# Manual Testing Questions For Freshers

## 1. Explain what is software testing.

It is the process of analyzing any given piece of software to determine if it meets shareholders' needs as well as detecting defects, and ascertaining the item's overall quality by measuring its performance, features, quality, utility, and completeness. Bottom line, it's quality control.

## 2. What is quality control, and how does it differ from quality assurance?

Quality control is the process of running a program to determine if it has any defects, as well as making sure that the software meets all of the requirements put forth by the stakeholders. Quality assurance is a process-oriented approach that focuses on making sure that the methods, techniques, and processes used to create quality deliverables are applied correctly.

## 3. What exactly is manual software testing, and how does it differ from automated software testing?

Manual software testing is a process where human testers manually run test cases, then generate the resulting test reports. With automation software testing, these functions are executed by automation tools such as test scripts and code. The tester takes the end user's role to determine how well the app works.

## 4. What are the advantages of manual testing?

Manual testing's strengths are:

- It's cheaper
- You get visual feedback that's accurate and quick

- It’s ideal for testing minor changes
- It’s perfect for ad hoc testing
- Testers don’t have to know anything about automation tools
- It’s great for testing UI’s

## 5. On the other hand, what are the drawbacks to manual testing?

Manual testing’s weaknesses are:

- Susceptible to human error
- Some tasks may be difficult to accomplish manually, requiring more time to complete
- The cost adds up, so it’s more expensive in the long run
- You cannot record the manual testing process, so it’s hard to replicate it

## 6. What kind of skills are needed for someone to become a software tester?

Software testers need skills such as:

- Problem-solving skills
- Excellent written and verbal communication skills
- Detail-oriented
- Able to handle the pressure
- Can work solo or as a team member equally well
- Organizational skills

- Related technical skills

## 7. Explain what is SDLC.

This is an acronym for Software Development Life Cycle and encompasses all of the stages of software development, including requirement gathering and analysis, designing, coding, testing, deployment, and maintenance.

## 8. What is a test case?

Test case is used to check whether an application complies with its requirements. It is a documented set of circumstances including prerequisites, input values, and expected outcomes.

## 9. What is a test scenario?

A test scenario is derived from a use case. It's used to test an application's feature from beginning to end. Multiple test cases can be accommodated by a single test scenario. When there is a time constraint during testing, scenario testing comes in handy.

## 10. What is a test plan?

A test plan is a formal document that specifies the scope of testing, the method to be used, the resources needed, and the estimated time to complete the testing process. It is derived from the specifications (Software Requirement Specifications).

## 11. What is test data?

Test data is information that is used to test software with various inputs and determine whether the resulting output matches the intended result. This data is generated based on the needs of the company.

## 12. What is a test script?

An automated test case created in any programming or scripting language is known as a test script. These are essentially a collection of instructions for evaluating an application's functionality.

## 13. What types of manual testing are there? Break them down.

Manual testing is broken down into:

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

## 14. What is black box testing, and what are the various techniques?

Software testers employ black-box testing when they do not know the internal architecture or code structure. The techniques are:

- Equivalence Partitioning
- Boundary value analysis
- Cause-effect graphing

## 15. What is white box testing and its various techniques?

Unlike black-box testing, white box involves analyzing the system's internal architecture and/or its implementation, in addition to its source code quality. It's techniques are:

- Statement Coverage
- Decision Coverage

So far, if you have any doubts about these Manual testing interview questions/ software testing interview questions, please ask in the comment section below.

## 16. Explain the difference between alpha testing and beta testing.

Alpha testing is at the developer's site before release. Potential clients conduct beta testing at their websites.

## 17. What's the difference between verification and validation?

Verification evaluates the software at the development phase, ascertaining whether or not a product meets the expected requirements. On the other hand, validation evaluates the software after the development phase, making it sure it meets the requirements of the customer.

## 18. What's a testbed?

It's not furniture. A testbed is an environment used for testing an application, including the hardware as well as any software needed to run the program to be tested.

## 19. What is Sanity testing?

Sanity testing is testing done at the release level to test the main functionalities. It's also considered an aspect of regression testing.

Got a question for us? Please mention it in the comments section on this Manual Testing Interview Questions article and we will get back to you.

## 20. When should developers implement configuration management procedures?

This should be done during test planning.

## 21. List the four different test levels

The four levels are:

- Unit/component/program/module testing
- Integration testing
- System testing
- Acceptance testing

## 22. What's the difference between a bug and a defect?

A bug is a fault in the software that's detected during testing time, while a defect is a variance between expected results and actual results, detected by the developer after the product goes live.

## 23. What about the difference between an error and a failure?

If a program can't run or be compiled during development, it's an error. If an end-user discovers an issue with the software, it's a failure.

## 24. What's GUI testing?

This tests the interface between the software and the end-user. Short for Graphics User Interface.

## 25. When should testing end?

There are a few criteria for ending testing:

- The bug rate has fallen below an agreed-upon level
- The testing or release deadlines have arrived
- The testing budget is out of funds
- A certain percentage of test cases have passed
- The alpha or beta testing periods have ended
- Code, functionality, or requirements coverage have been met at a declared point

These were some basic manual testing interview questions. In the coming section, we bring to you some advanced level manual testing interview questions.

## Advanced Level Manual Testing Interview Questions

## 26. What are the different types of Software testing?

Software testing is classified into two main categories.

1. Functional testing
2. Non-Functional testing

## 27. Explain Functional Testing

Functional testing is a type of black-box testing. It focuses on the software's functional requirements rather than its internal implementation. A functional requirement refers to the system's needed behaviour in terms of input and output.

It checks the software against the functional requirements or specification, ignoring non-functional characteristics like performance, usability, and dependability.

The purpose of functional testing is to ensure that the software up to snuff in terms of functionality and to solve the difficulties of its target users.

Some of the types of functional Testing are -

- Unit Testing
- Integration Testing
- Regression Testing
- System Testing
- Smoke Testing
- Performance Testing
- Stress Testing

## 28. Explain Non functional testing

Non-functional testing examines the system's non-functional requirements, which are characteristics or qualities of the system that the client has specifically requested. Performance, security, scalability, and usability are among them.

Functional testing is followed by non-functional testing. It examines aspects that are unrelated to the software's functional requirements. Non-functional testing assures that the programme is safe, scalable, and fast, and that it will not crash under excessive pressure.

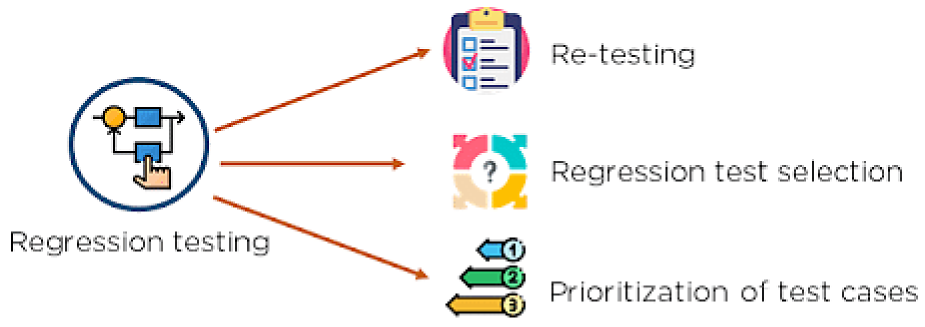
## 29. Mention a few advantages of Automated testing.

The following are some major advantages of automated testing -

- Automated test execution is quick and saves a significant amount of time.
- Human mistakes are eliminated during testing when test scripts are carefully prepared.
- CI tools like Jenkins, which may also be set to distribute daily test results to key stakeholders, can be used to schedule test execution for a nightly run.
- Automation testing uses a lot less resources. Test execution requires nearly no time from QAs once the tests have been automated. QA bandwidth can be used for other exploratory work.

## 30. What is Regression Testing?

Regression Testing is a full or partial selection of already executed test cases that are re-executed to ensure existing functionalities work fine.



Steps involved are -

1. Re-testing: All of the tests in the current test suite are run again. It turns out to be both pricey and time-consuming.
2. Regression tests are divided into three categories: feature tests, integration tests, and end-to-end testing. Some of the tests are chosen in this step.
3. Prioritization of test cases: The test cases are ranked according to their business impact and important functionalities.

## 31. What is Test Harness?

A test harness is a collection of software and test data used to put a programme unit to the test by running it under various conditions such as stress, load, and data-driven data while monitoring its behaviour and outputs.

## 32. Differentiate between Positive and Negative Testing

| Positive Testing | Negative Testing |
|---|---|
| Positive testing ensures that your software performs as expected. The test fails if an error occurs during positive testing. | Negative testing guarantees th your app can gracefully deal w unexpected user behaviour o incorrect input. |
| In this testing, the tester always looks for a single set of valid data. | Testers use as much ingenuit as possible when validating th app against erroneous data. |

## 33. What is a Critical Bug?

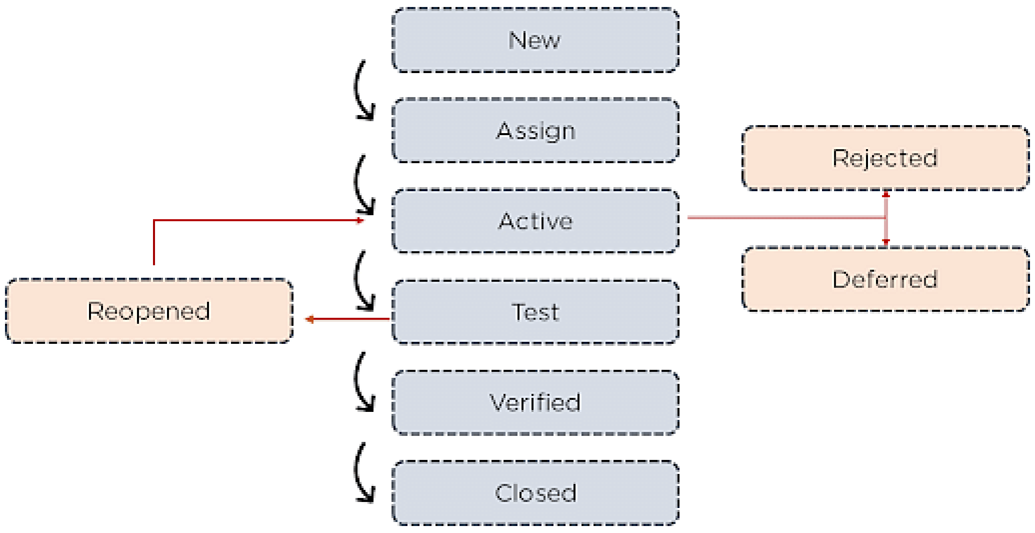
A critical bug is one that has the potential to affect the bulk of an application's functioning. It indicates that a significant portion of functionality or a critical system component is utterly broken, with no way to proceed. The application cannot be delivered to end users until the critical bug has been fixed.

## 34. What is Test Closure?

Test Closure is a document that summarises all of the tests performed throughout the software development life cycle, as well as a full analysis of the defects fixed and errors discovered. The total number of experiments, the total number of experiments executed, the total number of flaws detected, the total number of defects settled, the total number of bugs not settled, the total number of bugs rejected, and so on are all included in this memo.

## 35. Explain the defect life cycle.

A defect life cycle is a process by which a defect progresses through numerous stages over the course of its existence. The cycle begins when a fault is discovered and concludes when the defect is closed after it has been verified that it will not be recreated.



## 36. What is the pesticide paradox? How to overcome it?

According to the pesticide paradox, if the same tests are done repeatedly, the same test cases will eventually stop finding new bugs. Developers will be especially cautious in regions where testers discovered more flaws, and they may overlookPositive and Negative Testing?

other areas. Methods for avoiding the pesticide conundrum include:

- To create a completely new set of test cases to put various aspects of the software to the test.
- To create new test cases and incorporate them into existing test cases.

It is possible to detect more flaws in areas where defect levels have decreased using these methods.

## 37. What is API testing?

API testing is a sort of software testing that entails evaluating application programming interfaces (APIs) to see if they meet functionality, reliability, performance, and security requirements. Simply put, API testing is designed to detect defects, inconsistencies, or departures from an API's expected behaviour. Typically, applications are divided into three layers:

The user interface is also known as the presentation layer.

For business logical processing, the Business Layer or application user interface is used.

API testing is done at the most vital and important layer of software architecture, the Business Layer, for modelling and manipulating data.

## 38. What is System testing?

System testing is a type of testing in which the entire software is tested. System testing examines the application's compliance with its business requirements.

## 39. What is Acceptance testing?

Acceptance testing is a type of testing done by a possible end-user or customer to see if the software meets the business requirements and can be used.

## 40. Differentiate between bug leakage and bug release

Bug Leakage - When tested software is pushed into the market and the end-user discovers defects, this is known as bug leakage. These are bugs that the testing team overlooked throughout the testing phase.

Bug Release - When a certain version of software is launched into the market with some known bugs that are expected to be fixed in later versions, this is known as a bug release. These are low-priority issues that are highlighted in the release notes when sharing with end-users.

## 41. What do you mean by Defect Triage?

Defect triage is a procedure in which defects are prioritised depending on a variety of characteristics such as severity, risk, and the amount of time it will take to fix the fault. The defect triage meeting brings together several stakeholders - the development team, testing team, project manager, BAs, and so on – to determine the order in which defects should be fixed.

## 42. What is Integration Testing? What are its types?

Integration testing is performed after unit testing. We test a group of linked modules in integration testing. Its goal is to identify faults with module interaction.

The following are the types of integration testing -

- Big Bang Integration Testing — After all of the modules have been merged, big bang integration testing begins.
- Top-down Integration Testing — In top-down integration, testing and integration begin at the top and work their way down.
- Bottom-up Integration Testing — In bottom-up integration testing, lower-level modules are tested before moving up the hierarchy to higher-level modules.
- Hybrid Integration Testing — Hybrid integration testing combines top-down and bottom-up integration testing techniques. The integration with this approach starts at the middle layer, and testing is done in both directions.

## 43. What is a stub?

Many times, when top-down integration testing is performed, lower-level modules are not produced until top-level modules are tested and integrated. Stubs or dummy modules are used in these circumstances to emulate module behaviour by delivering a hard-coded or predicted result based on the input variables.

## 44. What is code coverage?

The quantity of code covered by the test scripts is referred to as code coverage. It conveys the scope of the test suite's coverage of the application.

## 45. What is a cause-effect graph?

A cause-effect graph testing technique is a black-box test design technique that uses a graphical representation of the input (cause) and output (effect) to construct the test. This method employs a variety of notations to describe AND, OR, NOT, and other relationships between the input and output conditions.

## 46. Explain equivalence class partitioning.

Equivalence class partitioning is a black-box testing technique based on specifications. A set of input data that defines multiple test conditions is partitioned into logically comparable groups in equivalence class partitioning, so that utilising even a single test data from the group for testing can be considered as similar to using all the other data in that group.

## 47. What is boundary value analysis?

The border values of the classes of the equivalence class partitioning are used as input to the test cases in boundary value analysis, which is a software testing technique for designing test cases.

## 48. What is your approach towards a severely buggy program? How would you handle it?

In such cases, the best course of action is for testers to go through the process of reporting any flaws or blocking-type issues that arise, with an emphasis on critical bugs. Because this sort of crisis might result in serious issues such as insufficient unit or integration testing, poor design, wrong build or release methods, and so on, management should be contacted and given documentation as proof of the problem.

## 49. What if an organization's growth is so rapid that standard testing procedures are no longer feasible? What should you do in such a situation?

This is a very prevalent issue in the software industry, especially with the new technologies that are being used in product development. In this case, there is no simple answer; however, you could:

- Hire people who are good at what they do.
- Quality issues should be ‘fiercely prioritised' by management, with a constant focus on the client.
- Everyone in the company should understand what the term "quality" implies to the end-user.

## 50. When can you say for sure that the code has met its specifications?

Most businesses have coding "standards" that all developers are expected to follow, but everyone has their own opinion on what is best, as well as how many regulations are too many or too few. There are many methods available, such as a traceability matrix, to guarantee that requirements are linked to test cases. And when all of the test cases pass, that means the code satisfies the requirement.