## **Q14 What is Functional testing?**

**FUNCTIONAL TESTING** is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

## **Q15 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 a non-functional test would be to check how many people can simultaneously login into a software.

## **Q16 What is regression testing?**

Regression testing is a **software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements**. Regression testing is responsible for the overall stability and functionality of the existing features.

## **Q17 What is Smoke and Sanity Testing?**

- :- 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.

## **Q18 What is UI Testing?**

**UI testing** is a testing type that helps testers ensure that all the fields, labels, buttons, and other items on the screen function as desired. It involves checking screens with controls, like toolbars, colors, fonts, sizes, icons, and others, and how these respond to the user input.

## **Q19 What is Responsive Testing?**

**Responsive testing** involves how a website or web application looks and behaves on different devices, screen sizes, and resolutions. The goal of responsive testing is to ensure that the website or web application can be used effectively on various devices, including desktops, laptops, tablets, and smartphones.

## **Q20 What is Cross Browser Testing?**

**Cross Browser testing** is a type of non-functional testing that lets you check whether your website works as intended when accessed through: Different Browser-OS combinations i.e., on popular browsers like Firefox, Chrome, Edge, Safari—on any of the popular operating systems like Windows, macOS, iOS and Android.

## **Q21 What's the difference between verification and validation in testing?**

**:- Verification** is a process of determining if the software is designed and developed as per the specified requirements.

**:- Validation** is the process of checking if the software (end product) has met the client's true needs and expectations

## **Q22 What is the difference between a bug, a defect and an error?**

**Error** :- We can say that a mistake made by a programmer during coding is called an error.

**Defect** :- an error found during the testing in the development phase is called a defect.(Deviation from requirement)

**Bug**:- an error found during the testing phase is called a bug.

## **Q23 What is the test case and test scenario?**

- The **test case** is an in-detailed document that includes **all possible inputs** such as **positive and negative**, and **the navigation steps**. These are implemented during the testing process to check whether the software application is performing the task for which it was developed or not.
- **Test scenario** It is a detailed document of test cases that cover **end to end functionality** of a software application in liner statements. The test scenario is a high-level classification of testable requirements. Before performing the test scenario, the test engineer needs to consider the test cases for each scenario.

## Q24 What are the phases involved in the Software Testing Life Cycle?

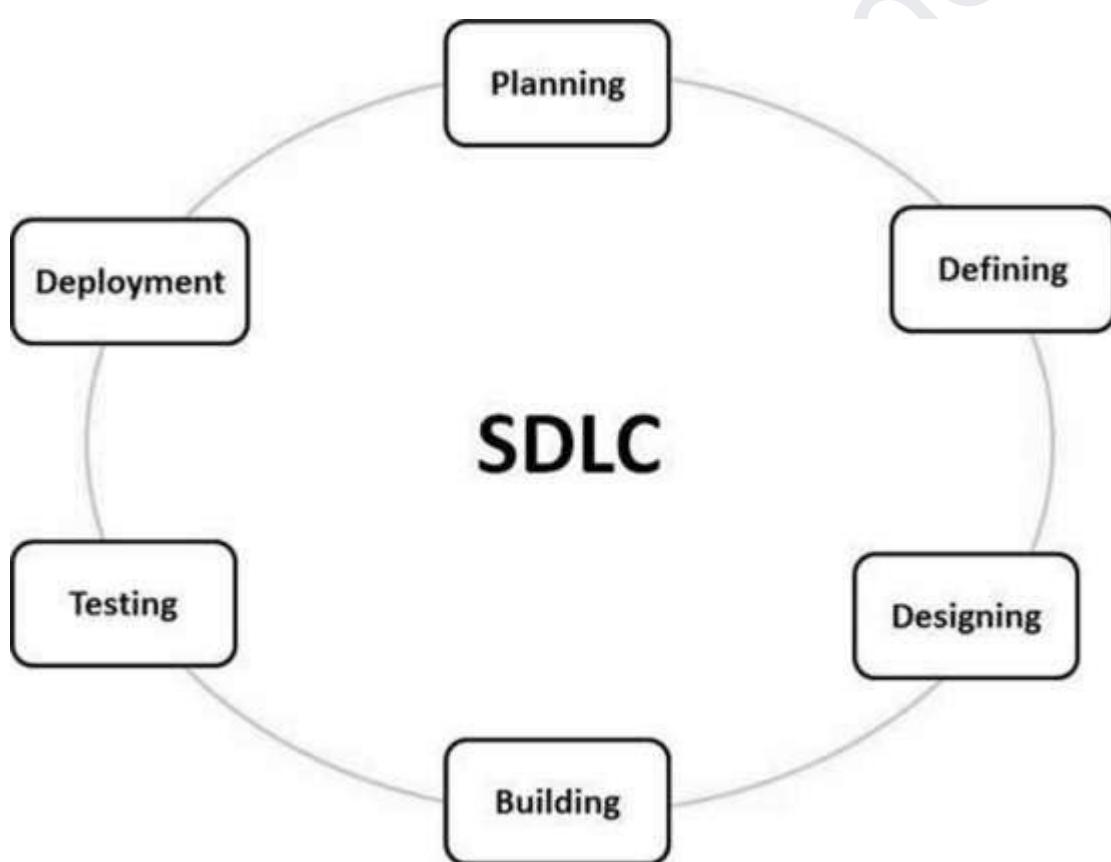
The different phases involved in the software testing life cycle are:

Phases	Explanation
Requirement Analysis	QA team understands the requirement in terms of what we will testing & figure out the testable requirements.
Test Planning	In this phase, the test strategy is defined. Objective & the scope of the project is determined.
Test Case Development	Here, detailed test cases are defined and developed. The testing team also prepares the test data for testing.
Test Environment Setup	It is a setup of software and hardware for the testing teams to execute test cases.
Test Execution	It is the process of executing the code and comparing the expected and actual results.
Test Cycle Closure	It involves calling out the testing team member meeting & evaluating cycle completion criteria based on test coverage, quality, cost, time, critical business objectives, and software.

## Q25 What is SDLC?

**SDLC** is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages –

## **Stage 1: Planning and Requirement Analysis**

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

## **Stage 2: Defining Requirements**

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

## **Stage 3: Designing the Product Architecture**

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

## **Stage 4: Building or Developing the Product**

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

### **Stage 5: Testing the Product**

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

### **Stage 6: Deployment in the Market and Maintenance**

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

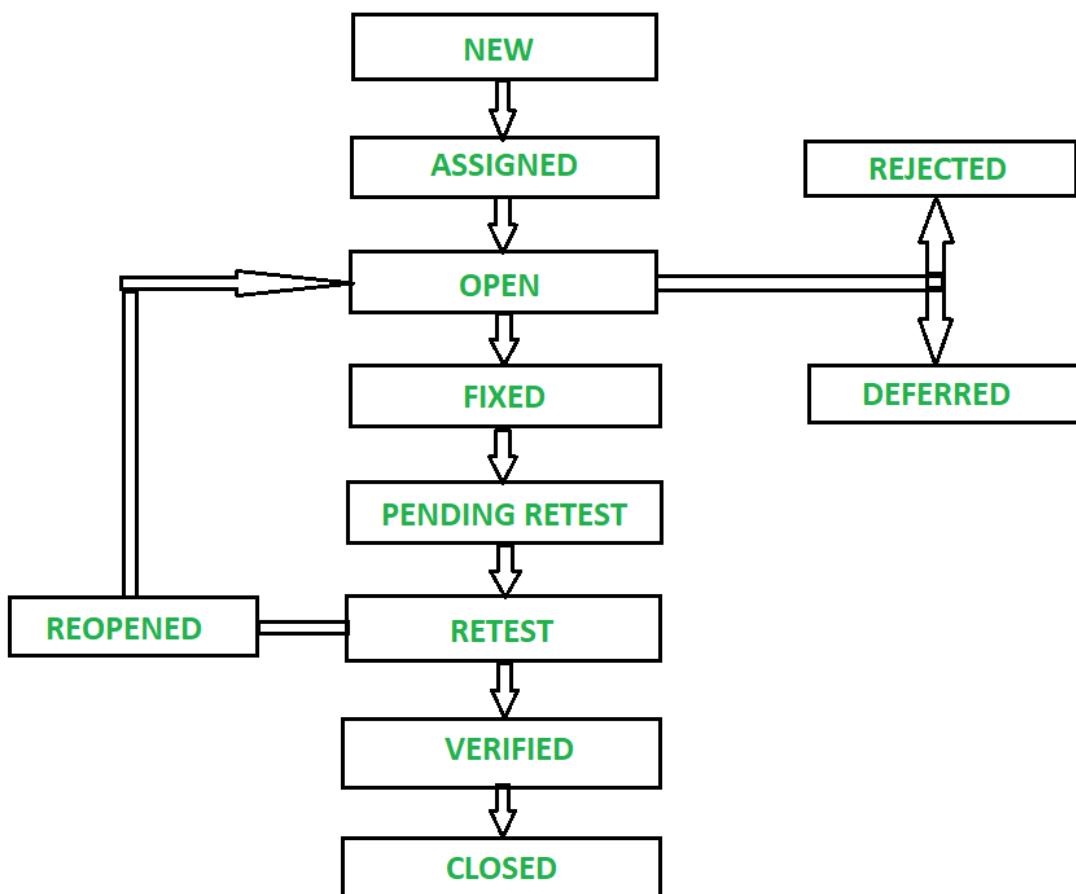
Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

## **Q26 Bug Life Cycle?**

- **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team.
- **Open:** The developer starts analyzing and works on the defect fix.
- **Fixed:** When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
- **Pending retest:** Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software

testing remains pending from the testers end, the status assigned is “pending retest.”

- **Retest:** Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”



- **Verified:** The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
- **Reopen:** If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.

- **Closed:** If the bug is no longer exists then tester assigns the status “Closed.”
- **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”
- **Rejected:** If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
- **Deferred:** If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs.
- **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

## Q27 What is the difference between Severity and Priority?

**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.

## Q28 What are the 7 principles of testing?

### The seven principles of testing

- **Testing shows the presence of defects, not their absence.**
- **Exhaustive testing is impossible.**
- **Early testing saves time and money.**
- **Defects cluster together.**
- **Beware of the pesticide paradox.**
- **Testing is context dependent.**
- **Absence-of-errors is a fallacy.**

## Q29 Test Cases of Pen?

Prerequisite for performing all these test cases is, a Pen must have a refill and the refill must have ink.

1. The grip of the pen: Verify if you are able to hold the pen comfortably.
2. Writing: Verify if you are able to write smoothly.
3. Verify that the pen is not making any sound while writing.
4. Verify the ink flow. It should not overflow nor get a break either.
5. Verify the quality of the material used for the pen.
6. Verify if the company or pen name is visible clearly.
7. Verify if the pen color or text written on the pen is not getting removed easily.
8. Verify, whether the width of the line drawn by the pen is as per the expectations or not.
9. Verify the ink color, it should be consistent from the start till the end.
10. Verify if a pen can write on a variety of papers like smooth, rough, thick, thin, glossy etc.
11. Verify for the waterproof ink. [Not for gel and ink pens].
12. Verify if the ink will not get dried easily by keeping the pen open for some time. [Not for ink pen]
13. Verify if any other refill fits in the pen or not.
14. Verify that the pen doesn't have sharp edges or corners.
15. Verify if the ink and external assembly of the pen is made of non-toxic material.

## **Negative Test Cases/Scenarios**

1. Put the pen in water and then try to write. Verify if you are able to write with this pen. The pen can get wet because of the water spill on the table or during the rainy season. It can be due to any reason.
2. Drop the pen from some height (Table height) in the upside-down position. Verify if you are able to write with this pen. By mistake, the pen can any time fall on the ground. So testing this possibility to know its impact, will help us in knowing the quality of the pen.

For both the above test cases, the frequency that these scenarios will happen may not be very high but it is not even very low. So by knowing its impact, we will be able to know more about the quality of the pen.

## **Performance Test Cases/Scenarios**

1. Verify how fast you can write with this pen.
2. Verify if the pen will perform the same even though you use it continuously for hours.
3. Verify 'How much can be written in one refill/ink sac?'
4. Verify if the tip or nib of the pen is not destroyed after continuous writing for hours.

## **Out of Scope Test Cases**

1. Verifying the pen working in different gravity.
2. Verifying the pen working at different temperatures (Especially at 0 degrees).

## **Q30 Test Cases 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, normal household chair.
- Check if there is back support in the chair.
- Check if there is support for hands in the chair.
- Verify the paint's type and color.
- Verify if the chair's material is brittle or not.
- Check if cushion is provided with chair or not.
- Check the condition when washed with water or the effect of water on the chair.
- Verify that the dimension of chair is as per the specifications.
- Verify that the weight of the chair is as per the specifications.
- Check the height of the chair's seat from floor.

## **Q31 Test Cases of Login Functionality?**

- Verify that as soon as the login page opens, by default the cursor should remain on the username textbox.
- Verify that the user is able to navigate or access the different controls by pressing the 'Tab' key on the keyboard.
- Check if the password is in masked form when typed in the password field.
- Check if the password can be copy-pasted or not.
- Verify that the user is able to login by entering valid credentials and clicking on the 'Login' button.
- Verify that the user is able to login by entering valid credentials and pressing Enter key.

- Check that the user is not able to login with an invalid username and password.
- Verify that the validation message gets displayed in case the user leaves the username or password field blank.
- Check that the validation message is displayed in case the user exceeds the character limit of the user name and password fields.
- Verify that the reset button functionality is on the login page. Clicking on it should clear the textbox's content.
- Verify if there is a checkbox with the label “remember password” on the login page.
- Verify that closing the browser should not log out an authenticated user. Launching the application should lead the user to the login state only.

## Security Test Cases for Login Page

- Verify that there is a limit on the total number of unsuccessful login attempts. So that a user cannot use a brute-force mechanism to try all possible combinations of username-password.
- Verify that in case of incorrect credentials, a message like “incorrect username or password” should get displayed. Instead of an exact message pointing to the incorrect field. This is because a message like “incorrect password” will help a hacker in knowing that the username is correct. In this way, he will just need to try a different combination on the password field only.
- Verify the login session timeout duration. So, once logged in a user cannot be authenticated for a lifetime.
- Verify that once logged in, clicking the back button doesn't log out the user.

- Verify if SQL Injection attacks work on the login page. The application should not be vulnerable to SQL injection attacks.
- Verify that the XSS vulnerability should not work on the login page.

## 1 To 3 Years Of Experience

### **Q1 What is Quality Assurance and what are the different activities involved in Quality assurance?**

Quality assurance is a process-driven approach that checks if the process of developing the product is correct and conforming to all the standards. It is considered a preventive measure. This is because it identifies the weakness in the process to build software. It involves activities like document review, test case review, walk-throughs, inspection.

### **Q2 What is Quality Control and what are the different types of testing involved in QC?**

Quality control is a product-driven approach that checks that the developed product conforms to all the specified requirements. It is considered a corrective measure as it tests the built product to find the defects. It involves different types of testing like functional testing, performance testing, usability testing, etc.

### **Q3 What are the different types of testing?**

Ans. Testing can broadly be defined into two types-

- **Functional testing** – [Functional testing](#) involves validating the functional specifications of the system.
- **Non Functional testing** – [Non-functional testing](#) is a type of testing that involves testing of non-functional requirements of the system such as performance, scalability, security, endurance, portability, etc.

the way the testing is done, it can be categorized as-

- **Black-box testing** – In black-box testing, the tester need not have any knowledge of the internal architecture or implementation of the system. The tester interacts with the system through the interface providing input and validating the received output.
- **White box testing** – In white box testing, the tester analyses the internal architecture of the system as well as the quality of source code on different parameters like code optimization, code coverage, reusability, etc.
- **Gray box testing** – In gray box testing, the tester has partial access to the internal architecture of the system e.g. the tester may have access to the design documents or database structure. This information helps the tester to test the application better.

#### **Q4 Is there any difference between retesting and regression testing?**

Possible differences between retesting and regression testing are as follows:

- We perform retesting to verify the defect fixes. But, the regression testing assures that the bug fix does not break other parts of the application.
- Regression test cases verify the functionality of some or all modules.
- Regression testing ensures the re-execution of passed test cases. Whereas, retesting involves the execution of test cases that are in a failed state.
- Retesting has a higher priority over regression. But in some cases, both get executed in parallel.

## **Q5 As per your understanding, list down the key challenges of software testing.**

Following are some of the key challenges of software testing:

- The lack of availability of standard documents to understand the application
- Lack of skilled testers
- Understanding the requirements: Testers require good listening and understanding capabilities to be able to communicate with the customers the application requirements.
- The decision-making ability to analyze when to stop testing
- Ability to work under time constraints
- Ability to decide which tests to execute first
- Testing the entire application using an optimized number of test cases



## **Q6 What is a test plan?**

Ans. A test plan is a formal document describing the scope of testing, the approach to be used, the resources required, and the time estimate for carrying out the testing process. It is derived from the requirement documents (Software Requirement Specifications).

## **Q7 What is a test scenario?**

Ans. A test scenario is derived from a use case. It is used for end-to-end testing of a feature of an application. A single test scenario can cater to multiple test cases. Scenario testing is particularly useful when there is a time constraint while testing.

## **Q8 What is a Test case?**

Ans. A test case is used to test the conformance of an application with its requirement specifications. It is a set of conditions with pre-requisites, input values, and expected results in a documented form.

## **Q9 What are some attributes of a test case?**

- Ans. A test case can have the following attributes-
  - TestCaseId – A unique identifier of the test case.
  - Test Summary – One-liner summary of the test case.
  - Description – Detailed description of the test case.
  - Prerequisite or pre-condition – A set of prerequisites that must be followed before executing the test steps.
  - Test Steps – Detailed steps for performing the test case.
  - Expected result – The expected result in order to pass the test.
  - Actual result – The actual result after executing the test steps.
  - Test Result – Pass/Fail status of the test execution.
  - Automation Status – Identifier of automation – whether the application is automated or not.
  - Date – The test execution date.
  - Executed by – Name of the person executing the test case.

## **Q10 What are some defect reporting attributes?**

Ans. Some of the attributes of a Defect report are-

- DefectId – A unique identifier of the defect.
- Defect Summary – A one-line summary of the defect, more like a defect title.
- Defect Description – A detailed description of the defect.
- Steps to reproduce – The steps to reproduce the defect.
- Expected Result – The expected behavior from which the application is deviating because of the defect.
- Actual Result- The current erroneous state of the application w.r.t. the defect.
- Defect Severity – Based on the criticality of the defect, this field can be set to minor, medium, major, or show stopper.
- Priority – Based on the urgency of the defect, this field can be set on a scale of P0 to P3.

## **Q11 What are some of the bug or defect management tools?**

Ans. Some of the most widely used Defect Management tools are – Jira, Bugzilla, Redmine, Mantis, Quality Center, etc.

## **Q12 What is Test data?**

Ans. Test data is data that is used to test the software with different inputs and helps to check whether the corresponding output is as per the expected result or not. This data is created based on the business requirements.

## **Q13 What is defect priority?**

Ans. A defect priority is an urgency of fixing the defect. Normally the defect priority is set on a scale of P0 to P3 with the P0 defect having the most urgency to fix.

## **Q14 What is defect severity?**

Ans. Defect severity is the severity of the defect impacting the functionality. Based on the organization, we can have different levels of defect severity ranging from minor to critical or show stopper.

## **Q15 Explain the procedure for manual testing?**

The manual testing process comprises the following steps:

- Planning and Control
- Analysis and Design
- Implementation and Execution
- Evaluating exit criteria and Reporting
- Test Closure activities.

## **Q16 What is a blocker?**

Ans. A blocker is a bug of high priority and high severity. It prevents or blocks testing of some other major portion of the application as well.

## **Q17 What is a critical bug?**

Ans. A critical bug is a bug that impacts a major functionality of the application and the application cannot be delivered without fixing the bug. It is different from the blocker bug as it doesn't affect or block the testing of other parts of the application.

## **Q18 What is Static Testing?**

Ans. Static testing is a kind of testing for reviewing the work products or documentation that are being created throughout the entire project. It allows for reviewing the specifications, business requirements, documentation, processes, and functional requirements in the initial phase of testing. So that the testers involved in it can understand the requirements in more detail before starting the testing lifecycle which intends to help in delivering a quality product.

## **Q19 What is Dynamic Testing?**

Ans. The type of testing performed by executing or running the application under test either manually or using automation is called dynamic testing.

## **Q20 Explain the different types of specification-based test design techniques.**

Ans. Specification-based test design techniques are also referred to as black-box testing. It involves testing based on the specification of the system under test without knowing its internal architecture. The different types of specification-based test design or black box testing techniques are-

- **Equivalence partitioning** – Grouping test data into logical groups or equivalence classes with the assumption that all the data items lying in the classes will have the same effect on the application.
- **Boundary value analysis** – Testing using the boundary values of the equivalence classes taken as the test input.
- **Decision tables** – Testing using decision tables showing the application's behavior based on a different combination of input values.
- **Cause-effect graph** – Testing using a graphical representation of the result or outcome and all the factors that affect the outcome.
- **State transition testing** – Testing based on the state machine model.
- **Use case testing** – Testing carried out using use cases.

## **Q21 Explain equivalence class partitioning.**

Ans. Equivalence class partitioning is a specification-based black-box testing technique. In equivalence class partitioning, a set of input data that defines different test conditions are partitioned into logically similar groups such that using even a single test data from the group for testing can be considered similar to using all the other data in that group.

For example, for testing a Square program (a program that prints the square of a number), the equivalence classes can be-

Set of Negative numbers, whole numbers, decimal numbers, sets of large numbers, etc.

## **Q22 What is boundary value analysis?**

Ans. Boundary value analysis is a software testing technique for designing test cases wherein the boundary values of the classes of the equivalence class partitioning are taken as input to the test cases e.g. if the test data lies in the range of 0-100, the boundary value analysis will include test data – 0,1, 99, 100

## **Q23 What is decision table testing?**

Ans. Decision table testing is a type of specification-based test design technique or black-box testing technique in which testing is carried out using decision tables showing the application's behavior based on different combinations of input values.

Decision tables are particularly helpful in designing test cases for complex business scenarios involving the verification of applications with multiple combinations of input.

## **Q24 What is a cause-effect graph?**

Ans. A cause-effect graph testing is a black-box test design technique in which a graphical representation of input i.e. cause and output i.e. effect is used for test designing. This technique uses different notations representing AND, OR, NOT, etc relations between the input conditions leading to output.

## **Q25 What is state transition testing?**

Ans. State transition testing is a black box test design technique based on a state machine model. State transition testing is based on the concept that a system can be defined as a collection of multiple states and the transition from one state to another happens because of some event.

## **Q26 What is use case testing?**

Ans. Use case testing is a black-box testing approach in which testing is carried out using use cases. A use-case scenario is seen as an interaction between the application and actors(users). These use cases are used for depicting requirements and hence can also serve as a basis for acceptance testing.

## **Q27 What is Test Coverage?**

Ans. It is a metric that measures the amount of testing performed on software while executing the test cases. Test coverage for any software can be calculated as the percentage of the number of test areas or coverage items covered with respect to the total number of test areas.

The higher the test coverage, the more the part of the software gets covered by test cases and hence, the more effective the testing will be.

## **Q28 List the different types of severity.**

The criticality of a bug can be low, medium, or high depending on the context.

- User interface defects – Low
- Boundary-related defects – Medium
- Error handling defects – Medium
- Calculation defects – High
- Misinterpreted data – High
- Hardware failures – High
- Compatibility issues – High
- Control flow defects – High
- Load conditions – High

## **Q29 What is code coverage?**

Ans. Code coverage is the measure of the amount of code covered by the test scripts. It gives the idea of the part of the application covered by the test suite.

## **Q30 What are the different types of functional testing?**

Functional testing covers the following types of validation techniques:

- Unit testing
- Smoke testing
- UAT
- Sanity testing
- Interface testing
- Integration testing
- System testing
- Regression testing

## **Q31 What are the different levels of testing?**

Ans. Testing can be performed at different levels during the development process. Performing testing activities at multiple levels helps in the early identification of bugs. The different levels of testing are –

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

## **Q32 What is unit testing?**

Ans. Unit testing is the first level of testing and it involves testing individual modules of the software. It is usually performed by developers.

## **Q33 What is integration testing?**

Ans. Integration testing is performed after unit testing. In integration testing, we test the group of related modules. It aims at finding interfacing issues between the modules.

## **Q34 What are the different types of integration testing?**

Ans. The different type of integration testing is-

1. Big bang Integration Testing – In big bang integration testing, testing starts only after all the modules are integrated.
2. Top-down Integration Testing – In top-down integration, testing/integration starts from top modules to lower-level modules.
3. Bottom-up Integration Testing – In bottom-up integration, testing starts from lower-level modules to higher-level modules up in the hierarchy.

4. Hybrid Integration Testing – Hybrid integration testing is the combination of both Top-down and bottom-up integration testing. In this approach, the integration starts from the middle layer, and testing is carried out in both the direction

### **Q35 What is Exploratory Testing?**

Exploratory testing is a type of software testing where the tester actively explores the application under test to identify defects, rather than following a predetermined test plan or script. It's an iterative and interactive process that allows the tester to gain a deeper understanding of the software and to uncover unexpected issues or behaviors.

### **Q36 What is “use case testing”?**

In order to identify and execute the functional requirement of an application from start to finish “use case” is used and the techniques used to do this is known as “Use Case Testing.”

### **Q37 What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

SDLC deals with development/coding of the software while STLC deals with validation and verification of the software

### **Q38 What is a traceability matrix?**

The relationship between test cases and requirements is shown with the help of a document. This document is known as a traceability matrix.

### **Q39 What is the difference between static and dynamic testing?**

**Static testing:** During the Static testing method, the code is not executed, and it is performed using the software documentation.

**Dynamic testing:** To perform this testing the code is required to be in an executable form.

## **Q40 What is System Testing?**

**System Testing** is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is defined as a series of different tests whose sole purpose is to exercise the full computer-based system.

## **Q41 What is acceptance testing?**

Ans. Acceptance testing is testing performed by the potential end-user or customers to check if the software conforms to the business requirements and can be accepted for use.

## **Q42 What is UAT Testing?**

Ans UAT testing is the last phase of the testing lifecycle. Its main focus is to validate that software is working in accordance with business requirements. It also ensures that the application is user-friendly and can handle complex scenarios at its best before releasing the product to real-world users.

## **Q43 What is End-To-End Testing?**

Ans. End-to-End testing is a type of testing where the entire application undergoes testing, to test that each functionality of the software is working as expected and there is no loophole remaining in it. It ensures that the application is user-friendly and meets the business requirements.

## **Q44 What is alpha testing?**

Ans. Alpha testing is a type of acceptance testing that is performed by testers or the internal employees of the organization at the developer site.

## **Q45 What is beta testing?**

Ans. Beta testing is the testing done by end-users at the end user's site. It allows users to provide direct input about the software to the development company.

## **Q46 What is Adhoc Testing?**

Ans. Adhoc testing is an unstructured way of testing that is performed without any formal documentation or proper planning.

## **Q47 What is monkey testing?**

Ans. Monkey testing is a type of testing that is performed randomly without any predefined test cases or test inputs.

## **Q48 How is monkey testing different from Adhoc testing?**

Ans. In the case of Adhoc testing although there are no predefined or documented test cases still testers have an understanding of the application. While in the case of monkey testing testers don't have any understanding of the application.

## **Q49 What is exploratory testing?**

Ans. Exploratory testing is a type of testing in which new test cases are added and updated while exploring the system or executing test cases. Unlike scripted testing, test design and execution go parallel in exploratory testing.

## **Q50 What is load testing?**

Ans. Load testing is a type of performance testing which aims at finding an application's performance under the expected workload. During load testing, we evaluate the response time, throughput, error rate, etc parameters of the application.

## **Q51 What is stress testing?**

Ans. Stress testing is a type of performance testing in which an application's behavior is monitored under a higher workload than expected. Stress testing is done to find memory leaks and the robustness of the application.

## **Q52 What is volume testing?**

Ans. Volume testing is a type of performance testing in which the performance of the application is evaluated with a large amount of data. It checks the scalability of the application and helps in the identification of a bottleneck with a high volume of data.

## **Q53 What is endurance testing or Soak testing?**

Ans. Endurance testing is a type of performance testing which aims at finding issues like memory leaks when an application is subjected to load tests for a long period of time.

## **Q54 What is spike testing?**

Ans. Spike testing is a type of performance testing in which the application's performance is measured while suddenly increasing the number of active users during the load test.

## **Q55 What is UI testing?**

Ans. UI or user interface testing is a type of testing that aims at finding Graphical User Interface defects in the application and checks that the GUI conforms to the specifications.

## **Q56 What is usability testing?**

Ans. Usability testing is the type of testing that aims at determining the ease of using the application. It aims at uncovering the usability defects in the application.

## **Q57 What is Accessibility testing?**

Ans. Accessibility testing is the type of testing which aims at determining the ease of use or operation of the application specifically for people with disabilities.

## **Q58 What is compatibility testing?**

Ans. Compatibility testing is validating software to see how compatible the software is with a particular environment – operating system, platform, or hardware.

## **Q59 What is configuration testing?**

Ans. Configuration testing is the type of testing used to evaluate the configurational requirements of the software along with the effect of changing the required configuration.

## **Q60 What is localization testing?**

Ans. Localization testing is a type of testing in which we evaluate the application's customization(a localized version of the application) in a particular culture, locale or country.

## **Q61 What is globalization testing?**

Ans. Globalization testing is a type of testing in which an application is evaluated for its functioning across the world in different cultures, languages, locales, and countries.

## **Q62 What is negative testing?**

Ans. Negative testing is a type of testing in which the application's robustness(graceful exiting or error reporting) is evaluated when provided with invalid input or test data.

## **Q63 What is security testing?**

Ans. Security testing is a type of testing which aims at evaluating the integrity, authentication, authorization, availability, confidentiality, and non-repudiation of the application under test.

## **Q64 What is penetration testing?**

Ans. Penetration testing or pen testing is a type of security testing in which an application is evaluated(safely exploited) for different kinds of vulnerabilities that any hacker could exploit.

## **Q65 What is robustness testing?**

Ans. Robustness testing is a type of testing that is performed to find the robustness of the application i.e. the ability of the system to behave gracefully in case of erroneous test steps and test input.

## **Q66 What is concurrency testing?**

Ans. Concurrency testing is multi-user testing in which an application is evaluated by analyzing the application's behavior with concurrent users accessing the same functionality.

## **Q67 What is backend testing?**

Ans. Backend testing is a type of testing that involves testing the backend of the system which comprises testing the databases and the APIs in the application.

## **Q68 What is A/B testing?**

Ans. A/B testing is a type of testing in which the two variants of the software product are exposed to the end-users and on analyzing the user behavior on each variant, the better variant is chosen and used thereafter.

## **Q69 What is risk analysis?**

Ans. Risk analysis is the analysis of the risk identified and assigning an appropriate risk level to the defect based on its impact on the application.

## **Q70 What is the difference between regression and retesting?**

Ans. Regression testing involves testing the application to verify that a new code change doesn't affect the other parts of the application. Whereas, in retesting, we verify if the fixed issue is resolved or not.

## **Q71 What is the difference between black-box and white-box testing?**

Ans.

- **Black-box testing** is a type of testing in which the internal architecture of the code is not required for testing. It is usually applicable for system and acceptance testing.
- **white-box testing** requires internal design and implementation knowledge of the application being tested. It is usually applicable for Unit and Integration testing.

## **Q72 What is the difference between smoke and sanity testing?**

Ans. The difference between smoke and sanity testing is-

- Smoke testing is a type of testing in which all major functionalities of the application are tested before carrying out exhaustive testing. Whereas, sanity testing is a subset of regression testing which is carried out when there is some minor fix in the application in a new build.
- In smoke testing, shallow-wide testing is carried out while in Sanity, narrow-deep testing (for a particular functionality) is done.
- The smoke tests are usually documented or automated. Whereas, the sanity tests are generally not documented or unscripted.

## **Q73 What is the difference between Release and Build?**

Ans. A build is an executable file provided by the developers to the testing team for testing the application. It undergoes various iterations of fixing and testing until the application works as expected. Once the application becomes stable and ready for the end-users, it's released in the market.

Whereas, a release is installable software provided to the end-users after it gets certified by the testing team. During the release of any software to the client, release notes are attached to it that includes a number of defects still open, covered user stories, change requirements, and version of the release.

## **Q74. What is API testing?**

Ans. **API Testing** is a software testing type that validates Application Programming Interfaces (APIs). The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces. In API Testing, instead of using standard user inputs(keyboard) and outputs, you use software to send calls to the API, get output, and note down the system's response. API tests are very different from GUI Tests and won't concentrate on the look and feel of an application. It mainly concentrates on the business logic layer of the software.

## **Q75. When should you stop the testing process?**

The testing activity ends when the testing team completes the following milestones.

### **Test case execution**

The successful completion of a full test cycle after the final bug fix marks the end of the testing phase.

### **Testing deadline**

The end date of the validation stage also declares the closure of the validation if no critical or high-priority defects remain in the system.

### **Code Coverage(CC) ratio**

It is the amount of code concealed via automated tests. If the team achieves the intended level of code coverage (CC) ratio, then it can choose to end the validation.

### **Mean Time Between Failure (MTBF) rate**

Mean time between failure (MTBF) refers to the average amount of time that a device or product functions before failing. This unit of measurement includes only operational time between failures and does not include repair times, assuming the item is repaired and begins functioning again. MTBF figures are often used to project how likely a single unit is to fail within a certain period of time.

## **Q76 What is meant by test coverage?**

Test coverage is a quality metric to represent the amount (in percentage) of testing completed for a product. It is relevant for both functional and non-functional testing activities. This metric is used to add missing test cases.

## **Q77 Is it possible to achieve 100% testing coverage? How would you ensure it?**

It's considered not possible to perform 100% testing of any product. But you can follow the below steps to come closer.

- **Set a hard limit on the following factors:**
  - Percentage of test cases passed
  - Number of bugs found
- **Set a red flag if:**
  - Test budget is depleted
  - Deadlines are breached
- **Set a green flag if:**
  - The entire functionality gets covered in test cases
  - All critical and major bugs must have a 'CLOSED' status

## **Q78 What are unit testing and integration testing?**

Unit testing has many names such as module testing or component testing.

Many times, it is the developers who test individual units or modules to check if they are working correctly.

Whereas, integration testing validates how well two or more units of software interact with each other.

There are three ways to validate integration:

- Big Bang approach
- Top-down approach
- Bottom-up approach

## **Q79 Can we do system testing at any stage?**

No. System testing should start only if all modules are in place and they work correctly. However, it should be performed before UAT (user acceptance testing).

## **Q80 What are the drawbacks of manual testing?**

**1. Manual Testing requires more time or more resources, sometimes both Time and Resources.**

**2. Less Accuracy**

(Human Users (Testers) may make mistakes, so we cannot expect more accuracy in Manual Testing,

**3. Performance testing is impractical in manual testing.**

**4. Comparing a large amount of data is impractical.**

(Comparing two Databases that have thousands of records is impractical, but it is very useful in Test Automation.)

**5. Processing change requests during software maintenance takes more time.**

**6. Batch Testing is possible, but for each test execution Human user interaction is mandatory.**

**7. GUI Objects Size difference and Color combinations etc.. are not easy to find in Manual Testing.**

**8. Manual Test Case scope is very less, if it is Automated test, scope is more.**

**9. Executing the same tests, again and again, is time taking process as well as Tedious.**

**10. For every release you must rerun the same set of tests which can be tiresome.**

**Q81. Explain the difference between alpha testing and beta testing.**

- **Alpha Testing** – It is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is a type of user acceptance testing.
- **Beta Testing** – It is performed by real users of the software application in a real environment. Beta Testing is also a type of user acceptance testing.

**Q82 What Test Plans consists of?**

Test design, scope, test strategies, approach are various details that Test plan document consists of.

1. Test case identifier
2. Scope
3. Features to be tested
4. Features not to be tested
5. Test strategy & Test approach
6. Test deliverables
7. Responsibilities
8. Staffing and training

## 9. Risk and Contingencies

### **Q83 What is agile testing and why is it important?**

Agile testing is a software testing process that evaluates software from the customers' point of view. It is favorable as it does not require the development team to complete coding for starting QA. Instead, both coding and testing go hand in hand. However, it may require continuous customer interaction.

### **Q84 Why is it impossible to test a program thoroughly?**

Here are the two principal reasons that make it impossible to test a program entirely.

- Software specifications can be subjective and can lead to different interpretations.
- A software program may require too many inputs, outputs, and path combinations.

### **Q86 What is the purpose of the end-to-end testing?**

End-to-end testing is a testing strategy to execute tests that cover every possible flow of an application from its start to finish. The objective of performing end-to-end tests is to discover software dependencies and to assert that the correct input is getting passed between various software modules and sub-systems.

### **Q87 What is a testbed in manual testing?**

The testbed is an environment configured for testing. It is an environment used for testing an application, including the hardware as well as any software needed to run the program to be tested. It consists of hardware, software, network configuration, an application under test, other related software.

### **Q88 What is the difference between UAT (User Acceptance Testing) and System testing?**

**System Testing:** System testing is finding defects when the system undergoes testing as a whole; it is also known as end-to-end testing. In such type of testing, the application suffers from beginning till the end.

**UAT:** User Acceptance Testing (UAT) involves running a product through a series of specific tests which determines whether the product will meet the needs of its users.

## **Q89 Mention the difference between Data Driven Testing and Retesting?**

Retesting: It is a process of checking bugs that are actioned by the development team to verify that they are fixed.

Data Driven Testing (DDT): In data driven testing process, the application is tested with multiple test data. The application is tested with a different set of values.

## **Q90 How will you overcome the challenges faced due to the unavailability of proper documentation for testing?**

If the standard documents like System Requirement Specification or Feature Description Document are not available, then QAs may have to rely on the following references, if available.

- Screenshots
- A previous version of the application
- Wireframes

Another reliable way is to have discussions with the developer and the business analyst. It helps in solving the doubts, and it opens a channel for bringing clarity on the requirements. Also, the emails exchanged could be useful as a testing reference.

Smoke testing is yet another option that would help verify the main functionality of the application. It would reveal some very basic bugs in the application. If none of these work, then we can just test the application from our previous experiences.

## **Q91 How much testing is sufficient? Or, is it possible to do exhaustive testing of the software?**

It is impossible to exhaustively test software or prove the absence of errors, no matter how specific your test strategy is.

An extensive test that finds hundreds of errors doesn't imply that it has discovered them all. There could be many more errors that the test might have missed. The absence of errors doesn't mean there are no errors, and the software is perfect. It could easily mean ineffective or incomplete tests. To prove that a program works, you'd have to test all possible inputs and their combinations.

Consider a simple program that takes a string as an input that is ten characters long. To test it with each possible input, you'd have to enter 2610 names, which is impossible. Since exhaustive testing is not practical, your best strategy as a tester is to pick the test cases that are most likely to find errors. Testing is sufficient when you have enough confidence to release the software and assume it will work as expected.

## **Q93 Why developers shouldn't test the software they wrote?**

Developers make poor testers. Here are some reasons why:

- They try to test the code to make sure that it works, rather than testing all the ways in which it doesn't work.
- Since they wrote it themselves, developers tend to be very optimistic about the software and don't have the correct attitude needed for testing: to break software.
- Developers skip the more sophisticated tests that an experienced tester would perform to break the software. They follow the happy path to execute the code from start to finish with proper inputs, often not enough to get the confidence to ship software in production.

However, it doesn't mean that developers shouldn't test the software before sending it to the tester. Developer testing helps find many bugs that are caused by programming errors. These are hard to find for a tester because they don't always have access to the source code

## **Q94 What is automated testing?**

As the name suggests, automated testing, which is also called test automation, is the programmatic execution of the tests. The tester uses an automation tool or software like Selenium to write code that performs the following tasks.

1. Automatically run the software.
2. Feed the input data to the system.
3. Examine the output with the expected outcome.
4. Fail the test if the results don't match. Otherwise, pass the test.

## **Q95 What are the various components of Selenium?**

Selenium consists of the following components:

- Selenium Remote Control (RC).
- Selenium Integrated Development Environment (IDE).
- Selenium WebDriver.
- Selenium Grid.

## **Q96 What are the different HTTP status codes that a server can return?**

An HTTP status code is a three-digit number that indicates the status of an incoming HTTP request, that is, if the request has been completed or not.

A server can send the following five types of responses for an HTTP request.

1. **Information (100 - 199):** These status codes provide a temporary response. The response consists of the status line and optional headers and terminates by an empty line.
2. **Success (200 - 299):** Indicate that the incoming HTTP request was successfully received, understood, and accepted.
3. **Redirect (300 - 399):** These status codes indicate further actions the client should take to satisfy the HTTP request. It can mean that the requested resource may have moved temporarily or permanently. It can also redirect the client to another URL.
4. **A client error (400 - 499):** Indicate a problem with the client who initiated the HTTP request.
5. **Server error (500 - 599):** The 5XX status code indicates a problem on the server while processing the request.

## **Q97 What is the most effective way to map automation testing success?**

The success of Automation testing can be measured using the following criteria:

- Savings on labor & other costs.
- Defect Detection Ratio.
- Automating the execution process and reducing the release time

## **Q98 Which test cases are written first: white boxes or black boxes?**

Test cases for the black box testing are usually written first, followed by test cases for the white box testing. An outline of the design or project plan and the requirements document is required to write black-box test cases.

Documents such as these are readily available at the beginning of the project. The initial phase of a project isn't the right time to start white box testing because it requires more architecture clarification that isn't available yet. Therefore, white-box test cases are typically written after black-box test cases have been developed.

## **Q99 What is meant by browser automation?**

It's a process of automatically testing a web application's functionality in a browser, where a program launches the browser, navigates to the application, and interacts with the user interface by clicking buttons or links, just like an average user would.

The only difference is that the browser automation can test this very quickly and often, whereas the same test would take a human tester a long time. It's part of automated testing. Some essential tools for browser testing include Selenium, protractor.js, and cypress.

## **Q100 What do you mean by Test Matrix and Traceability Matrix?**

Test Matrix: It is referred to as a testing tool that is used to capture actual quality, effort, resources, plan, and time required to capture all the phases of software testing. It only covers the testing phase of the life cycle.

Requirement Traceability Matrix (RTM): It is referred to as a document, usually present in the form table, that is used to trace and demonstrate the relationship between the requirements and other artifacts of the project right from start to end. In simple words, it maps between test cases and customer requirements.

## **Q101 What is the MAIN benefit of designing tests early in the life cycle?**

It helps prevent defects from being introduced into the code.

## **Q102 Give an example of Low priority-Low severity, Low priority-High severity, High priority-Low severity, and High priority-High severity defects.**

Ans. Below are examples for different combinations of priority and severity-

1. **Low priority-Low severity** – A spelling mistake in a page not frequently navigated by users.
2. **Low priority-High severity** – Application crashing in some very corner cases.
3. **High priority-Low severity** – Slight change in logo color or spelling mistake in the company name.
4. **High priority-High severity** – Issue with login functionality.

## 3 To 5 Years Of Experience

Q19. When should you opt for manual testing over automation testing?

There are a lot of cases when manual testing is best suited over automation testing, like:

- **Short-time projects:** Automated tests are aimed at saving time and resources yet it takes time and resources to design and maintain them. For example, if you are building a small promotional website, it can be much more efficient to rely on manual testing.
- **Ad-hoc Testing:** In ad-hoc testing, there is no specific approach. Ad-hoc testing is a totally unplanned method of testing where the understanding and insight of the tester is the only important factor. This can be achieved using manual testing.
- **Exploratory Test:** This type of testing requires the tester's knowledge, experience, analytical, logical skills, creativity, and intuition. So human involvement is important in exploratory testing.
- **Usability Testing:** When performing usability testing, the tester needs to measure how user-friendly, efficient, or convenient the software or product is for the end-users. Human observation is the most important factor, so manual testing sounds seems more appropriate.

Q26. What is the test harness?

Q27. What is test closure?

Q29. Define what is a critical bug.

What is a Failure in software testing?

Q31. What is Defect Cascading in Software Testing?

Q32. What is the term 'quality' mean when testing?

Q36.What is a top-down and bottom-up approach in testing?

**Top-Down** – Testing happens from top to bottom. That is, high-level modules are tested first, and after that low-level modules. Lastly, the low-level modules are incorporated into a high-level state to guarantee the framework is working as it is expected to.

**Bottom-Up** – Testing happens from base levels to high-up levels. The lowest level modules are tested first and afterward high-level state modules. Lastly, the high-level state modules are coordinated to a low level to guarantee the framework is filling in as it has been proposed to.

Q38. What is the difference between static testing and dynamic testing?

Static Testing	Dynamic Testing
<b>Static Testing is a white box testing technique, it includes the process of exploring the records to recognize the imperfections in the very early stages of SDLC.</b>	<b>Dynamic testing includes the process of execution of code and is done at the later stage of the software development lifecycle. It validates and approves the output with the expected results.</b>
<b>Static Testing is implemented at the verification stage.</b>	<b>Dynamic testing starts during the validation stage.</b>
<b>Static testing is performed before the code deployment.</b>	<b>Dynamic testing is performed after the code deployment</b>
<b>The program's code error detection and execution is not a concern in this type of testing.</b>	<b>Execution of code is necessary for dynamic testing.</b>

### Q39. How will you determine when to stop testing?

Deciding when to stop testing can be quite difficult. Many modern software applications are so complex and run in such an interdependent environment, that complete testing can never be done. Some common factors in deciding when to stop testing are:

- Deadlines (release deadlines, testing deadlines, etc.)
- Test cases completed with certain percentage passed
- When the test budget is depleted
- Coverage of code or functionality or requirements reaches a specified point
- Bug rate falls below a certain level
- When Beta or alpha testing period ends

### Q40. What if the software is so buggy it can't really be tested at all?

Often testers encounter a bug that can't be resolved at all. In such situations, the best bet is for testers to go through the process of reporting whatever bugs or blocking-type problems initially show up, with the focus being on critical bugs. Since this type of problem can cause severe problems such as insufficient unit testing or insufficient integration testing, poor design, improper build or release procedures, etc managers should be notified and provided with some documentation as evidence of the problem.

### Q41. How you test a product if the requirements are yet to freeze?

It's possible that a requirement stack is not available for a piece of product. It might take serious effort to determine if an application has significant unexpected functionality, and it would indicate deeper problems in the software development process. If the functionality isn't necessary to the purpose of the application, it should be removed. Else, create a test plan based on the assumptions made about the product. But make sure you get all assumptions well documented in the test plan.

Q44. What are the cases when you'll consider to choose automated testing over manual testing?

Automated testing can be considered over manual testing during the following situations:

- When tests require periodic execution
- Tests include repetitive steps
- Tests need to be executed in a standard runtime environment
- When you have less time to complete the testing phase
- When there is a lot of code that needs to be repeatedly tested
- Reports are required for every execution

Q45. What is 'configuration management'?

Every high-functioning organization has a "master plan" that details how they are supposed to operate and accomplish tasks. Software development and testing are no different. Software configuration management (SCM) is a set of processes, policies, and tools that organize, control, coordinate, and track:

- code
- documentation
- problems
- change requests
- designs and tools
- compilers and libraries

Q46. Is it true that we can do system testing at any stage?

In system testing, all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified. So, no. The system testing must start only if all units are in place and are working properly. System testing usually happens before the UAT (User Acceptance Testing).

**Ques.21. What is a Test script?**

Ans. A test script is an automated test case written in any programming or scripting language. These are basically a set of instructions to evaluate the functioning of an application.

**Ques.11. What is automation testing?**

Ans. Automation testing is a type of software testing that involves automated test case execution using an automation tool. It helps in reducing the test execution time as the test scripts written once, can be run automatically any number of times without any human intervention.

**Ques.12. What are some advantages of automation testing?**

Ans. Some advantages of automation testing are-

1. Test execution using automation is fast and saves a considerable amount of time.
2. Carefully written test scripts remove the chance of human error during testing.
3. Test execution can be scheduled for a nightly run using CI tools like Jenkins which can also be configured to provide daily test results to relevant stakeholders.
4. Automation testing is very less resource-intensive. Once the tests are automated, test execution requires almost no time of QAs. Saving QA bandwidth for other exploratory tasks.

### **Ques.13. What are some disadvantages of automation testing?**

Ans. Some disadvantages of automation testing are-

1. It requires skilled automation testing experts to write test scripts.
2. Additional effort to write scripts is required upfront.
3. Automation scripts are limited to verification of the tests that are coded. These tests may miss some error that is very glaring and easily identifiable to human(manual QA).
4. Even with some minor change in the application, script update and maintenance is required.

### **Ques.14. What is performance testing?**

Ans. Performance testing is a type of non-functional testing in which the performance of the system is evaluated under expected or higher load. The various performance parameters evaluated during performance testing are – response time, reliability, resource usage, scalability, etc. The different types of performance testing are – Load, Stress, Endurance, Spike, and Volume Testing.

Q47. What are some best practices that you should follow when writing test cases?

Few guidelines that you need to follow while writing test cases are:

- Prioritize which test cases to write based on the project timelines and the risk factors of your application.
- Remember the 80/20 rule. To achieve the best coverage, 20% of your tests should cover 80% of your application.
- Don't try to test cases in one attempt instead improvise them as you progress.
- List down your test cases and classify them based on business scenarios and functionality.
- Make sure test cases are modular and test case steps are as granular as possible.
- Write test cases in such a way that others can understand them easily & modify if required.
- Always keep end-users' requirements in the back of your mind because ultimately the software designed is for the customer
- Actively use a test management tool to manage stable release cycle.
- Monitor your test cases regularly. Write unique test cases and remove irrelevant & duplicate test cases.

Q50. Can automation testing replace manual testing?

Automation testing isn't a replacement for manual testing. No matter how good automated tests are, you cannot automate everything. Manual tests play an important role in software development and come in handy whenever you have a case where you cannot use automation. Automated and manual testing each have their own strengths and weaknesses. Manual testing helps us understand the entire problem and more flexibly explore other angles of tests. On the other hand,

automated testing helps save time in the long run by accomplishing a large number of surface-level tests in a short time.

### **Ques.34. What are the different test design techniques?**

Ans. Test design techniques are different standards of test designing that allow systematic and widely accepted test cases. The different test design techniques can be categorized as static test design techniques and dynamic test design techniques.

1. Static Test Design Techniques – The test design techniques which involve testing without executing the code. The various static test design techniques can be further divided into two parts manual and using tools-

- ○ Manual static design techniques
  - Walkthrough
  - Informal reviews
  - Technical reviews
  - Audit
  - Inspection
  - Management review
- ○ Static design techniques using tools

- Static analysis of code – It includes analysis of the different paths and flows in the application and different states of the test data.
- Compliance with coding standards – This evaluates the compliance of the code with the different coding standards.
- Analysis of code metrics – The tool used for static analysis is required to evaluate the different metrics like lines of code, complexity, code coverage, etc.

## 2. Dynamic Test Design Techniques – Dynamic test design techniques involve testing by running the system under test.

- Specification-based – Specification-based test design techniques are also referred to as black-box testing. These involve testing based on the specification of the system under test without knowing its internal architecture.
- Structure-based – Structure-based test design techniques are also referred to as white box testing. In these techniques, the knowledge of the code or internal architecture of the system is required to carry out the testing.

- Experienced-based – The experienced-based techniques are completely based on the experience or intuition of the tester. The two most common forms of experienced-based testing are – Adhoc testing and exploratory testing.

#### **Ques.45. What is structure-based testing?**

Ans. Structure-based test design techniques are also referred to as white box testing. In these techniques, the knowledge of the code or internal architecture of the system is required to carry out the testing. The various kinds of testing structure-based or white testing techniques are-

- **Statement testing** – A white box testing technique in which the test scripts are designed to execute the application's code statements. Its coverage is measured as the line of code or statements executed by test scripts.
- **Decision testing/branch testing** – A testing technique in the test scripts is designed to execute the different decision branches (e.g. if-else conditions) in the application. Its coverage is measured as the percentage of decision points out of the

total decision points in the application.

- **Condition testing** – Condition testing is a testing approach in which we test the application with both True and False outcomes for each condition. Hence for  $n$  conditions, we will have  $2^n$  test scripts.
- **Multiple condition testing** – In multiple condition testing, the different combinations of condition outcomes are tested at least once. Hence for 100% coverage, we will have  $2^n$  test scripts. This is very exhaustive and very difficult to achieve 100% coverage.
- **Condition determination testing** – It is an optimized way of multiple condition testing in which the combinations which don't affect the outcomes are discarded.
- **Path testing** – Testing the independent paths in the system(paths are executable statements from entry to exit points).

Q17. What's the role of documentation in Manual Testing?

**5 To 7 Years Of Experience**