Manual testing is key to the software development process, as it helps identify usability and interface issues that automated tests might miss. Top companies like Uber, Google, Netflix, and Amazon use it to ensure a smooth user experience.

In this interview preparation guide, we provide you with the top 50 Manual Testing interview questions for both beginners and experienced professionals. You’ll find interview questions on **core manual testing concepts**, **test case development**, **bug tracking**, **usability testing**, **regression testing**, and more to help you succeed in your manual testing interviews.

**Table of Content**

* [Manual Testing Interview Questions for Freshers](https://www.geeksforgeeks.org/manual-testing-interview-questions/#manual-testing-interview-questions-for-freshers)
* [Manual Testing Questions for Intermediate](https://www.geeksforgeeks.org/manual-testing-interview-questions/#manual-testing-questions-for-intermediate)
* [Manual Testing Interview Questions for Experienced](https://www.geeksforgeeks.org/manual-testing-interview-questions/#manual-testing-interview-questions-for-experienced)
* [Real-Time Interview Questions on Manual Testing](https://www.geeksforgeeks.org/manual-testing-interview-questions/#realtime-interview-questions-on-manual-testing)

**Manual Testing Interview Questions for Freshers**

In this section, we have compiled the most frequently asked manual software testing questions for freshers. So, if you are a newbie in the [software testing](https://www.geeksforgeeks.org/software-testing-tutorial) world, then explore this section to know what types of questions you will face during the software tester job profile

**1. What is Manual Software Testing?**

Manual Testing is a type of software testing process where test cases are executed manually without using any automated tool.

* The tester verifies the software functionality manually.
* The tester has a list of all the test cases that need to be manually tested. They go through each test case and manually verify the output.
* It is inefficient in comparison to automated testing, slow, and not repeatable in a consistent manner.
* Manual testing is prone to human misjudgment.

**2. What are the Advantages of Manual Testing?**

* **Better for Short-Lifecycle Projects:**Manual testing is better for projects with short life cycles.
* **Time and cost-efficiency:**It is better to opt for manual testing for small, easy projects to save time, money, and resources.
* **A good** product-oriented**option for GUI testing:**Manual testing can be done accurately for GUI Testing.
* **Easy to learn:**It is easy to learn for new testers.
* **Uses human intelligence:**Manual testing allows testers to use their higher cognitive abilities to detect errors. This helps them to find errors that may be missed during automated testing.
* **Detect errors outside the code:**Manual testing helps testers to locate bugs that do not affect the code such as server response time.

**3. What are the Drawbacks of Manual Testing?**

* **Time-consuming:**Manual testing requires time as the tester executes test cases manually and testing complex programs could take some time.
* **Human error:**Manual testing is prone to human error. By making mistakes when executing test cases, testers may come up with erroneous results.
* **Difficult to measure:**It is difficult to assess the efficiency of the manual testing process as it is difficult to keep track of the quality of the test cases executed, errors discovered, and test coverage attained.
* **Costly:**Manual testing can be expensive, particularly for big, complex projects, or when frequent releases are required.

**4. List key challenges of Manual Testing.**

Below are some of the key challenges of manual testing:

* **Lack of standard documentation:**Lack of standard documentation to understand the application fully to have a better insight into the application makes it difficult for the tester to create test cases for manual testing efficiently.
* **Unavailability of skilled testers:**Manual testing depends on Human intelligence, and analytical skills to design test cases that can ensure the best coverage.
* **Deciding test cases order:**It is very important to prioritize, and categorize the test cases and decide the order in which to execute the test cases.
* **Ability to know when to stop testing: A tester needs to have product-oriented** knowledge and the ability to decide when to stop testing the application to save time and effort.

**5. How Manual Testing is different from Automated Testing?**

Manual testing is a software testing process in which a tester tests each test case one by one in an individual manner. Whereas in automated testing, the tester utilizes tools and scripts to automate testing efforts.

| **Parameters** | **Manual Testing** | **Automated Testing** |
| --- | --- | --- |
| **Definition** | In manual testing, the tester executes test cases manually. | In automated testing, the tester uses tools and scripts to automate the process of testing. |
| **Working** | Manual testing works by requiring analysts and QA engineers to be involved in everything right from test case creation to actual test execution. | Automated testing involves testers writing test scripts that automate test case execution. |
| **Strength** | Manual testing is better at handling complex scenarios. | Automated testing is much faster and covers many permutations. |
| **Weakness** | Manual testing is slow and tedious. | Automated testing requires coding and test maintenance. |
| **Test coverage** | It isn’t easy to ensure sufficient test coverage. | It is easy to ensure sufficient test coverage. |

**6. Who is a Manual Tester?**

The manual tester is a professional who is responsible for conducting quality checks on the software applications without using [automation tools](https://www.geeksforgeeks.org/automation-tools-for-testing-android-applications). They are responsible for manually checking the software for errors and fixing them. They must have appropriate skills to be able to meet the company’s requirements.

**7. What is the role of documentation in Manual Testing?**

Documentation plays a vital role in manual testing. It is important to document all steps taken in the testing process to ensure sufficient test coverage and accurate results. It provides an audit trail, which can be used to evaluate past test results and identify areas of improvement. It also serves as a reference for other testers who may be unfamiliar with the system.

**8. What are the Priority and Severity in Software Testing?**

Priority is the order in which the developer should resolve the defect, on the other hand, Severity is the degree of impact that defect has on the operation of the product.

* Priority indicates how soon the bug is fixed and severity shows the seriousness of the defect on the product functionality.
* Priority is driven by the business values and Severity is driven by the functionality of the product.

**9. What is Test Harness?**

Test Harness is a collection of stubs, drivers, and other supporting tools that are required to automate the test execution. It executes tests using a test library and generates test reports.

* It helps automate the testing procedures and thus increases the productivity of the system through automation.
* A test harness is a collection of numerous things to test software and report its results.
* It helps developers to measure the cove coverage at a code level.

**10. What is a Test Bed?**

Test Bed is a test execution environment that is configured for testing. It consists of specific hardware, software, [operating system](https://www.geeksforgeeks.org/operating-systems), network configuration, other system software, and application software.

**11. What is test data?**

Test data is data that is used by test cases to determine if the software working correctly or not. It is collected into a document known as a test data document so that testers can easily access it when they run their tests.

**12. What is Quality Control(QC) in testing?**

Quality Control is a set of methods that are used by the organization to ensure the quality of software by identifying defects and correcting defects in the developed software.

* It provides the identification of defects.
* It is product-oriented.
* It is a reactive process.
* The testing team is responsible for Quality Control.

**13. What is Test Closure?**

Test closure is a document that provides a summary of all the tests covered during the [software development lifecycle](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc).

* It includes various activities like test completion reporting, a summary of test results as well and the test completion matrix.
* It gives us an outline of the tests conducted during the software testing and details of the errors and bugs found and resolved during the testing phase.

**14. What is Random Testing?**

Random testing also known as Monkey Testing is a type of software testing in which the system is tested with the help of generating random and independent inputs and test cases.

* It is performed where the defects are not identified at regular intervals.
* It saves time and effort than actual test efforts.

**15. What is Defect Cascading?**

Defect cascading in software testing means when one defect leads to the discovery of other defects. It often occurs because the original defect was not fixed properly. This cascading causes a chain reaction of errors, making it difficult to source of the problem.

* It can lead to many issues like minor performance slowdowns, system crashes, etc making it a severe risk to developers and testers.
* Understanding defect cascading can help to prevent them from happening in their process.

**16. What is a Test Driver?**

Test drivers are used in Bottom-up integration testing to simulate the behavior of the upper-level modules that are not yet integrated.

* They act as temporary replacements for a calling module.
* They give the same output as that of the actual product.
* They are used when the software needs to interact with an external system and are usually more complex than stubs.

**17. What is a Stub?**

Stubs are used in Top-Down integration testing thus increasing the efficiency of the testing process.

* They are developed by software developers to use them in place of modules, if the respective modules are not developed, missing in the developing stage, or currently unavailable.
* It simulates a module that has all the capabilities of the unavailable module.

**18. What is Defect Triage?**

Defect triage is a procedure that involves detecting and prioritizing problems, allocating them to development, and tracking them.

* The goal is to evaluate, prioritize, and assign the resolution of defects.
* It is also known as bug triage.
* It is based on the severity and priority of the defects in software.

**19. What is API Testing?**

[API testing](https://www.geeksforgeeks.org/api-testing-software-testing) is a type of software testing that validates APIs. It aims to check the functionality, reliability, performance, and security of the programming interfaces.

* It verifies that the API returns the correct response for different input values.
* It ensures that the different components of a system can communicate with each other correctly and that the system can handle a large volume of requests.

**20. What is Alpha Testing?**

Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or the public. It is one of the user acceptance tests that is done near the end of the development of the software.

* It is generally performed by the homestead software engineers or quality assurance staff.
* It is used to identify and resolve critical bugs and issues in the software before it is released to the public.
* It is performed in a controlled environment like a lab or a test network and is used to simulate real-world use cases.

**21. What is Beta Testing?**

Beta testing is the process of testing a software product or service in a real-world environment before its official release. It helps identify bugs and errors that may have been missed during the development process.

* During beta testing, the software is made available to a selected group of users who are willing to test the product and provide feedback to the developers.
* The beta testers typically use the software in various ways, attempting to find any issues, bugs, or usability problems.
* They then provide feedback on their experience, reporting any problems or issues encountered.

**22. What are the types of Manual Testing?**

Manual testing is of the following different types:

1. **Black box testing:**Black box testing is a software testing method that focuses on testing the functionality of the software without dealing with the internal structures or workings.
2. **White box testing:**White box testing is a software testing method that tests the internal structure of the application as opposed to its functionality.
3. **Unit testing:**Unit testing is a software testing method in which the smallest testable parts of the application called units are tested for proper operation.
4. **System testing:**System testing is a software testing method in which the QA team evaluates how the various components of an application interact together in a fully integrated system.
5. **Integration testing:**Integration testing is a type of software testing method in which different modules or units of a software application are tested as a combined entity.
6. **Acceptance testing:**Acceptance testing is a type of software testing method that is performed after system testing and before making the system available for actual use.

**23. List the roles and responsibilities of Manual testers.**

Below are some of the roles and responsibilities of Manual Tester:

1. Analyzing client requirements.
2. Reviewing written code for compliance with project specifications and requirements.
3. Creating a test environment for executing the test cases.
4. Organizing and conducting review meetings.
5. Detecting and fixing bugs.
6. Monitor system errors and discuss them with the team.

**24. Describe Manual Testing Process.**

Manual testing is a process of identifying bugs and errors in software without the use of automated tools. Below is the process of manual testing:

1. **Identify the scope of testing:**The first step is to identify the scope of testing and the scope can vary from a functionality to an end-to-end system.
2. **Design test cases:**The next step is to design test cases including test scenarios, data, expected results, and all other details that are necessary to perform the tests on the identified scope.
3. **Execute test cases:**After designing the test cases, testers execute the test cases to find the discrepancies between the actual result and the expected result.
4. **Record results:**Testers should record the results for further analysis.

**25. What are the different levels of Manual Testing?**

Different levels of Manual Testing are:

1. **Unit Testing:**Unit testing is a process of testing the individual pieces of code within the system called units. The main focus is on the functional accuracy of the standalone modules.
2. **Integration Testing:**Integration testing involves combining and testing the individual units to see if they work together as expected or not.
3. **System Testing:**System testing involves testing all the components of the product as a whole to ensure that overall product requirements are met or not.
4. **User Acceptance Testing:**User Acceptance Testing is the final step in the testing process that determines if the software is ready for release or not.

**26. What are the Skills required for Manual Testing?**

Some of the important skills required for manual testing are:

* A strong analytical ability.
* Ability to report test results professionally.
* Familiarity with agile methodologies.
* Ability to write test cases for manual testing.
* Knowledge of concepts required for manual testing like SDLC, [STLC](https://www.geeksforgeeks.org/software-testing-life-cycle-stlc), [SQL](https://www.geeksforgeeks.org/sql-tutorial), etc.
* Understanding of manual testing tools like [JIRA](https://www.geeksforgeeks.org/jira-tutorial), JMeter, etc.
* Understanding of test management tools and test tracking tools.

**27. When to use Manual Testing over Automation Testing?**

There are many scenarios when manual testing can be opted over automation testing in a project:

* **Adhoc Testing:**Adhoc testing can be achieved using manual testing as there is not no specific approach and is performed without planning and documentation. In ad hoc testing the understanding and insight of the tester plays an important role.
* **Exploratory test:**Exploratory testing depends upon the tester’s knowledge, experience, and logical skills, so human involvement is a must here and manual testing is the best choice for exploratory tests.
* **Usability testing:**In Usability testing human observation plays a very important role as it is measured by the tester how user-friendly, and efficient the software is for the end users. So manual testing is an appropriate choice for usability testing.
* **Short lifecycle projects:**Automation testing is not appropriate for short-term projects as it requires high investment and planning and manual testing on the other hand aims to save time and resources.

**28. What are Manual Testing Tools?**

Manual testing is a software testing method in which the tester manually executes the test cases without the use of automated technologies. The tools that help in this process are known as manual testing tools. Some examples of manual testing tools are Bugzilla, JMeter, JIRA, etc.

**29. List some Manual Testing Tools.**

Below are some of the commonly used Manual Testing tools:

1. **Trac:**Trac is one of the most powerful manual testing tools that is developed in Python and is a web-based program.It is compatible with a variety of databases like [SQLite](https://www.geeksforgeeks.org/introduction-to-sqlite), [MySQL](https://www.geeksforgeeks.org/mysql-tutorial), MS-SQL, etc.
2. **TestLink:**TestLink is a high-quality product that has more functions in a comparable package. It is simple to use as the program is available to use through a browser.
3. **JMeter:**JMeter is an open-source tool for performance testing of static and dynamic resources and dynamic web applications. It has an easy-to-use and clear interface accepting JVMs from Windows, Mac OS X, Linux, and other platforms.
4. **Bugzilla:**Bugzilla is an open-source application that helps customers and clients to keep track of issues. It has a simple-to-useGoogle-style bug search that also searches the complete text of a bug.
5. **Load Runner:**It is the most commonly used performance testing tool that is used to categorize the most prevalent causes of performance problems rapidly. It is compatible with a variety of development tools and protocol stacks and it helps to lower the cost of distributed load testing.

**30. What are the best practices for writing test cases for Manual Testing?**

Below are some of the best practices that can be followed for writing the test cases:

* **Prioritize test cases:**Prioritize which test cases to build based on the application’s risk considerations and project timeframes.
* **Follow the 80/20 rule:**To ensure sufficient coverage, it is better to have 20% of the test cases cover 80% of the application.
* **Categorize test cases:**List all the test cases and categorize the test cases according to business scenarios and functionality.
* **Design modular test cases:**Make sure that the test cases are modular and as detailed as possible.
* **Remove duplicate test cases:**Remove irrelevant and duplicate test cases.

**31. Can Automation Testing replace Manual Testing?**

Automation testing cannot completely replace Manual testing as it is not possible to automate everything. Manual testing can be used in situations where automation isn’t possible. Both automated and manual testing have their advantages and disadvantages.

**32. List the differences between the Test Case and Test Scenario.**

| **Parameters** | **Test Case** | **Test Scenario** |
| --- | --- | --- |
| **Definition** | A test case is a detailed document that provides details about the testing strategy, testing process, preconditions, and expected output. | A test scenario gives one-line information about what to test and is derived from the use case. |
| **Action Level** | These are low-level actions. | These are high-level actions. |
| **Objective** | The purpose is to verify the test scenario by implementing the steps. | The purpose of the test scenario is to cover the end-to-end functionality of software functionality. |
| **Time Consumption** | It takes more time. | It takes less time. |
| **Input** | It can be obtained from test scenarios. | It can be obtained from the use case. |

**33. What is Smoke Testing?**

[Smoke testing](https://www.geeksforgeeks.org/smoke-testing-software-testing) also known as Build Verification Testing is a software testing method that is performed at the beginning of the development process to make sure that the most critical functions of the software applications are working correctly.

* It is done to quickly identify and fix the major issues before more detailed testing is performed.
* The goal is to determine whether the build is stable enough to proceed with further testing.

**34. What is Regression Testing?**

Regression testing is a software testing process of testing previously tested programs to ensure that the defects have not been introduced or uncovered in unchanged areas of the software as a result of the changes made in the software.

**35. List the differences between Sanity Testing and Smoke Testing.**

| **Parameters** | **Sanity Testing** | **Smoke Testing** |
| --- | --- | --- |
| **Definition** | Sanity testing is performed to check whether the new functionality/ bug has been fixed. | Smoke testing is performed to make sure that the critical functionality of the system is working correctly. |
| **Purpose** | The goal of sanity testing is to verify rationality. | The goal of smoke testing is to verify stability. |
| **Documentation** | Sanity testing isn’t documented. | Smoke testing is documented. |
| **Who performs** | Testers perform sanity testing. | Software developers or testers perform smoke testing. |
| **Software build stability** | The software build is relatively stable at the time of sanity testing. | The software build may be either stable or unstable during smoke testing. |

**36. What is Top-Down Integration Testing?**

Top-down integration Testing is an Integration testing technique in which testing is done by integrating two or more modules by moving down from top to bottom through the control flow of the architecture structure.

* High-level modules are tested first and then the low-level modules are tested.
* Stubs are the modules that act as temporary replacements for the called module.

**37. List the differences between Regression and Retesting.**

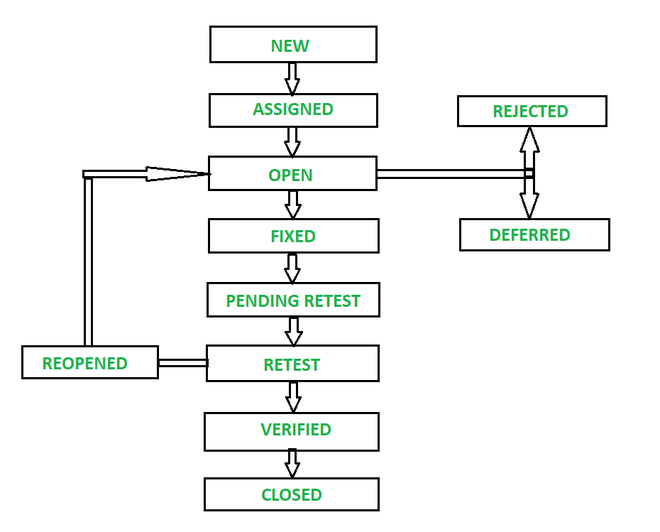
| **Parameters** | **Regression Testing** | **Retesting** |
| --- | --- | --- |
| **Definition** | Regression testing is done to ensure that the changes have not affected the unchanged part of the product. | Retesting is done to ensure that the test cases which failed in the last execution are fixed. |
| **Purpose** | The purpose of regression testing is to check that the new code changes should not have any side effects on the existing functionalities. | The purpose of retesting is to check whether the functionality has been restored following a bug fix. |
| **Is Automation possible?** | Automating regression testing is possible as Manual testing can be time-consuming and expensive. | Automating test cases for retesting is not possible. |
| **Test cases considered** | Regression testing is done for passed test cases. | Retesting is done for failed test cases. |
| **Defect Verification** | Defect Verification is not part of regression testing. | Defect Verification is part of retesting. |

**38. List some Test Management Tools.**

1. **QACoverage:**QACoverage is a test management tool that is cost-effective, boosts test productivity, and provides visibility to better handle the QA process. It provides the ability to upload 1000 requirements and test cases from Excel spreadsheets in seconds and supports complete traceability between requirements, test cases, and defects.
2. **TestRail:**TestRail is a web-based test case management tool that helps teams organize testing efforts and get real-time insights into testing activity. It helps to capture details about test cases with screenshots and expected results. It is possible to compare results across multiple test runs, configurations, and milestones.
3. **SpiraTest:**SpiraTest is a test management tool from Infectra that helps agile teams deliver high-quality software faster and with greater confidence. It helps users manage all their tests, requirements, and bugs in one place. It allows for easy importing of data from many modern applications.
4. **Testiny:**Testiny is a test management tool that aims to make manual testing and QA management as seamless as possible. It helps testers perform tests without adding bulky overhead to the testing process.
5. **TestMonitor:**TestMonitor is an end-to-end test management tool that supports advanced test case design capable of supporting thousands of cases. It supports comprehensive result tracking and smart reporting with many filter and visualization options.

**39. What is Bug Lifecycle?**

Bug lifecycle also known as Defect Life Cycle is the life cycle of a defect or bug from which it goes through covering a specific set of states in its entire life. The below diagram illustrates the actual workflow of the Defect Life Cycle:



The above diagram shows different states of Defect in the Defect Life Cycle and these are as follows :

1. **New:**When any new defect is identified by the tester, it falls into a ‘New’ state. It is the first state of the Bug Life Cycle.
2. **Assigned:**Defects that are in the status of ‘New’ will be approved and that newly identified defect will be assigned to the development team to work on the defect and resolve that.
3. **Open:** In this ‘Open’ state the defect is being addressed by the developer team and the developer team works on the defect for fixing the bug. Based on some specific reason if the developer team feels that the defect is not appropriate then it is transferred to either the ‘Rejected’ or Deferred’ state.
4. **Fixed:** After necessary changes of codes or after fixing the identified bug developer team marks the state as fixed.
5. **Pending Retest:**During the fixing of the defect is completed, the developer team passes new code to the testing team for a retest. The code/application is pending for retesting at the Tester side so the status is assigned as ‘Pending Retest’.
6. **Retest:** At this stage, the tester starts work of retesting the defect to check whether the defect is fixed by the developer or not, and the status is marked as ‘Retesting’.
7. **Reopen:**After ‘Retesting’ if the tester team finds that the bug continues like previously even after the developer team has fixed the bug, then the status of the bug is again changed to ‘Reopened’. Once again bug goes to the ‘Open’ state and goes through the life cycle again. This means it goes for Re-fixing by the developer team.
8. **Verified:**The tester re-tests the bug after it got fixed by the developer team and if the tester does not find any kind of defect/bug then the bug is fixed and the status assigned is ‘Verified’.
9. **Closed:**It is the final state of the Defect Cycle, after fixing the defect by the developer team when testing found that the bug had been resolved and did not persist they marked the defect as a ???? lost’ state.

**40. List some Bug Tracking Tools.**

1. **JIRA:**One of the most essential bug-tracking tools is Jira. Jira is an open-source platform used in manual testing for bug tracking, project management, and problem tracking. Jira contains a variety of capabilities such as reporting, recording, and workflow. We can monitor all types of faults and issues connected to software that is created by the test engineer in Jira.
2. **BugHerd:**BugHerd is the simplest way to monitor issues, collect feedback, and manage web page feedback. It also saves information like the browser, CSS selector data, operating system, and even a screenshot to help quickly recreate and fix errors. It is the most user-friendly tool for tracking problems and managing website feedback.
3. **Bugzilla:**Bugzilla is an open-source program that is used to assist the customer and client in keeping track of issues. It is also used as a test management tool since it allows us to quickly connect other test case management solutions such as ALM, Quality Centre, and so on.
4. **Axosoft:**Axosoft is a bug-tracking solution that may be used with hosted or on-premises applications. It is a Scrum team project management tool.
5. **Backlog:** Backlog is a web-based error/bug tracking and project management application designed for software development teams. The tool comes with a variety of features, including subtasks and detailed status charts, as well as iOS and Android apps.

**41. List the differences between Quality Assurance (QA) and Quality Control (QC).**

| **Parameters** | **Quality Assurance (QA)** | **Quality Control (QC)** |
| --- | --- | --- |
| **Definition** | QA is a group of activities that ensures that the quality of processes used during software development is always maintained. | QC is a group of activities to detect defects in the developed software. |
| **Focus** | QA focuses on assuring that the quality requested will be achieved. | QC focuses on fulfilling the quality request. |
| **Orientation** | QA is process-oriented. | QC is product-oriented. |
| **Tool Type** | QA is a managerial tool. | QC is a corrective tool. |
| **Example** | Verification | Validation |

**42. What is Pesticide Paradox?**

Pesticide paradox means if the same tests are repeated over and over again then the same test cases will no longer find new bugs. Some of the methods to prevent pesticide paradoxes are to write a whole new set of test cases to exercise different parts of the software or to prepare new test cases and add them to the existing test cases.

**43. What is a Critical Bug?**

A critical bug is a bug that tends to affect the majority of the functionality of the given application or software. The software cannot be released unless the critical bug is addressed.

**44. What makes a good test engineer?**

A good test engineer should have the following traits:

* Detail Oriented and organized.
* Has excellent problem-solving skills.
* Has strong communication and collaboration skills.
* A good test engineer must be up to date on the latest technologies.

**45. List the differences between Alpha testing and Beta testing.**

| **Parameters** | **Alpha Testing** | **Beta Testing** |
| --- | --- | --- |
| **Definition** | Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or the public. | Beta testing is the process of testing a software product or service in a real-world environment before its official release. |
| **Performed By** | Alpha testing is performed by testers who are usually employees of the organization. | Beta testing is performed by clients who are not part of the organization. |
| **Type of testing** | Alpha testing involves both white box and black box testing. | Beta testing involves black box testing. |
| **Parameters checked** | Reliability and security testing are not checked in alpha testing. | Reliability, security, and robustness are checked during beta testing. |
| **Testing environment** | Alpha testing requires a testing environment or lab. | Reliability, security, and robustness are checked during beta testing. |

**46. How many test cases can be executed in a day in Manual Testing?**

It depends upon the test case complexity and the size. Some test cases have few steps and some have more test steps.

**47. How do you derive test cases?**

It depends upon the project, sometimes we derive test cases from requirements and sometimes from use cases.

**48. How much time is required to write a test case?**

This depends upon the complexity of the software project.

**49. Were you involved in Test plan documentation in your career?**

Yes, I was involved in test plan documentation in my last project. I have identified Entry criteria, exit criteria, features to be tested, etc.

**50. Why did you choose Software Testing as a career?**

I would love to be a Software tester because I love solving puzzles and testing is like solving a puzzle, not only finding bugs but breaking into the system through stress testing.

### 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.

### 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

### 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.

21. List the four different test levels

The four levels are:

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

We live in a world that is becoming increasingly online. As a result, the demand for new software and apps is increasing to meet the ever-increasing number of consumers. However, since all this new development necessitates a quality control mechanism, software testers are in higher demand.

This article provides you with many of the top manual testing interview questions/ software testing interview questions that you can use to boost your confidence before sitting down for that crucial interview. If you’re already involved in some aspects of [software development](https://www.simplilearn.com/reasons-why-software-development-is-good-career-choice-for-women-article), you should take a look anyway, with the ultimate goal of upskilling. After all, the better command you have of every stage of software development, the more your marketability increases.

We’ll begin with the easy questions and work up to the tougher ones. Once you get through the manual testing interview questions, though, you may want to pay close attention to what follows, which sheds some light on the future of [software testing.](https://www.simplilearn.com/how-to-build-career-in-software-testing-article)

### 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](https://www.simplilearn.com/what-is-quality-control-article) 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](https://www.simplilearn.com/best-automation-testing-tools-for-software-development-article) 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](https://www.simplilearn.com/manual-testing-article) 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](https://www.simplilearn.com/tutorials/programming-tutorial/coding-for-beginners), 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](https://www.simplilearn.com/white-box-vs-black-box-testing-rar397-article) 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.

### 26. Why is Software Testing Required?

Software testing is required to ensure the quality and reliability of a software product.

* Testing helps to uncover any bugs, errors, or other issues in the software so that they can be addressed and fixed before the product is released.
* Testing also ensures that the software meets all the requirements specified by the customer and works as expected.
* Finally, testing helps to ensure that the software is secure and can withstand malicious attacks.

### 27. What are the different levels of manual testing?

The different levels of manual testing are:

* Unit Testing
* Integration Testing
* System Testing
* User Acceptance Testing
* Performance Testing
* Security Testing
* Compatibility Testing
* Usability Testing
* Installation Testing
* Smoke testing
* Sanity testing
* Regression Testing

### 28. Explain the procedure for manual testing?

Manual testing is a process of identifying bugs and errors in a software product without the use of automated tools. The procedure for manual testing is as follows:

* Identify the scope of testing: The first step of manual testing is to identify the scope of testing. The range could be a specific module, functionality, feature, or end-to-end system.
* Design test cases: The next step is to design test cases based on the identified scope. The test cases should include test scenarios, data, expected results, and all other details necessary to perform the tests.
* Execute the test cases: After designing the test cases, the testers execute them to find any discrepancies between the expected and actual results.
* Record the results: While performing the tests, the testers should record the results for further analysis.

### 29. What is the test case?

A test case is a set of conditions or variables under which a tester will determine whether a software system or one of its features is working as it was originally established for it to do. It may take the form of input, action, or environmental conditions. In addition, a test case includes requirements, test steps, verification steps, prerequisites, outputs, and actual results.

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

Documentation is an integral part of manual testing. It is essential to document all steps taken in the testing process to ensure thorough test coverage and accurate results. Documentation provides an audit trail, which can be used to evaluate past test results and identify areas for improvement. Additionally, it is a reference for other testers who may be unfamiliar with the system or application under test.

### Related Testing Interview Guides

|  |  |
| --- | --- |
| [Mobile Testing](https://www.simplilearn.com/mobile-testing-interview-questions-and-answers-article) | [API Testing](https://www.simplilearn.com/top-api-testing-interview-questions-article) |
| [Selenium](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers) | [Automation Testing](https://www.simplilearn.com/automation-testing-interview-questions-and-answers-article) |
| [SQL](https://www.simplilearn.com/top-sql-interview-questions-and-answers-article) | [Database Testing](https://www.simplilearn.com/database-testing-interview-questions-article) |

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

Software testing is classified into two main categories.

1. Functional testing
2. Non-Functional testing

### 32. 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 behavior 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](https://www.simplilearn.com/what-is-integration-testing-examples-challenges-approaches-article)
* Regression Testing
* System Testing
* Smoke Testing
* Performance Testing
* Stress Testing

### 33. Explain Non functional testing

[Non-functional testing](https://www.simplilearn.com/what-is-non-functional-testing-article) examines the system's non-functional requirements, which are the system's characteristics or qualities that the client has specifically requested. These include performance, security, scalability, and usability.

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.

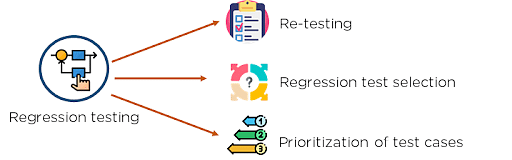
### 34. Mention a few advantages of Automated testing.

The following are some major advantages of [automated testing](https://www.simplilearn.com/what-is-automation-testing-article) -

* 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](https://www.simplilearn.com/tutorials/jenkins-tutorial), 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.

### 35. 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.

### 37. 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 that your app can gracefully deal with unexpected user behaviour or incorrect input. |
| In this testing, the tester always looks for a single set of valid data. | Testers use as much ingenuity as possible when validating the app against erroneous data. |

### 38. 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.

### 39. 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

### 46. 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.

## Become a Software Development Professional

* 40% Annual Growth
* 10 million new jobs

#### Automation Testing Masters Program

* + Comprehensive blended learning program
  + 200 hours of Applied Learning

8 months

[**View Program**](javascript:void(0))

#### Full Stack Java Developer Masters Program

* + Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts
  + Hands-on practice through 20+ projects, assessments, and tests

7 months

[**View Program**](javascript:void(0))

### Here's what learners are saying regarding our programs:

* Daniel Altufaili

#### Daniel Altufaili

##### IT infrastructure oprations, Johnson Electric

This Program had a tremendous impact on my career. The learning experience, including the patient and knowledgeable lecturers, was enriching. The blended learning approach allowed me to gain valuable skills in IT, IoT, and ML. This program helped me excel in my current role in the United States and led to a promotion and a 20% salary hike.

* Jonathan Mabiala

#### Jonathan Mabiala

##### Java Software Developer, Desjardins

I chose to upskill after moving from the United States to Canada in 2019. My journey in programming began during engineering, but I lost touch after college. Realizing I needed certification to advance, I enrolled in Simplilearn. The live classes boosted my confidence, allowing me to transition to a Java Software Developer role.

Not sure what you’re looking for?[View all Related Programs](https://www.simplilearn.com/mobile-and-software-development?source=InpageBannerCategory)

### 33. Explain Non functional testing

[Non-functional testing](https://www.simplilearn.com/what-is-non-functional-testing-article) examines the system's non-functional requirements, which are the system's characteristics or qualities that the client has specifically requested. These include performance, security, scalability, and usability.

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.

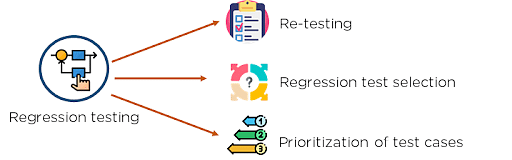
### 34. Mention a few advantages of Automated testing.

The following are some major advantages of [automated testing](https://www.simplilearn.com/what-is-automation-testing-article) -

* 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](https://www.simplilearn.com/tutorials/jenkins-tutorial), 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.

### 35. 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.

### 36. 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.

### 37. 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 that your app can gracefully deal with unexpected user behaviour or incorrect input. |
| In this testing, the tester always looks for a single set of valid data. | Testers use as much ingenuity as possible when validating the app against erroneous data. |

### 38. What is a Critical Bug?

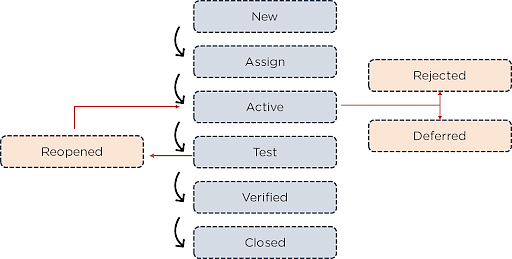
A critical bug is one that has the potential to affect the bulk of an application's functioning. It indicates that a significant portion of functionality or a critical system component is utterly broken, with no way to proceed. The application cannot be delivered to end users until the critical bug has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

### 39. 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.

### 40. 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.



### 41. 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.

### 42. What is API testing?

[API testing](https://www.simplilearn.com/tutorials/jmeter-tutorial/jmeter-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.

## Become a Software Development Professional

* 40% Annual Growth
* 10 million new jobs

#### Automation Testing Masters Program

* + Comprehensive blended learning program
  + 200 hours of Applied Learning

8 months

[**View Program**](javascript:void(0))

#### Full Stack Java Developer Masters Program

* + Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts
  + Hands-on practice through 20+ projects, assessments, and tests

7 months

[**View Program**](javascript:void(0))

### Here's what learners are saying regarding our programs:

* Daniel Altufaili

#### Daniel Altufaili

##### IT infrastructure oprations, Johnson Electric

This Program had a tremendous impact on my career. The learning experience, including the patient and knowledgeable lecturers, was enriching. The blended learning approach allowed me to gain valuable skills in IT, IoT, and ML. This program helped me excel in my current role in the United States and led to a promotion and a 20% salary hike.

* Jonathan Mabiala

#### Jonathan Mabiala

##### Java Software Developer, Desjardins

I chose to upskill after moving from the United States to Canada in 2019. My journey in programming began during engineering, but I lost touch after college. Realizing I needed certification to advance, I enrolled in Simplilearn. The live classes boosted my confidence, allowing me to transition to a Java Software Developer role.

Not sure what you’re looking for?[View all Related Programs](https://www.simplilearn.com/mobile-and-software-development?source=InpageBannerCategory)

### 43. 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.

### 44. 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.

### 45. 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

### 47. 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.

### 48. 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.

### 49.  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.

### 50. 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.

### 51. 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.

### 52. 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.

### 54. 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.

### 55. 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.

### 56. What is the difference between manual testing and automation testing?

Manual testing is the process of manually testing software for defects. It requires a tester to manually execute the test steps and compare the actual and expected results. Automation testing uses special software to control the execution of tests and compare the results with the desired results. As a result, automation testing is much faster than manual testing and can reduce the time required to complete a test cycle.

### 57. When should you opt for manual testing over automation testing?

Manual testing should be used over automation testing when the tests are particular or require human interpretation. Manual testing is also better suited for exploratory testing, usability testing, and testing on multiple operating systems or unique hardware.

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

The phases involved in the Software Testing Life Cycle are:

* Test Planning
* Test Analysis
* Test Design
* Test Implementation
* Test Execution
* Test Results Analysis
* Test Closure

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

A top-down and bottom-up approach in testing refers to the order of testing.

* Top-down testing begins at the highest level and works downward. Thus, each higher-level component is tested in isolation from the lower-level components.
* Bottom-up testing starts at the lowest level and works upward. Thus, each lower-level component is tested in isolation from higher-level components.
* 65. What is the difference between smoke testing and sanity testing?

Smoke testing is a high-level test used to ensure the most critical functions of a software system are working correctly. It is a quick test that can be used to determine whether it is worth investing time and energy into further, more extensive testing. Sanity testing is a more specific test used to check that recent changes to a system have not caused any new, unwanted behavior. It ensures that basic features are still functioning as expected after minor changes have been made.

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

Static testing is a type of testing performed without executing the code of a software application. Instead, it includes reviews, inspections, and walkthroughs.

Dynamic testing is a type of testing that involves executing the code of a software application to determine the results of certain functions and operations. It includes unit testing, integration testing, and acceptance testing.