



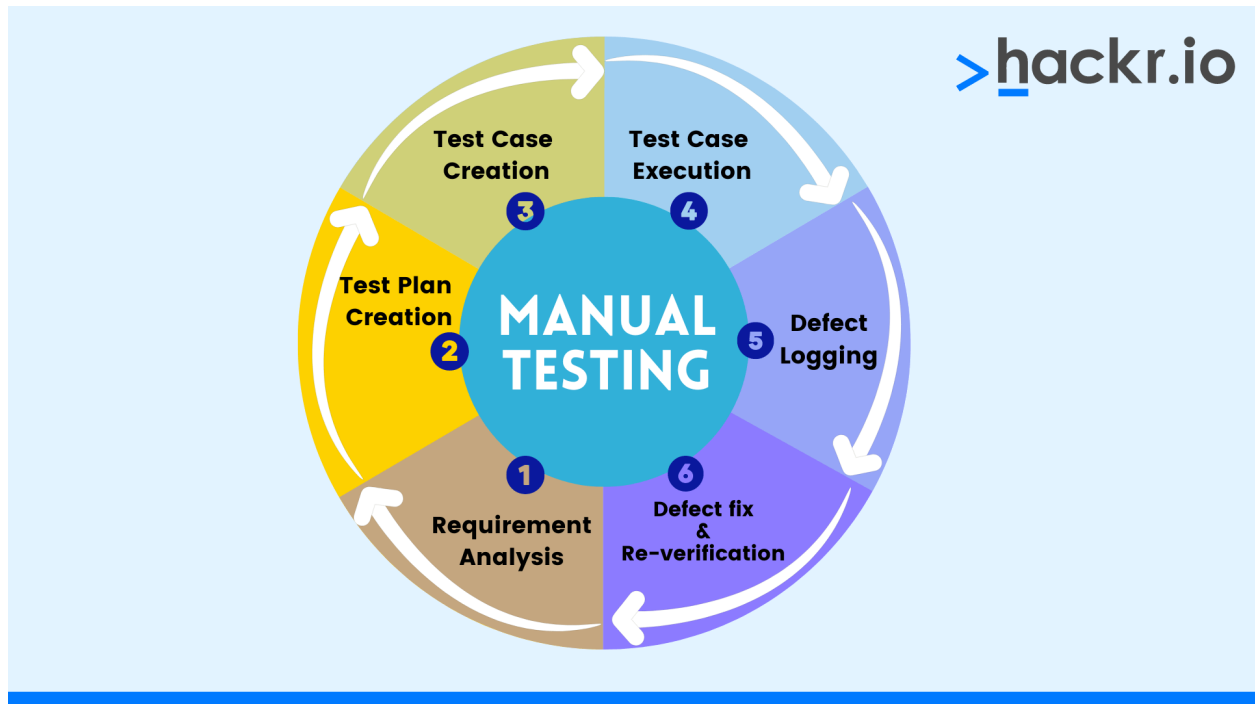
If you're thinking of becoming a manual tester, you'll have to ensure that you're knowledgeable about the basics and more advanced concepts. Manual testers will have to prepare well, just like you would for any other interview.

Here we list some of the most common manual testing interview questions. We've divided this list into beginner manual testing questions and advanced manual testing questions.

## Top Manual Testing Interview Questions and Answers

### Manual Testing Beginner Interview Questions

#### 1. How would you define Manual Testing?



Software testing is a validation process that makes sure a system works according to business requirements. It evaluates and qualifies a system on various aspects such as accuracy, completeness, usability, and efficiency.

## 2. What is use-case testing?

"Use case" is used to identify and execute the functional requirement of an application from start to finish, and the technique used to do this is known as "Use Case Testing."

## 3. What is Static Testing?

Static testing is a white box testing technique that, with the help of a checklist, directs developers to verify their code to find errors in it. Developers can start static testing without finalizing the program or application. Static testing is more cost-effective than dynamic testing as it covers more areas than dynamic testing in a shorter time.

## 4. What are the various types of manual testing?

In manual testing, the software is tested without any automation tools or scripts. The various types of manual testing are:

- **Smoke Testing:** Smoke testing tests for the presence of critical issues in the software and ensures that the software adheres to its primary objectives. It overlooks the minor and cosmetic errors and tests only the core functionality. For example, testing a new feature.
- **Cross-Browser Testing:** a website may show some effects and displays differently. Thus, it is crucial to test each functionality, especially UI on different browsers to ensure the functionality is working correctly. For example, checking JavaScript functionality on all browsers.
- **Acceptance Testing:** Acceptance testing, better known as User Acceptance Testing (UAT), aims to test how much the application can cater to the needs of users. It is done after all the other testing types have finished and the product is ready to go live. Generally, it is performed by an actual user or the client. For example, a user can run through the end-to-end process of purchasing a mobile phone without any hassles.
- **Beta Testing:** Beta testing is done during the pilot phase of the project, i.e., a soft launch just before the website is made fully public. It is done to get feedback from real users based on their experience. This integration can be tested only once the application is put live. For example, a service was independently developed but used other systems to perform correctly.
- **Exploratory Testing:** This is like random testing, where a person can test anything they want. They can click on any button, go back and forth on the website pages, and explore the site end to end. Usually, this type of testing is done by designers, product managers, and even developers to identify any critical usability issues. For example, the impact of the browser back button from a credit card landing page.

- **Negative Testing:** Through negative testing, testers can identify how the application handles errors and exceptions by purposefully entering wrong inputs. For example, leaving the password field blank while logging into a page.
- **Usability Testing:** This is one of the most important testing types because it involves dealing with how a user feels about using your product. It tests whether your website is easy to navigate, whether they can search what they want quickly, that they can do their business fast, and so on. It deals with user satisfaction.

## 5. What is API testing?

API testing is the testing of the Application Programming Interface directly as well as during integration testing to ensure the APIs deliver the required functionality, performance, security, and reliability. The tests are written to check if the API gives correct responses to specific requests. API testing involves unit testing, load testing, functional testing, security testing, penetration testing, Web UI testing, etc.

## 6. What is the difference between system testing and integration testing?

### System Testing

The whole system is checked as one (end-to-end).

It includes both functional and non-functional testing (usability, performance, etc.).

Done after integration testing.

### Integration Testing

The interfaces between different interconnected systems are checked.

It checks only the functionality of the integrated components.

Done after the unit testing phase.

The various types are performance testing, usability testing, functional testing, installation testing, security testing, etc.

Uses only black-box testing techniques.

Some methods or approaches to perform integration testing are top-down, bottom-up, hybrid, and big-bang.

Both white-box and black-box techniques are used.

## 7. What is test closure?

Test closure is a document that details the tests conducted during the entire [SDLC](#), the analysis of the bugs and errors found and corrected, the density of defects, etc. It is a memo that indicates the formal completion of the testing procedure.

## 8. Explain the pesticide paradox. How do you deal with it?

Pesticide paradox is a phenomenon wherein the more you test your software, the more immune it becomes to your tests. Testers should always look for new strategies, approaches, and test cases to overcome this, for different parts of the same applications, so that they can find bugs.

## 9. Explain the difference between smoke testing and sanity testing.

### Smoke Testing

Verifies the basic functionalities and check if the build is working fine and has no errors.

Done on initial builds, i.e., it is the first check performed once a build is done.

Performed for every build.

### Sanity Testing

Verifies the newly added functionalities or bug fixes are corrected and working fine.

Done on a stable build after smoke testing is successful.

Performed on stable builds to check if the old functionality is working after the new changes are done.

## **10. How do you know when to stop testing?**

The most straightforward way would be when no defects are not found in the software. However, it is not possible to have software entirely free of bugs. We can determine the exit criteria for testing based on the deadlines, budget, and the extent of testing performed.

Usually, testers can find most of the major and critical bugs during the first and second weeks of testing. After the third and fourth weeks, minor and cosmetic defects are taken care of, and the application moves into the regression testing phase. Once regression is completed, we can be assured that 99% of test scenarios have been covered, and software is ready to be rolled out.

## **11. What is configuration management?**

Configuration management is a set of management practices, interrelated processes, and tools that help manage work items, and keep them updated and accurate so that they are readily available for all the users involved. It also helps us maintain all the work items in conformation with the requirements of the project and ensures any changes are appropriately documented, evaluated, and implemented after authorization.

## **12. What is the difference between a test driver and a test stub?**

Both driver and stub are types of test harness that are used to provide a simulation environment for testing a module (component). They are not a part of the main software and are used only for testing purposes.

## **13. What is Data flow Testing?**

Data flow testing involves testing various paths in the application workflow to test the sequence of events and the status of variables and data objects at different stages. Through data flow testing, we can identify the variables that are used at every stage of the program's control flow. This helps us eliminate variables that are declared but never used or vice versa, deallocating variables before using them, or a variable whose value is changed multiple times before being used.

## **14. What is the difference between retesting and regression testing?**

### **Retesting**

Checks if the previously failed test cases have been fixed and are working fine.

The primary purpose is to ensure the bugs found in a previous build are fixed in the latest one.

Retesting affects the application delivery date and is of high priority; it is done before regression testing.

Retesting has to be done manually.

Only failed test cases are executed.

### **Regression testing**

Identifies new issues that may have occurred because of changes to the application functionality or introducing new functionality.

The basic purpose is to ensure that the new changes have not affected the existing working functionality.

Regression testing is done after all the other testing procedures are complete.

Test cases can be automated, and scripts can be run to test the same functionality again and again.

All the test cases from the start to the end of the application are executed.

## **15. What is the difference between Static Testing and Dynamic Testing?**

### **On the Basis Of**

#### **Static Testing**

#### **Dynamic Testing**

#### **When To Perform**

Testing was done without executing the program

Testing is done by executing the program

#### **What Does it Do**

This testing does the verification process

Dynamic testing does the validation process

What is it About	Static testing about preventing defects	Dynamic testing is about fixing and finding the defects
Output Of the Testing	Static testing gives an assessment of documentation and code	Dynamic testing gives bugs in the software system.
What Does it Involve?	Static testing involves a process and checklist to be followed	Dynamic testing involves test cases for execution
When is the Testing Performed w.r.t Compilation	This testing can be performed before the compilation	Dynamic testing is performed after compilation
What does it Cover?	Static testing covers the statement and structural coverage testing	Dynamic testing covers the executable file of the code
Cost of Finding Defects	Cost of finding defects and fixing is less	Cost of finding and fixing defects is high
Return on Investment	Return on investment is high as this process involved at an early stage	Return on investment is low as this process involves after the development phase

## 16. What is the difference between STLC and SDLC?

STLC deals with the verification and validation of the software while [SDLC](#) deals with the development/coding of the software.

## 17. What is Test coverage?

Test coverage is a quality metric that represents the percentage amount of testing completed for a product. It is relevant for both non-functional and functional testing activities. The metric is used to add missing test cases.

## 18. Can system testing be done at any stage?



No, we cannot do system testing at any stage, it must start only if all modules work correctly and are in place, but it should be performed before UAT (user acceptance testing).

## **19. What are the different kinds of software testing?**

Manual testers use the following [kinds of Software testing](#):

- Unit testing
- Integration testing
- Regression testing
- Shakeout testing
- Smoke testing
- Functional testing
- Performance testing
  - Load testing
  - Stress testing
  - Endurance testing
- White-box and Black-box testing
- Alpha and Beta testing
- System testing

## **20. What is agile testing and what is its importance?**

Agile testing evaluates software from the perspective of the customer. Its importance is that it does not require the development team to complete coding for starting QA. Instead, testing and coding go hand in hand. However, it may require continuous customer interaction.

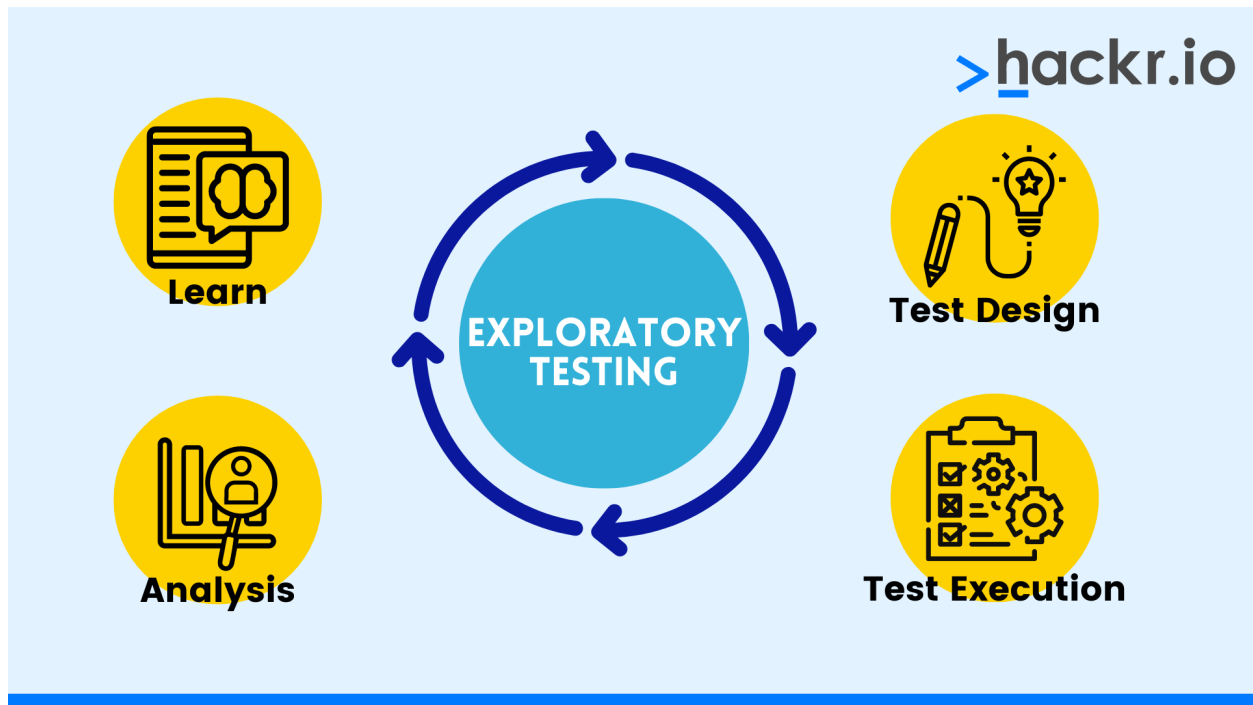
## **21. What should you do when you discover bugs?**

The following steps must be followed when a bug is encountered while testing:

- Run more tests to confirm that there is a clear description of the problem

- Ensure that the same problem doesn't exist with different inputs by running a few more tests
- Once we are certain about the full scope of the bug, we can add details and report it

## 22. What is Exploratory Testing?



Exploratory testing is testing an application without a test plan and test script. In exploring the testing test, explore the application based on the tester's knowledge as the tester does not know the application previously. Tester explores the application like an end-user and tries to use it. While using the application, the main motive is to find the bugs which are in the application.

## 23. What is white boxing testing and what are the types?

White box testing technique involves the selection of test cases based on an analysis of the internal structure i.e., code coverage, branches coverage, paths coverage, condition coverage, and more, of a system or a component. It is also referred to as structural testing or code-based testing. The different types of white box testing are:

1. Statement Coverage
2. Decision Coverage

## **24. What is verified in white box testing?**

The white box testing following verifies the following:

1. The security holes in the code
2. The broken or incomplete paths in the code
3. The flow of structure as per the document specification
4. The expected outputs
5. All conditional loops in the code that checks the complete functionality of the application
6. Line by line coding and cover testing

## **25. What is black box testing and what are the techniques?**

Black box testing is the testing method that involves testing the software without knowing the internal structure of the program or code. This testing is usually checking the functionality of an application.

Black box testing techniques are:

1. Equivalence Partitioning
2. Boundary value analysis
3. Cause-effect graphing

## **26. What are verification and validation?**

The verification process involves evaluating software in the development phase. It helps the tester decide whether the product of a given application satisfies the specific requirements.

The validation process involves evaluating software after the development process and check whether it meets the customer requirements.

## **27. What is mutation testing?**

The technique to identify if a set of test cases is useful by intentionally introducing various bugs and retesting with original test cases to determine if the bugs are detected.

## **28. What are the categories of debugging?**

The categories for debugging are:

1. Brute force debugging
2. Backtracking
3. Cause elimination
4. Program Slicing
5. Fault tree analysis

## **Manual Testing Experienced Interview Questions**

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

User Acceptance Testing (UAT) is a process of running a product through a series of specific tests, thereby determining whether the product will meet the needs of its users.

System Testing is also known as end-to-end testing, is finding defects when the system undergoes testing as a whole.

### **30. What is testing type and what are the commonly used testing types?**

Testing type is a standard procedure that is followed to get an expected test outcome.

The commonly used testing types are:

- **Unit Testing:** Tests the smallest code of an application.
- **API Testing:** Tests the created API for the application.
- **Integration Testing:** Combines the individual software modules and tests them.
- **System Testing:** Tests the complete system.

- **Install/Uninstall Testing:** Testing done from the perspective of client/customer view
- **Agile Testing:** Testing using the Agile technique

### **31. What are the key challenges of software testing?**

Some of the challenges of software testing are:

- Availability of standard documents and resources
- to understand the application
- Shortage of skilled testers
- **Understanding the requirements:** Testers require excellent understanding and listening capabilities to be able to communicate with the customers about the conditions.
- The ability to analyze when to stop testing
- Ability to work under time constraints
- Ability to make the decisions
- Using an optimized number of test cases and testing the entire application

### **32. What are functional test cases and non-functional test cases?**

Functional testing is defined as testing the ‘functionality’ of a software or an application under review. The software is examined as per client requirements. A document called the requirement specification or software specification is used as a guide to testing the application.

### **33. What is Non-functional testing?**

When an application works as per the user’s expectation, efficiently and smoothly under any condition, then it is a reliable application. It is critical to test these parameters based on the quality; this process is called non-functional testing.

### **34. Differentiate between errors, bugs, and defects.**

A slip in coding is an error. A defect is an error spotted by a manual tester. The defect which is admitted by the development team is known as a bug. Functional failure occurs when a built code misses on the requirement.

### **35. Differentiate between quality assurance, quality control, and manual testing.**

A systematic and planned way of monitoring the quality of the process, which is followed to produce a quality product, refers to Quality Testing(QA). QA modifies the process and tracks the test reports to meet the expectations of the project.

Quality Control (QC) is about the quality of the product. QC suggests improvements apart from finding defects. Hence, QC implements the process set by QA. The testing team is responsible for quality control.

The software testing process ensures that the product which is developed by developers meets the requirements of the user. The testing aims to find bugs and to get them fixed by the development team, thereby maintaining the quality of the product to be delivered to the customer.

### **36. What are drivers and stubs in manual testing?**

Drivers and stubs are part of incremental testing. Incremental testing consists of two approaches, namely the bottom-up and top-down approach. Drivers are used in bottom-up testing, and the stub is used for a top-down approach. The stub, which is a dummy code or program, is used to test the main module.

### **37. What are the differences between test cases, test scenarios, and test scripts?**

**Test Cases:** A document that contains the steps to be executed. A test case document is already planned and contains edge cases.

**Test Scenarios:** Any functionality that can be tested is a test scenario. It is also called Test Possibility or Test Condition.

**Test Script:** It is a short-coded program used to test part of the functionality of the software system. In short, a written set of steps that should be performed manually.

### **38. What is a latent defect? How is it different from a Masked defect?**

A latent defect is also called a hidden defect. It is an existing defect in the system which might cause failure or malfunction but is not discoverable through general inspection.

A masked defect is a defect that exists but has not caused a failure yet as another defect prevents that part of the code from being executed.

### **39. What is a test deliverable?**

Test deliverables are a set of tools, documents, and components that are maintained and developed in support of testing. There are different test deliverables at every phase of the software development lifecycle:

- Before Testing
- During Testing
- After the Testing

### **40. What is a test plan and what information does it cover?**

A document that describes the approach, scope, resources, and schedule of testing activities is a test plan. The following details must be covered in the test plan:

- Test Strategy
- Test Objective
- Exit/Suspension Criteria
- Resource Planning
- Test Deliverables

### **41. Lists the risks that commonly lead to project failure.**

These are the risks that commonly lead to project failure:

- Not having enough human resource
- Testing Environment may not be set up properly
- Limited Budget

- Time Limitations

#### **42. What does a test report contain and what are the benefits of a test report?**

A test report consists of the following:

- Test Objective
- Project Information
- Defect
- Test Summary

Following are the benefits of test reports:

- Informs current status of quality of product and project
- Allows customers and stakeholders to take corrective action
- A final document decides if the product is ready for release

#### **43. What is the function of “phantom” as a manual testing tool?**

Phantom is freeware used by the Windows GUI automation scripting language. It allows the user to take control of functions and windows automatically by simulating any combination of mouse clicks and keystrokes as well as menus, lists, and more.

#### **44. What is the difference between bug leakage and bug release?**

**Bug leakage:** This is when the bug is discovered by the end-user and is missed by the testing team at the time of testing. The defect that exists in the application. It is not detected by the tester and is eventually found by the user.

**Bug release:** When a particular version of the software is released with a set of known bugs, it is called a bug release. These bugs are of low priority. Bug release is done when a software company can afford the existence of bugs in the released software but not the time/cost for fixing it in that particular version.

#### **45. What is the difference between performance testing and monkey testing?**



Performance testing checks the scalability, speed, and stability characteristics of the system. Performance is identified with achieving throughput, response time, and resource-utilization levels that meet the performance objectives for a project.

Monkey testing is a technique in software testing that allows the user to test the application and checks its behavior by providing random inputs.

#### **46. What is the difference between Test Matrix and Traceability Matrix?**

**Test Matrix:** Test matrix captures the actual quality, effort, the plan, resources and time required to capture all phases of software testing

**Traceability Matrix:** Traceability Matrix involves mapping between test cases and customer requirements.

#### **47. Group the following techniques as either static or dynamic.**

- Equivalence Partitioning.
- Use Case Testing.
- Data Flow Analysis.
- Exploratory Testing.
- Decision Testing.
- Inspections

##### **Static:**

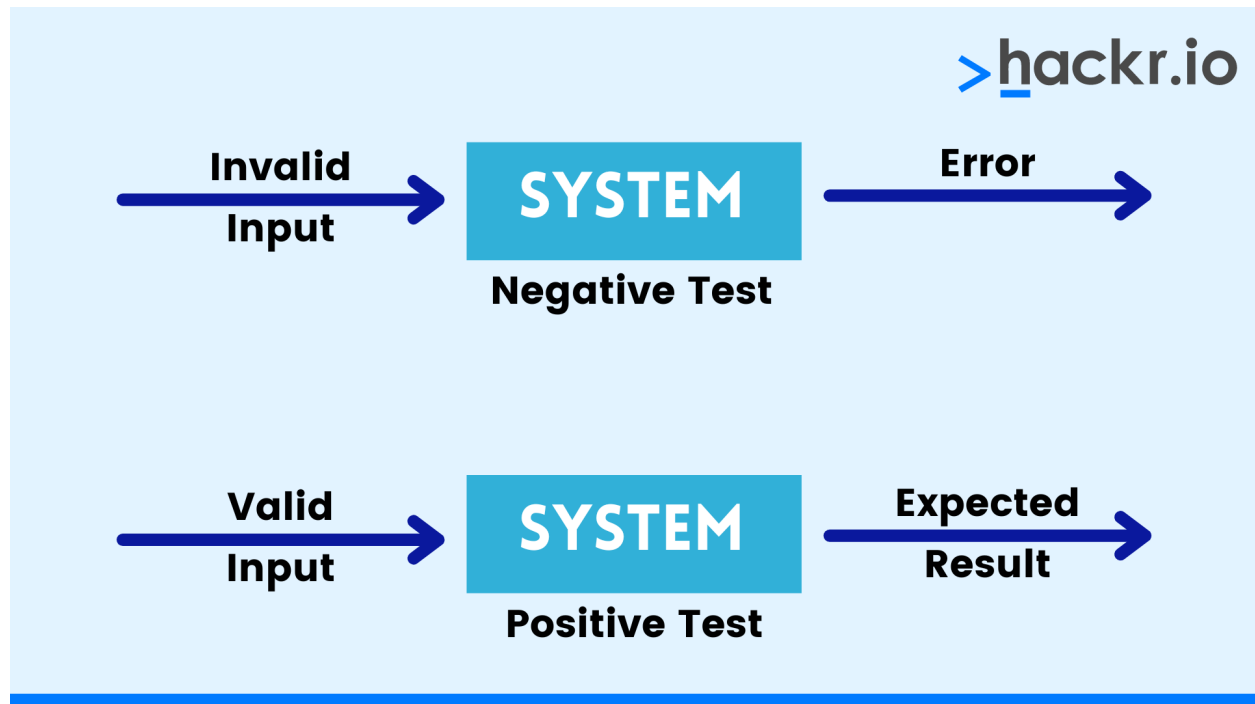
- Data Flow Analysis.
- Inspections

##### **Dynamic:**

- Equivalence Partitioning.
- Use Case Testing.
- Data Flow Analysis.
- Exploratory Testing.

- Decision Testing.
- Inspections

#### 48. What is positive and negative testing?



Positive testing is when a tester puts in a valid input and expects some action to be completed in accordance with the specification.

Negative testing is when you put in an invalid input and receive errors.

#### 49. When must testing be stopped?

Stopping the testing depends on the risks for the system under testing. There are some criteria based on which you can stop testing:

1. Deadlines (Testing, Release)
2. Test budget has been depleted
3. Bug rate fall below a certain level
4. Test cases completed with certain percentage passed
5. Alpha or beta periods for testing ends

6. Coverage of code, functionality or requirements are met to a specified point

## **50. What is equivalence partition testing?**

The Equivalence partitioning technique is when we divide the application input test data into each partition, from which we derive test cases. Using this technique reduces testing time.

## **51. What is decision-table testing and why is it important?**

Decision table testing is when the specification of testing systems take the form of cause-effect combinations or rules. A decision table lists the input in a column, with the outputs below the input in the same column. The remainder of the table explores combinations of inputs to define the outputs produced.

Decision tables are important because:

- Decision tables are helpful in test design techniques.
- It Provides a regular way of stating complex business rules which benefit the developers as well as the testers.
- Assists in the development process with the developer to do a better job. Testing with all combinations might be impractical.
- It is the most preferred choice for testing and requirements management.
- It is a structured exercise to prepare requirements when dealing with complex business rules.
- It is used in complicated model logic.

## **52. What is the main objective of the development of the manual testing system?**

Before the software can be actually used it is important that its suitability as per the requirement of the client is evaluated. In this scenario, a manual testing interview system comes into the picture. Its main aim is to manually conduct test cases without making the usage of any kind of automation tools. This developed as a result of the drawback of the automation testing which is that complete automation is impossible.

### 53. What are the advantages of Manual testing system compared to Automation testing?

The advantages of manual testing are:

- **Economical:** Manual testing is much more cost effective and small scale.
- **Human involvement:** The testing has a human working on it so it makes it more reliable and we get accurate interface feedback.
- **Saves time:** Saves time as the entire code needs not to be altered in the situation if minor quick fixes are to be made.
- **Real user experience:** With the help of manual testing, the user is able to get a good understanding of real user experience on all platforms.
- **Good capability:** Better at handling complicated use case scenarios.

### 54. What are the disadvantages of manual testing systems compared to Automation testing?

Some of the disadvantages of manual testing system are:

- **Increased chances of errors:** As the manual testing system is conducted by human beings, it is likely that an error can go unnoticed.
- **Expensive in the long run:** Although for a short period of time manual testing might appear to be cost-effective, in the long run it can be anything but.
- **Inability to conduct certain types of testing:** It is not possible to conduct load testing and performance testing through manual testing.

### 55. What are the differences between automation testing and manual testing?

The differences between manual and automation testing are:

## Manual Testing System

Manual testing cannot be conducted without human involvement.

Manual testing needs skilled labor which needs lots of time as well as high cost.

Different sorts of applications can be tested manually. Certain testing types like ad-hoc and monkey testing are more appropriate for manual execution.

Manual testing can be routine and tedious.

## Automation Testing System

Automation testing makes use of various kinds of tools to carry out the test cases.

Automation testing doesn't involve much labor, time and cost.

Automated testing is suitable only for stable systems and is generally used for Regression Testing.

The monotonous part of executing the same test cases repetitively is conducted by automation software in Automation Testing.

### 56. What is your understanding of End To End Testing?

End to End testing system is a way of testing an application to ensure whether it is working as expected. It conducts a test from the beginning of the application to the end. This testing confirms that the data integrity is maintained in between the different system components and the systems.

### 57. What is your understanding of System and Integration Testing?

The main objective of conducting system testing is to make sure that the system works as per the specified requirements.

The major objective of the integration testing is to test the interface in between the two different software units.

### **58. What is the Big Bang Approach?**

The Big Bang Approach is a method of integration testing which necessitates a side-by-side checking of all the system components. Its main advantage is that the tester can check the work of the entire system and its components.

### **59. What is Sanity Testing?**

Sanity Testing is conducted throughout the release phase to determine the main functionalities of the application without going into many details. Sanity checking is carried out at the release level.

### **60. What is Alpha, Beta and Gamma Testing?**

Alpha testing is done by the developers who develop the software and testers.

Sometimes it can be observed that the alpha testing is conducted by the purchaser or outsourcing team with the existence of developers or testers.

Beta testing is to be conducted by a specified number of ultimate users before delivery. Generally, it is conducted at the place of the end-user.

Gamma testing is performed to check the specified needs when the software is ready to be released. It is usually done at the place of the ultimate user. It is performed at the first hand by omitting all the in-house testing activities.

## **Conclusion**

That's it for our manual testing interview questions. These questions are those most frequently asked. You should make sure you know them thoroughly, and also put the concepts into practice.

Good luck!

Want to keep leveling up your interview skills?

[Check out the Top 50 Software Testing Interview Questions](#)

## **Bonus Tips**

In addition to these theory-based questions, you will also face coding tests.

Continue practicing programming because there's no doubt that they will be part of the manual testing interview questions.