# Edureka: 50Q

### ****Q1. How does quality control differ from quality assurance?****

|  |  |
| --- | --- |
| **Quality Control vs Quality Assurance** | |
| **Quality Control** | **Quality Assurance** |
| Quality control is a product-oriented approach 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. |

### ****Q2. What is Software Testing?****

[*Software Testing*](https://www.edureka.co/blog/software-testing-tutorial/#IntroductiontoSoftwareTesting)is a process used to identify the correctness, completeness, and quality of developed software. It includes a series of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the market.

### ****Q3. Why is Software Testing Required?****

Software testing is a mandatory process that guarantees that the software product is safe and good enough to be released to the market. Here are some compelling reasons to prove testing is needed:

* It points out the defects and errors that were made during the [development phases.](https://www.edureka.co/blog/software-testing-tutorial/#SoftwareDevelopmentLifeCycle)
* Reduces the coding cycles by identifying issues at the initial stage of the development.
* Ensures that software application requires lower maintenance cost and results in more accurate, consistent and reliable results.
* Testing ensures that the customer finds the organization reliable and their satisfaction in the application is maintained.
* Makes sure that software is bug-free and the quality of the product meets the market standard.
* Ensures that the application doesn’t result in any failures.

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

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

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

Become certified in automation testing get [Selenium Online training](https://www.edureka.co/selenium-certification-training)!

### ****Q5****. ****What is quality control? Is it similar to Quality Assurance?****

Quality control is a product-oriented approach 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.

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

Different types of manual testing are;

[[](https://www.edureka.co/software-testing-fundamentals-training)](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

### [Software Testing Fundamentals Course](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

* *[Instructor-led Sessions](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)*
* *[Real-life Case Studies](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)*
* *[Assessments](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)*
* *[Lifetime Access](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)*

[Explore Curriculum](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

* + Black Box Testing
  + White Box Testing
  + Unit Testing
  + System Testing
  + Integration Testing
  + Acceptance Testing

### ****Q7. Explain the difference between alpha testing and beta testing.****

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

### ****Q8. What are the different levels of manual testing?****

Four levels of manual testing are:

* [**Unit testing**](https://www.edureka.co/blog/what-is-unit-testing) – It is a way of testing the smallest piece of code referred to as a **unit** that can be logically isolated in a system. It is mainly focused on the functional correctness of the standalone module.
* [**Integration Testing**](https://www.edureka.co/blog/what-is-integration-testing-a-simple-guide-on-how-to-perform-integration-testing/) – It is a level of software testing where individual units are combined and tested to verify if they are working as they intend to when integrated. The main aim here is to test the interface between the modules.
* **System Testing** – In system testing all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified. There are dozens of types of system testing, including usability testing, regression testing, and functional testing.

**User Acceptance Testing** – The final level, acceptance testing, or UAT (user acceptance testing), determines whether or not the software is ready to be released.

### ****Q9. What is a testbed in manual testing?****

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

### ****Q10. Explain the procedure for manual testing?****

The manual testing process comprises the following steps:

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

In case you are facing any challenges with these Manual Testing interview questions, please comment on your problems in the section below.

### ****Q11. What is the test case?****

A[**test case**](https://www.edureka.co/blog/test-case-in-software-testing/) is a document that has a set of conditions or actions that are performed on the software application in order to verify the expected functionality of the feature.

Test cases describe a specific idea that is to be tested, without detailing the exact steps to be taken or data to be used. For example, in a test case, you document something like ‘**Test if coupons can be applied on actual price**‘.

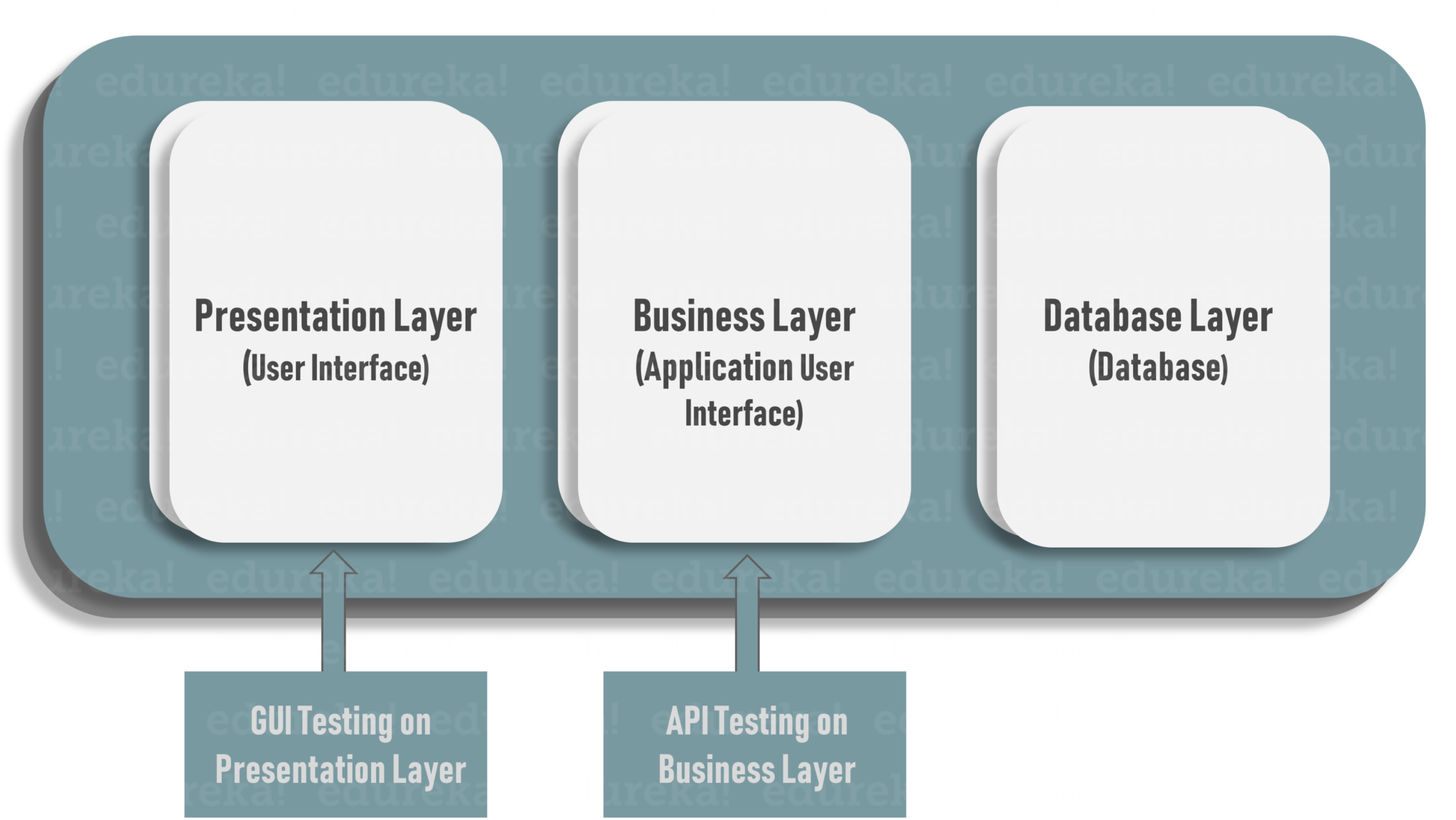
### ****Q12. What is API testing?****

[API testing](https://www.edureka.co/blog/what-is-api-testing) is a type of software testing where application programming interfaces (APIs) are tested to determine if they meet expectations for functionality, reliability, performance, and security. In simple terms, API testing is intended to reveal bugs, inconsistencies or deviations from the expected behavior of an API.

[[](https://www.edureka.co/software-testing-fundamentals-training)](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

### [Software Testing Fundamentals Course](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

[Watch The Course Preview](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)



Commonly, applications have three separate layers:

* Presentation Layer or user interface
* Business Layer or application user interface for business logic processing
* Database Layer for modeling and manipulating data

API testing is performed at the most critical layer of software architecture, the Business Layer.

### ****Q13. What’s the difference between verification and validation in testing?****

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| It is a static analysis technique. Here, testing is done without executing the code. Examples include – Reviews, Inspection, and walkthrough. | It is a dynamic analysis technique where testing is done by executing the code. Examples include functional and non-functional testing techniques. |

### ****Q14. What’s the difference between a bug and a defect?****

A [bug](https://www.edureka.co/blog/bugs-in-software-testing/#softwaretestingbugs) is a just fault in the software that’s detected during testing time. A defect is a variance between expected results and actual results, detected by the developer after the product goes live.

### ****Q15.What are the advantages of manual testing?****

Merits of manual testing are:

* It is a cheaper way of testing when compared to automated testing
* Analysis of product from the point of view of the end-user is possible only with manual testing
* GUI testing can be done more accurately with the help of manual testing as visual accessibility and preferences are difficult to automate
* East to learn for new people who have just entered into testing
* It is highly suitable for short-term projects when test-scripts are not going to be repeated and reused for thousands of times
* Best suited when the project is at the early stages of its development
* Highly reliable, since automated tests can contain errors and missed bugs

### ****Q16.What are the drawbacks of manual testing?****

De-merits of manual testing are:

* Highly susceptible to human error and are risky
* Test types like load testing and performance testing are not possible manually
* Regression tests are really time-consuming if they are done manually
* Scope of manual testing is very limited when compared to automation testing
* Not suitable in very large organizations and time-bounded projects
* The cost adds up, so, it’s more expensive to test manually in the long run

### ****Q17****. What’s the role of documentation in Manual Testing?

Documentation plays a critical role in achieving effective [software testing](https://www.edureka.co/testing-with-selenium-webdriver). Details like requirement specifications, designs, business rules, inspection reports, configurations, code changes, test plans, test cases, bug reports, user manuals, etc. should all be documented.

Documenting the test cases will facilitate you to estimate the testing effort you will need along with test coverage and tracking and tracing requirement. Some commonly applied documentation artifacts associated with software testing are:

[[](https://www.edureka.co/software-testing-fundamentals-training)](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

### [Software Testing Fundamentals Course](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/software-testing-fundamentals-training" \t "_blank)

1. Test Plan
2. Test Scenario
3. Test Case
4. Traceability Matrix

With this, we have completed basic questions based on manual testing. In the next part of this Manual Testing Interview Questions article, let’s discuss advanced level questions related to manual testing.

Want to upskill yourself to get ahead in your career? Check out this video

## ****Top 10 Technologies to Learn in 2022 | Edureka****



## ****Advanced Level Manual Testing Interview Questions****

### ****Q18. What is the difference between manual testing and automation testing?****

|  |  |
| --- | --- |
| Manual Testing | Automation Testing |
| In manual testing, the accuracy, and reliability of test cases are low, as manual tests are more prone to human error. | Automated testing, on the other hand, is more reliable as tools and scripts are used to perform tests. |
| The time required for manual testing is high as human resources perform all the tasks. | The time required is comparatively low as software tool execute the tests |
| In manual testing investment cost is low, but Return of Investment(ROI) is low as well. | In automation testing investment cost and Return of Investment, both are high. |
| Manual testing is preferred when the test cases are run once or twice. Also suitable for Exploratory, Usability and Adhoc Testing. | You can use test automation for Regression Testing, Performance Testing, Load Testing or highly repeatable functional test cases |
| Allows for human observation to find out any glitches. Therefore manual testing helps in improving the customer experience. | As there is no human observation involved, there is no guarantee of positive customer experience. |

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

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

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

### ****Q20.**** ****What are the phases involved in Software Testing Life Cycle?****

The different phases involved in the [software testing life cycle](https://www.edureka.co/blog/software-testing-life-cycle/#stlc) are:

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

In case you are facing any challenges with these Manual Testing interview questions, please comment on your problems in the section below.

### ****Q21. What is the difference between a bug, a defect and an error?****

**Bug** – A bug is a fault in the software that’s detected during testing time. They occur because of some coding error and leads a program to malfunction. They may also lead to a functional issue in the product. These are fatal errors that could block a functionality, results in a crash, or cause performance bottlenecks

**Defect** – A defect is a variance between expected results and actual results, detected by the developer after the product goes live. The defect is an error found AFTER the application goes into production. In simple terms, it refers to several troubles with the software products, with its external behavior, or with its internal features.

**Error** – An error is a mistake, misunderstanding, or misconception, on the part of a software developer. The category of developers includes software engineers, programmers, analysts, and testers. For example, a developer may misunderstand a design notation, or a programmer might type a variable name incorrectly – leads to an error. An error normally arises in software, it leads to a change the functionality of the program.

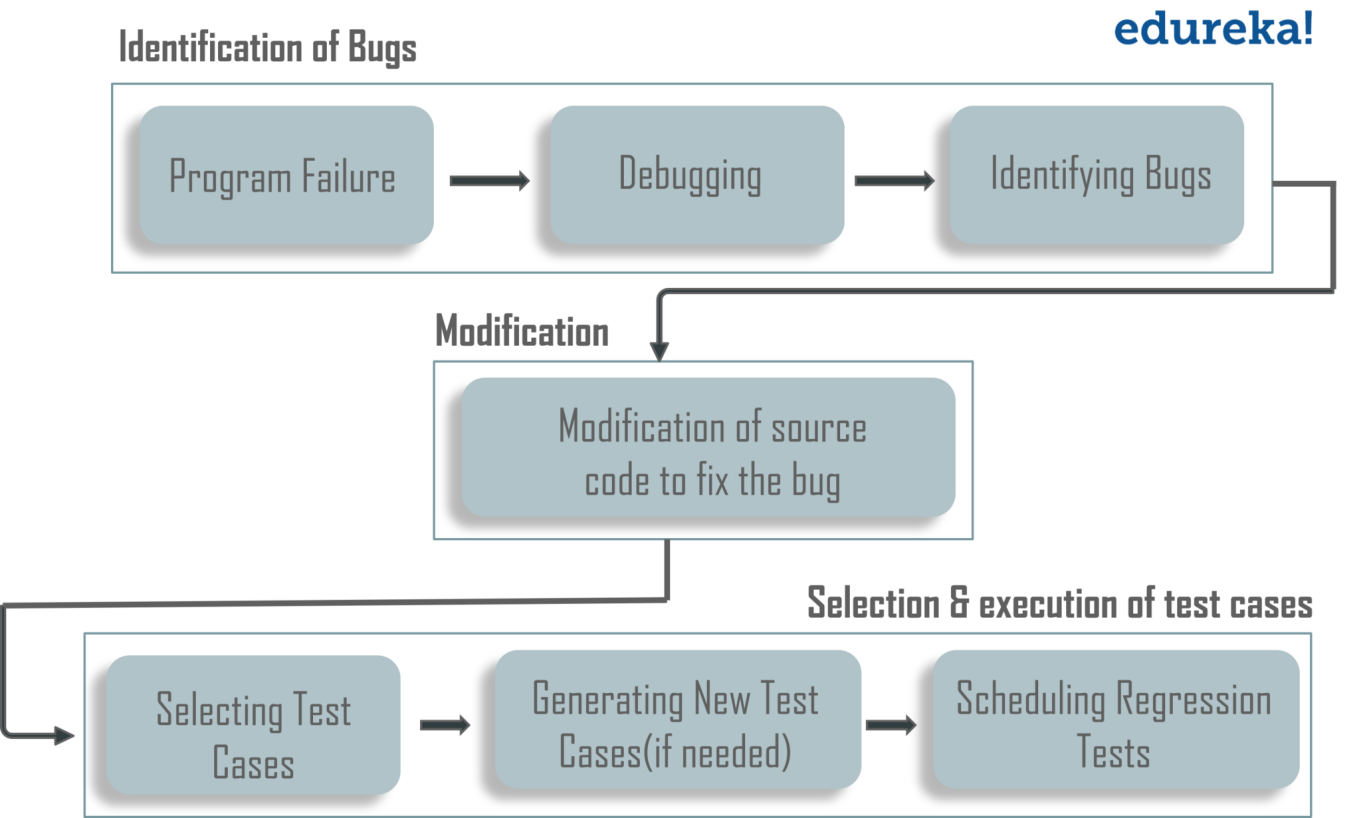
### ****Q22. What makes a good test engineer?****

A **software** **test engineer** is a professional who determines how to create a process that would best test a particular product in the software industry.

* A good test engineer should have a ‘test to break’ attitude, an ability to take the point of view of the customer
* Strong desire for quality and attention to minute details
* Tact and diplomacy to maintain a cooperative relationship with developers
* Ability to communicate with both technical (developers) and non-technical (customers, management) people
* Prior experience in the software development industry is always a plus
* Ability to judge the situations and make important decisions to test high-risk areas of an application when time is limited

### ****Q23.What is regression testing? When to apply it?****

“Testing of a previously tested program to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made is called [*Regression Testing*](https://www.edureka.co/blog/regression-testing).”



A regression test is a system-wide test whose main purpose is to ensure that a small change in one part of the system does not break existing functionality elsewhere in the system. It is recommended to perform regression testing on the occurrence of the following events:

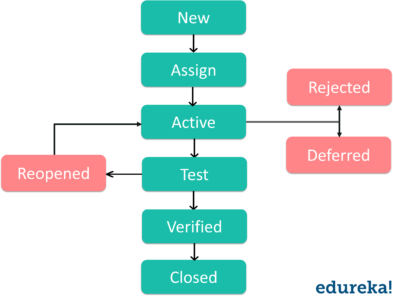
* When new functionalities are added
* In case of change requirements
* When there is a defect fix
* When there are performance issues
* In case of environment changes
* When there is a patch fix

### ****Q24. What is the difference between system testing and integration testing?****

|  |  |
| --- | --- |
| **System Testing** | **Integration Testing** |
| System Testing tests the software application as a whole to check if the system is compliant with the user requirements | Integration testing tests the interface between modules of the software application |
| Involves both functional and non-functional testings like sanity, usability, performance, stress an load | Only functional testing is performed to check whether the two modules when combined give the right outcome |
| It is high-level testing performed after integration testing | It is low-level testing performed after unit testing |

### ****Q25. Explain the defect life cycle.****

A**defect life cycle** is a process in which a defect goes through various phases during its whole lifetime. The cycle starts when a defect is found and ends when a defect is closed, after ensuring it’s not reproduced. Bug or defect life cycle includes the steps as shown in the below figure.



If you wish to learn in-depth about Bug Life Cycle then you can refer this article on [Software Testing Tutorial](https://www.edureka.co/blog/software-testing-tutorial/).

### ****Q26. What is the test harness?****

A test harness is the gathering of software and test information arranged to test a program unit by running it under changing conditions like stress, load, data-driven, and monitoring its behavior and outputs. Test Harness contains two main parts:

– A Test Execution Engine  
– Test script repository

### ****Q27. What is test closure?****

Test Closure is a document which gives a summary of all the tests conducted during the software development life cycle and also gives a detailed analysis of the bugs removed and errors found. This memo contains the aggregate no. of experiments, total no. of experiments executed, total no. of imperfections discovered, add total no. of imperfections settled, total no. of bugs not settled, total no of bugs rejected and so forth.

### ****Q28. What is the difference between Positive and Negative Testing?****

|  |  |
| --- | --- |
| **Positive Testing** | **Negative Testing** |
| **Positive testing** determines that your application works as expected. If an error is encountered during positive testing, the test fails | **Negative testing** ensures that your application can gracefully handle invalid input or unexpected user behavior |
| In this testing, tester always check for an only valid set of values | Testers apply as much creativity as possible and validating the application against invalid data |

### ****Q29. Define what is a critical bug.****

A critical bug is a [bug](https://www.edureka.co/blog/bugs-in-software-testing/) that has got the tendency to affect a majority of the functionality of the given application. It means a large piece of functionality or major system component is completely broken and there is no workaround to move further. Application cannot be distributed to the end client unless the critical bug is addressed.

### ****Q30. What is the pesticide paradox? How to overcome it?****

According to pesticide paradox, if the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs. Developers will be extra careful in those places where testers found more defects and might not look into other areas. **Methods to prevent pesticide paradox:**

* To write a whole new set of test cases to exercise different parts of the software.
* To prepare new test cases and add them to the existing test cases.

Using these methods, it’s possible to find more defects in the area where defect numbers dropped.

In case you are facing any challenges with these Manual Testing interview questions, please comment on your problems in the section below.

### ****Q31. What is Defect Cascading in Software Testing?****

Defect Cascading is the process of triggering other defects in the application. When a defect goes unnoticed while testing, it invokes other defects. As a result, multiple defects crop up in the later stages of development. If defect cascading continues to affect other features in the application, identifying the affected feature becomes challenging. You may make different test cases to solve this issue, even then it is difficult and time-consuming.

### ****Q32. What is the term ‘quality’ mean when testing?****

In general, quality software is reasonably bug-free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable. But again ‘quality’ is a subjective term. It will depend on who the ‘customer’ is and their overall influence in the scheme of things. For example, each type of ‘customer’ will have their own slant on ‘quality’ – the accounting department might define quality in terms of profits while an end-user might define quality as user-friendly and bug-free.

### ****Q33. What is black box testing, and what are the various techniques?****

[Black-Box Testing](https://www.edureka.co/blog/software-testing-methodologies-and-techniques/#BlackBoxTechniques), also known as specification-based testing, analyses the functionality of a software/application without knowing much about the internal structure/design of the item. The purpose of this testing is to check the functionality of the system as a whole to make sure that it works correctly and meets user demands. Various black-box testing techniques are:

* Equivalence Partitioning
* Boundary Value Analysis
* Decision Table Based Technique
* Cause-effect Graphing
* Use Case Testing

### ****Q34. What is white box testing, and what are the various techniques?****

[White-Box Testing](https://www.edureka.co/blog/software-testing-methodologies-and-techniques/#WhiteBoxTechniques) also known as structure-based testing, requires a profound knowledge of the code as it includes testing of some structural part of the application. The purpose of this testing is to enhance security, check the flow of inputs/outputs through application and to improve design and usability. Various white-box testing techniques are:

* Statement Coverage
* Decision Coverage
* Condition Coverage
* Multiple Condition Coverage

### ****Q35. What are the Experience-based testing techniques?****

[Experienced-based testing](https://www.edureka.co/blog/software-testing-methodologies-and-techniques/#ExperienceBasedTechniques) is all about discovery, investigation, and learning. The tester constantly studies and analyzes the product and accordingly applies his skills, traits, and experience to develop test strategies and test cases to perform necessary testing. Various experience-based testing techniques are:

* Exploratory Testing
* Error Guessing

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

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

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

### ****Q37. What is the difference between smoke testing and sanity testing?****

|  |  |  |
| --- | --- | --- |
| **Features** | **Smoke Testing** | **Sanity Testing** |
| **System Builds** | Tests are executed on initial builds of software product | Tests are done on builds that have passed smoke tests & rounds of regression tests |
| **Motive of Testing** | To measure the stability of the newly created build to face off more rigorous testing | To evaluate rationality & originality of the functionalities of software builds |
| **Subset of?** | Is a subset of acceptance testing | Is a subset of regression testing |
| **Documentation** | Involves documentation and scripting work | Doesn’t emphasize any sort of documentation |
| **Test Coverage** | Shallow & wide approach to include all the major functionalities without going too deep | Narrow & deep approach involving detailed testing of functionalities and features |
| **Performed By?** | Executed by developers or testers | Executed by testers |

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

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

With this, we have completed theory questions. In the next part of this Manual Testing Interview Questions article, let’s discuss some real-world scenario-based questions.

## ****Real-World Based Manual Testing Interview Questions****

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

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

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

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

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

In case you are facing any challenges with these Manual Testing interview questions, please comment on your problems in the section below.

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

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

### ****Q42.**** What if an organization is growing so fast that fixed testing processes are impossible? What to do in such situations?

This is a very common problem in the software industry, especially considering the new technologies that are being incorporated when developing the product. There is no easy solution in this situation, you could:  
• Hire good and skilled people   
• Management should ‘ruthlessly prioritize’ quality issues and maintain focus on the customer   
• Everyone in the organization should be clear on what ‘quality’ means to the end-user

### ****Q43. How do you know the code has met specifications?****

‘Good code’ is code that works, that is bug-free, and is readable and maintainable. Most organizations have coding ‘standards’ that all developers are supposed to adhere to, but everyone has different ideas about what’s best, or what is too many or too few rules. There are a lot of tools like traceability matrix which ensures the requirements are mapped to the test cases. And when the execution of all test cases finishes with success, it indicates that the code has met the requirement.

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

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

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

### ****Q45. What is ‘configuration management’?****

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

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

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

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

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

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

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

### ****Q48. Why is it that the boundary value analysis provides good test cases?****

The reason why boundary value analysis provides good test cases is that usually, a greater number of errors occur at the boundaries rather than in the center of the input domain for a test.

In boundary value analysis technique test cases are designed to include values at the boundaries. If the input is within the boundary value, it is considered ‘Positive testing.’ If the input is outside of the boundary value, it is considered ‘Negative testing.’ It includes maximum, minimum, inside or outside edge, typical values or error values.

Let’s suppose you are testing for an input box that accepts numbers from ’01 to 10′.

 Using the boundary value analysis we can define three classes of test cases:

* Test cases with test data exactly as the input boundaries of input: 1 and 10 (in this case)
* Values just below the extreme edges of input domains: 0 and 9
* Test data with values just above the extreme edges of input domains: 2 and 11

So the boundary values would be 0, 1, 2 and 9, 10, 11.

### ****Q49.Why is it impossible to test a program thoroughly or in other terms 100% bug-free?****

It is impossible to build a software product which is 100% bug-free. You can just minimize the error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result.

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

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

### ****Q50. Can automation testing replace manual testing?****

Automation testing isn’t a replacement for manual testing. No matter how good automated tests are, you cannot automate everything. Manual tests play an important role in software development and come in handy whenever you have a case where you cannot use automation. Automated and manual testing each have their own strengths and weaknesses. Manual testing helps us to understand the entire problem and explore other angles of tests with more flexibility. On the other hand, automated testing helps save time in the long run by accomplishing a large number of surface-level tests in a short time.

# Intellipat: 45Q

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

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

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

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

**Test case execution**

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

**Testing deadline**

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

**Code Coverage(CC) ratio**

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

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

### ****3. What do verification and validation mean in software testing?****

In software testing, verification is a process to confirm that product development is taking place as per the specifications and using the standard development procedures. The process comprises the following activities:

* Inspections
* Reviews
* Walk-throughs
* Demos

### Validation is a means to confirm that the developed product doesn’t have any bugs and is working as expected. It comprises the following activities:

* Functional testing
* Non-functional testing

***Preparing for a Job Interview! Check out our Top***[***Software Testing Interview Questions***](https://intellipaat.com/blog/interview-question/software-testing-interview-questions/)***.***

### ****4. What is static testing? When does it start and what does it cover?****

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

### ****5. Define Black-box testing.****

It is a standard software testing approach that requires testers to assess the functionality of the software as per the business requirements. The software is treated as a black box and validated as per the end user’s point of view.

### ****6. What is a test plan and what does it include?****

A test plan stores all possible testing activities to ensure a quality product. It gathers data from the product description, requirement, and use case documents.

The test plan document includes the following:

* Testing objectives
* Test scope
* Testing the frame
* Environment
* Reason for testing
* Criteria for entrance and exit
* Deliverables
* Risk factors

### Get 100% Hike!

Master Most in Demand Skills Now !

Top of Form





Bottom of Form

### ****7. What is meant by test coverage?****

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

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

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

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

### ****9. What are unit testing and integration testing?****

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

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

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

**There are three ways to validate integration:**

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

### ****10. Can we do system testing at any stage?****

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

### ****11. Mention the different types of software testing.****

**Various**[**types of Software Testing**](https://intellipaat.com/blog/types-of-software-testing/)**used by manual testers are as follows:**

* 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

**Career Transition**

[[](javascript:void(0);)](javascript:void(0);)

[[](javascript:void(0);)](javascript:void(0);)

[[](javascript:void(0);)](javascript:void(0);)

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

The **test driver** is a section of code that calls a software component under test. It is useful in testing that follows the bottom-up approach.

The **test stub** is a dummy program that integrates with an application to complete its functionality. It is relevant for testing that uses the top-down approach.

For example:

1. Let’s assume a scenario where we have to test the interface between Modules A and B. We have developed only Module A. Here, we can test Module A if we have the real Module B or a dummy module for it. In this case, we call Module B as the test stub.
2. Now, Module B can’t send or receive data directly from Module A. In such a scenario, we’ve to move data from one module to another using some external features called test driver.

### ****13. What is agile testing and why is it important?****

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

### ****14. What do you know about data flow testing?****

It is one of the white-box testing techniques.

Data flow testing emphasizes for designing test cases that cover control flow paths around variable definitions and their uses in the modules. It expects test cases to have the following attributes:

1. The input to the module
2. The control flow path for testing
3. A pair of an appropriate variable definition and its use
4. The expected outcome of the test case

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

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

## Intermediate Interview Questions

### ****16. The probability that a server-class application hosted on the cloud is up and running for six long months without crashing is 99.99 percentage. To analyze this type of a scenario, what test you will perform?****

Reliability testing

### ****17. What will you do when a bug turns up during testing?****

When a bug occurs, we can follow the below steps.

* We can run more tests to make sure that the problem has a clear description.
* We can also run a few more tests to ensure that the same problem doesn’t exist with different inputs.
* Once we are certain of the full scope of the bug, we can add details and report it.

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

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

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

### ****19. How do you test a product if the requirements are yet to be freezed?****

If the required specifications are not available for a product, then a test plan can be created based on the assumptions made about the product. But we should get all assumptions well-documented in the test plan.

### ****20. If a product is in the production stage and one of its modules gets updated, then is it necessary to ret****

It is suggested to perform a regression testing and run tests for all the other modules as well. Finally, the QA should also carry out a system testing.

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

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

* Screenshots
* A previous version of the application
* Wireframes

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

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

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

Possible differences between retesting and regression testing are as follows:

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

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

Following are some of the key challenges of software testing:

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

### ****24. What are the different types of functional testing?****

Functional testing covers the following types of validation techniques:

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

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

* Functional testing: It is testing the ‘functionality’ of a software or an application under test. It tests the behavior of the software under test. Based on the requirement of the client, a document called a software specification or requirement specification is used as a guide to test the application.
* Non-functional testing: In software terms, when an application works as per the user’s expectation, smoothly and efficiently under any condition, then it is stated as a reliable application. Based on quality, it is very critical to test these parameters. This type of testing is called non-functional testing.

### ****26. What do you understand by STLC?****

Software testing life cycle (STLC) proposes the test execution in a planned and systematic manner. In the STLC model, many activities occur to improve the quality of the product.

**The STLC model lays down the following steps:**

1. Requirement Analysis
2. Test Planning
3. Test Case Development
4. Environment Setup
5. Test Execution
6. Test Cycle Closure

### ****27. In software testing, what does a fault mean?****

Fault is a condition that makes the software fail to execute while performing the considered function.

[](https://intellipaat.com/test-architect-masters-program-training/)

### ****28. Difference between Bug, Defect, and Error.****

A slip in coding is indicated as an error. The error spotted by a manual tester becomes a defect. The defect which the development team admits is known as a bug. If a built code misses on the requirements, then it is a functional failure.

### ****29. How do severity and priority relate to each other?****

**Severity:** It represents the gravity/depth of a bug. It describes the application point of view.

**Priority:** It specifies which bug should get fixed first. It defines the user’s point of view.

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

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

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

## Advanced Interview Questions

### ****31. What do you mean by defect detection percentage in software testing?****

Defect detection percentage (DDP) is a type of testing metric. It indicates the effectiveness of a testing process by measuring the ratio of defects discovered before the release and reported after the release by customers.

For example, let’s say, the QA has detected 70 defects during the testing cycle and the customer reported 20 more after the release. Then, DDP would be: 70/(70 + 20) = 72.1%

### ****32. What does defect removal efficiency mean in software testing?****

Defect removal efficiency (DRE) is one of the testing metrics. It is an indicator of the efficiency of the development team to fix issues before the release.

It gets measured as the ratio of defects fixed to total the number of issues discovered.

For example, let’s say, there were 75 defects discovered during the test cycle while 62 of them got fixed by the development team at the time of measurement. The DRE would be 62/75 = 82.6%

Go through the [**Manual Testing Training**](https://intellipaat.com/software-testing-training-course/) to get clear understanding of Weak AI and Strong AI.

### ****33. What is the average age of a defect in software testing?****

Defect age is the time elapsed between the day the tester discovered a defect and the day the developer got it fixed.

While estimating the age of a defect, consider the following points:

* The day of birth of a defect is the day it got assigned and accepted by the development team.
* The issues which got dropped are out of the scope.
* Age can be both in hours or days.
* The end time is the day the defect got verified and closed, not just the day it got fixed by the development team.

### ****34. How do you perform automated testing in your environment?****

Automation testing is a process of executing tests automatically. It reduces human intervention to a great extent. We use different test automation tools like QTP, [**Selenium**](https://intellipaat.com/blog/tutorial/selenium-tutorial/introduction/), and WinRunner. Testing tools help in speeding up the testing tasks. These tools allow you to create test scripts to verify the application automatically and also to generate the test reports.

***Preparing for a Job Interview! Check out our blog on***[***Selenium Interview Questions***](https://intellipaat.com/blog/interview-question/selenium-interview-questions/)***now.***

### ****35. Is there any difference between quality assurance, quality control, and software testing. If so, what is it?****

Quality Assurance (QA) refers to the planned and systematic way of monitoring the quality of the process which is followed to produce a quality product. QA tracks the test reports and modifies the process to meet the expectation.

Quality Control (QC) is relevant to the quality of the product. QC not only finds the defects but suggests improvements too. Thus, a process that is set by QA is implemented by QC. QC is the responsibility of the testing team.

Software testing is the process of ensuring that the product which is developed by developers meets the users’ requirements. The aim of performing testing is to find bugs and make sure that they get fixed. Thus, it helps to maintain the quality of the product to be delivered to the customer.

### ****36. Tell me about some of the essential qualities an experienced QA or Test Lead must possess.****

**A QA or Test Lead should have the following qualities:**

1. Well-versed in software testing processes
2. Ability to accelerate teamwork to increase productivity
3. Improve coordination between QA and Dev engineers
4. Provide ideas to refine QA processes
5. Skill to conduct RCA meetings and draw conclusions
6. Excellent written and interpersonal communication skills
7. Ability to learn fast and to groom the team members

### ****37. What is a Silk Test and why should you use it?****

**Here are some facts about the Silk Test tool:**

1. Skill tool is developed for performing regression and functionality testing of an application.
2. It is used when we are testing Window-based, Java, web, and the traditional client/server applications.
3. Silk Test helps in preparing the test plan and managing it to provide direct accessing of the database and validation of the field.

[](https://intellipaat.com/selenium-training/)

### ****38. On the basis of which factors you would consider choosing automated testing over manual testing?****

**Choosing**[**automation testing over manual testing**](https://intellipaat.com/blog/automation-vs-manual-testing/)**depends on the following factors:**

1. Tests require periodic execution.
2. Tests include repetitive steps.
3. Tests execute in a standard runtime environment.
4. Automation is expected to take less time.
5. Automation is increasing reusability.
6. Automation reports are available for every execution.
7. Small releases like service packs include a minor bug fix. In such cases, executing the regression test is sufficient for validation.

### ****39. Tell me the key elements to consider while writing a bug report.****

**An ideal bug report should consist of the following key points:**

* A unique ID
* Defect description: A short description of the bug
* Steps to reproduce: They include the detailed test steps to emulate the issue. They also provide the test data and the time when the error has occurred
* Environment: Add any system settings that could help in reproducing the issue
* Module/section of the application in which the error has occurred
* Severity
* Screenshots
* Responsible QA: This person is a point of contact in case you want to follow-up regarding this issue

### ****40. Is there any difference between bug leakage and bug release?****

**Bug leakage:** Bug leakage is something, when the bug is discovered by the end user/customer and missed by the testing team to detect while testing the software. It is a defect that exists in the application and not detected by the tester, which is eventually found by the customer/end user.

**Bug release:** A bug release is when a particular version of the software is released with a set of known bug(s). These bugs are usually of low severity/priority. It 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.

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

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

Monkey testing is a technique in software testing where the user tests the application by providing random inputs, checking the behavior of the application (or trying to crash the application).

### ****42. What is exploratory testing?****

Exploratory testing is an approach to software testing, wherein testers learn simultaneously about the test design and test execution. In other words, it is a hands-on approach where testers are involved more in the test execution part than in planning.

### ****43. What is meant by system testing?****

System testing is a black-box testing technique, used on a complete integrated system, where it will test the system compliance as per the requirement.

[](https://intellipaat.com/test-architect-masters-program-training/)

### ****44. What are the benefits of test reports?****

Test reports will help us find the current status of a project and its quality. This can help stakeholders and customers take necessary actions. The complete documentation of test reports will help analyze different phases of the project.

### ****45. What is meant by latent defect?****

A latent defect is a hidden defect in an application/software, which cannot be identified by a user. However, this will not cause any failure to the application because the conditions will never be met.

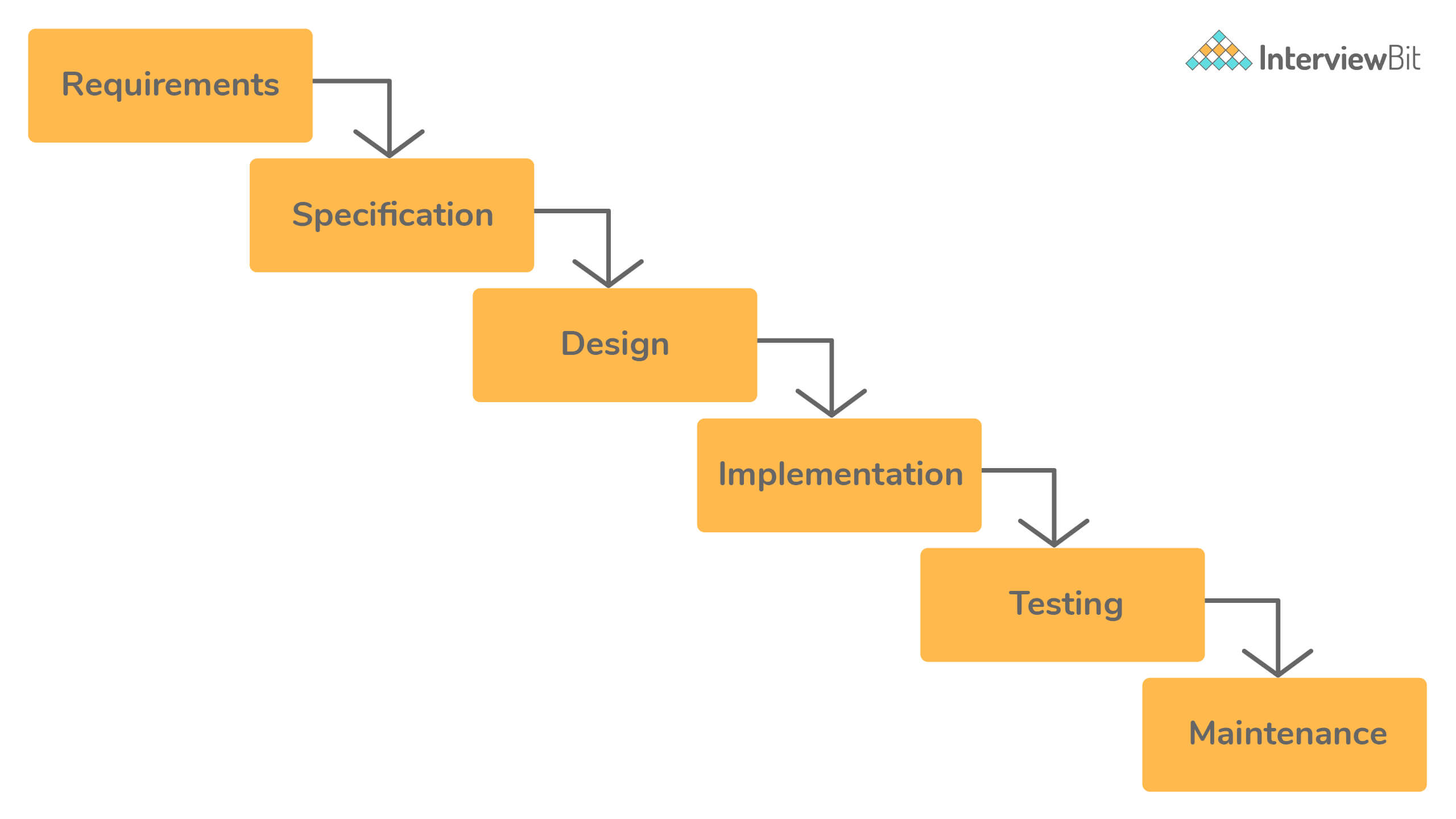
# 3.InterviewBit: 88Q

## Software Testing Interview Questions

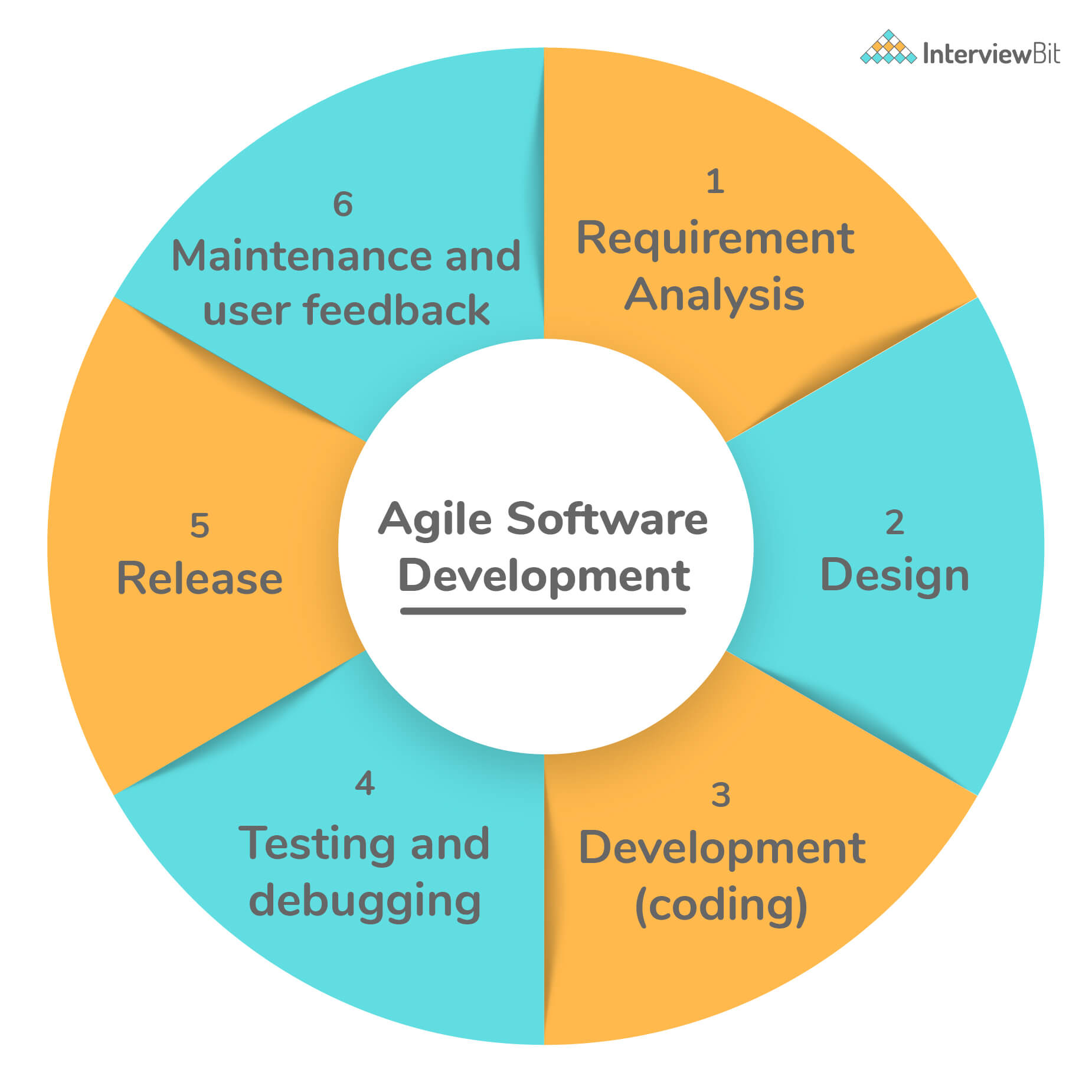
### 1. Explain the role of testing in software development?

Software testing comes into play at different times in different software development methodologies. There are two main methodologies in software development, namely Waterfall and Agile.

In a traditional waterfall software development model, requirements are gathered first. Then a specification document is created based on the document, which drives the design and development of the software. Finally, the testers conduct the testing at the end of the software development life cycle once the complete software system is built.

Waterfall Software Development Model

An agile software development model works in small iterations. You test the software in parallel as it is getting built. The developers build a small functionality according to the requirements. The testers test it and get customer feedback, which drives future development.



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

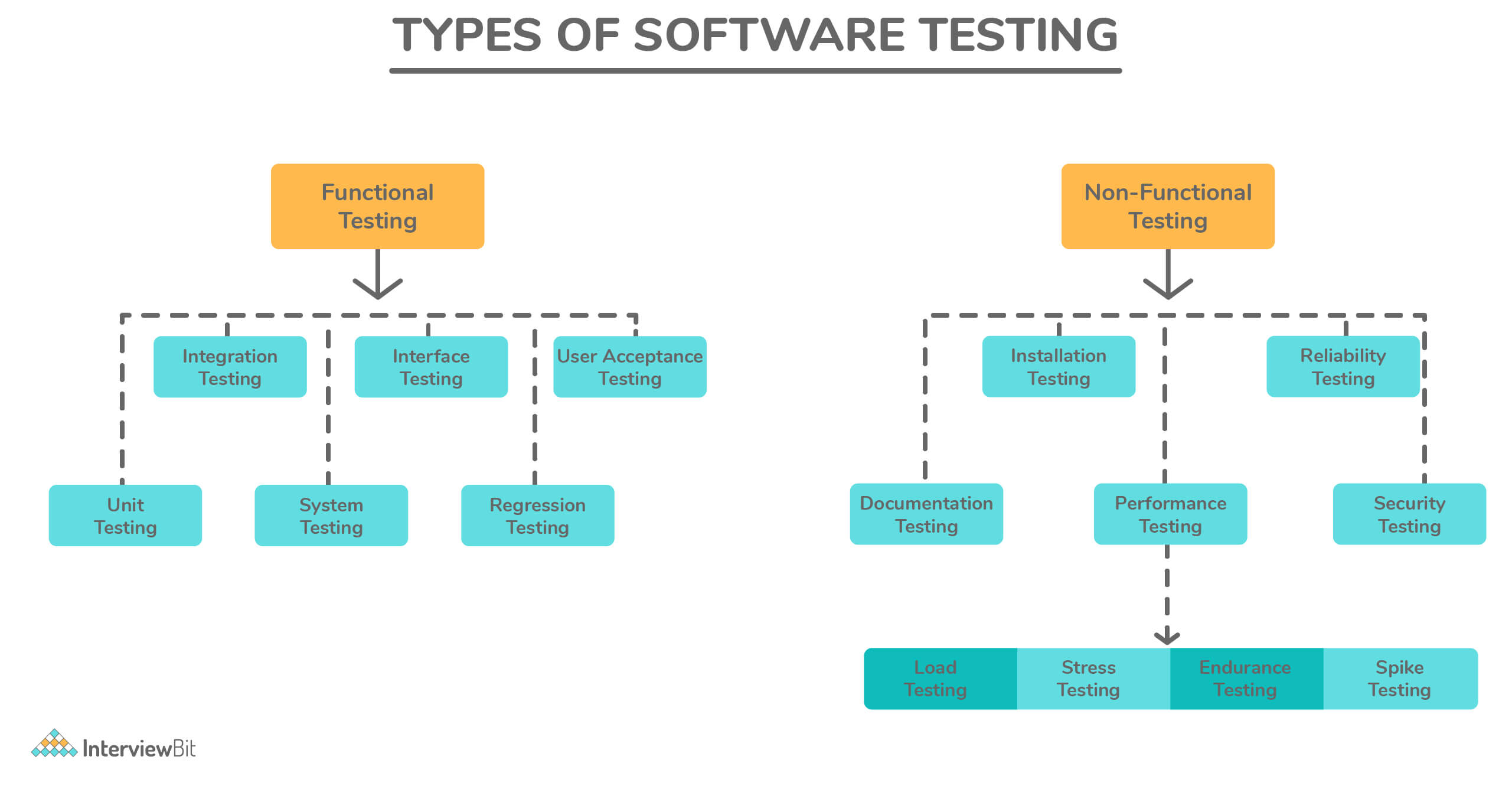
It is impossible to exhaustively test software or prove the absence of errors, no matter how specific your test strategy is.

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

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

### 3. What are the different types of testing?

You can test the software in many different ways. Some types of testing are conducted by software developers and some by specialized quality assurance staff. Here are a few different kinds of software testing, along with a brief description of each.



| **Type** | **Description** |
| --- | --- |
| Unit Testing | A programmatic test that tests the internal working of a unit of code, such as a method or a function. |
| Integration Testing | Ensures that multiple components of systems work as expected when they are combined to produce a result. |
| Regression Testing | Ensures that existing features/functionality that used to work are not broken due to new code changes. |
| System Testing | Complete end-to-end testing is done on the complete software to make sure the whole system works as expected. |
| Smoke Testing | A quick test performed to ensure that the software works at the most basic level and doesn’t crash when it’s started. Its name originates from the hardware testing where you just plug the device and see if smoke comes out. |
| Performance Testing | Ensures that the software performs according to the user’s expectations by checking the response time and throughput under specific load and environment. |
| User-Acceptance Testing | Ensures the software meets the requirements of the clients or users. This is typically the last step before the software is live, i.e. it goes to production. |
| Stress Testing | Ensures that the performance of the software doesn’t degrade when the load increases. In stress testing, the tester subjects the software under heavy loads, such as a high number of requests or stringent memory conditions to verify if it works well. |
| Usability Testing | Measures how usable the software is. This is typically performed with a sample set of end-users, who use the software and provide feedback on how easy or complicated it is to use the software. |
| Security Testing | Now more important than ever. Security testing tries to break a software’s security checks, to gain access to confidential data. Security testing is crucial for web-based applications or any applications that involve money. |

**You can download a PDF version of Software Testing Interview Questions.**

[**Download PDF**](javascript:void(0))

### 4. Why developers shouldn’t test the software they wrote?

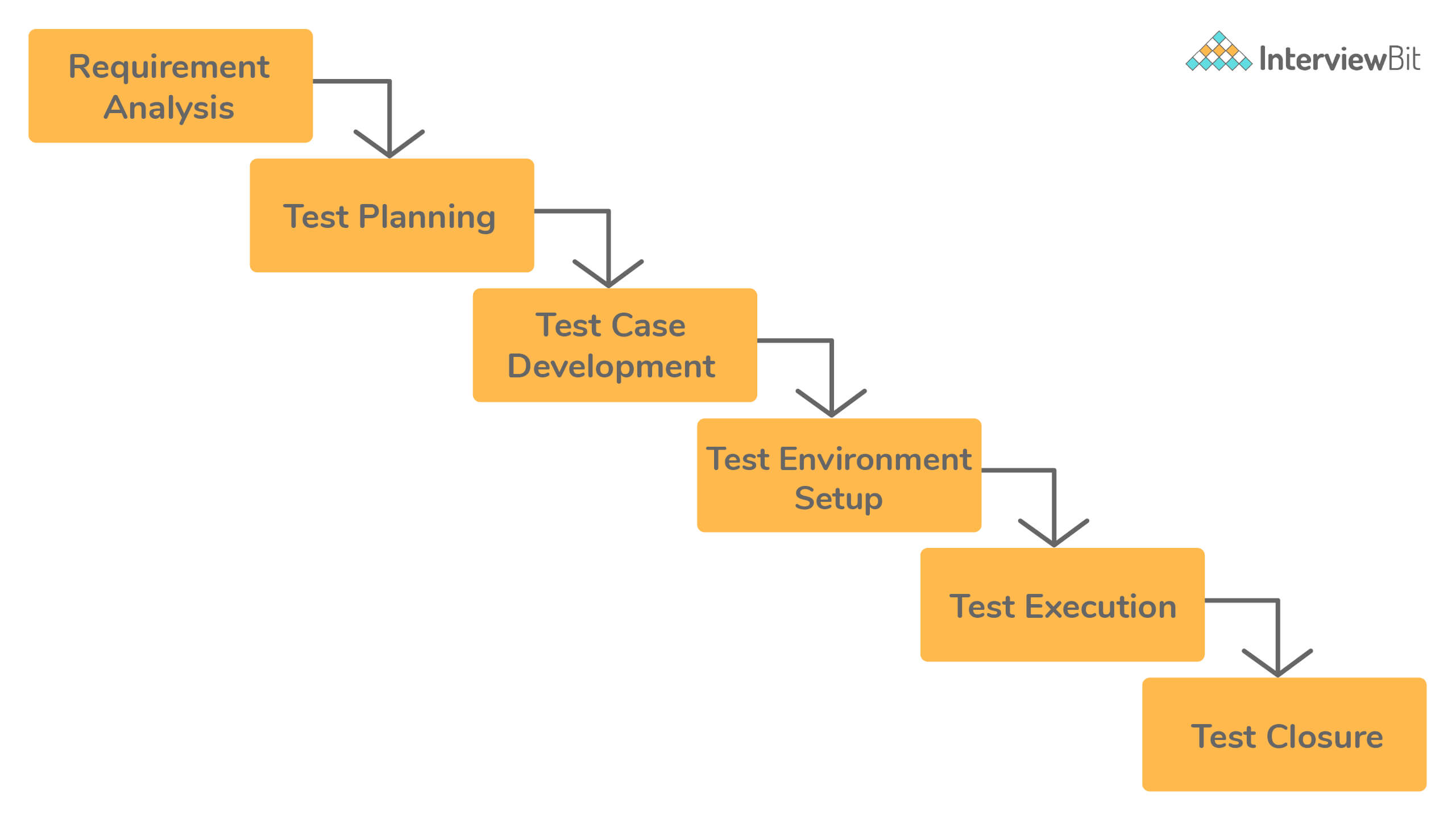
Developers make poor testers. Here are some reasons why:

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

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

### 5. What is the software testing life cycle?

Similar to software development, testing has its life cycle. During the testing, a tester goes through the following activities.



1. **Understand the requirements:** Before testing software or a feature, the tester must first understand what it is supposed to do. If they don’t know how the software is supposed to work, they can’t test it effectively.
2. **Test Planning and Case Development:** Once the tester has a clear understanding of the requirements, they can create a test plan. It includes the scope of testing, i.e., part of software under test and objectives for testing. Various activities are involved in planning the test, such as creating documentation, estimating the time and efforts involved, deciding the tools and platforms, and the individuals who will be conducting the tests.
3. **Prepare a test environment:** The development happens in a development environment, i.e., on a developer’s computer that might not represent the actual environment that the software will run in production. A tester prepares an environment with the test data that mimics the end user’s environment. It assists with realistic testing of the software.
4. **Generate the test data:** Though it is impossible to do exhaustive testing of the software, the tester tries to use realistic test data to give them the confidence that the software will survive the real world if it passes the tests.
5. **Test Execution:** Once the tester has a complete understanding of the software and has a test environment set up with the test data, they execute the test. Here, execution means that the tester runs the software or the feature under test and verifies the output with the expected outcome.
6. **Test Closure:** At the end of the test execution, there can be two possible outcomes. First, the tester finds a bug in the part of the software under test. In this case, they create a test record/bug report. Second, the software works as expected. Both these events indicate the end of the test cycle.

### 6. What qualities a software tester should have?

Any software tester's goal is to find out as many bugs and problems in the system so that the customers don't have to. Hence, a good software tester should have a keen eye for detail. They should know the ins and outs of the software they are testing and push every aspect of the software to its limits, to identify bugs that are hard to find with the software's regular use.

Having the domain knowledge of the application is essential. If a tester doesn't understand the specific problems the software is trying to solve, they won't be able to test it thoroughly.

A good tester should keep the end-user in mind when they are testing. Having empathy with the end-user helps the tester ensure that the software is accessible and usable. Simultaneously, the tester should possess basic programming skills to think from a developer's perspective, which allows them to notice common programming errors such as null-references, out-of-memory errors, etc.

Communication, both written and verbal, is an essential skill for a tester. A tester will frequently have to interact with both the developers and the management. They should be able to explain the bugs and problems found during testing to the developers. For each bug found, a good tester should provide a detailed bug report consisting of all the information a developer would need to fix that problem. They should be able to make a good case to the management if they are uncomfortable releasing the software if it contains unresolved issues.

### 7. What is functional testing?

[**Functional testing**](https://www.interviewbit.com/functional-testing-interview-questions/) is a form of black-box testing. As the name suggests, it focuses on the software's functional requirements rather than its internal implementation. A functional requirement refers to required behavior in the system, in terms of its input and output.

It validates the software against the functional requirements or the specification, ignoring the non-functional attributes such as performance, usability, and reliability.

Functional testing aims to answer the following questions, in particular:

* Does the software fulfill its functional requirements?
* Does it solve its intended users' problems?

### 8. What is a bug report?

During testing, a tester records their observations, findings, and other information useful to the developers or the management. All this data belongs to a test record, also called a bug report.

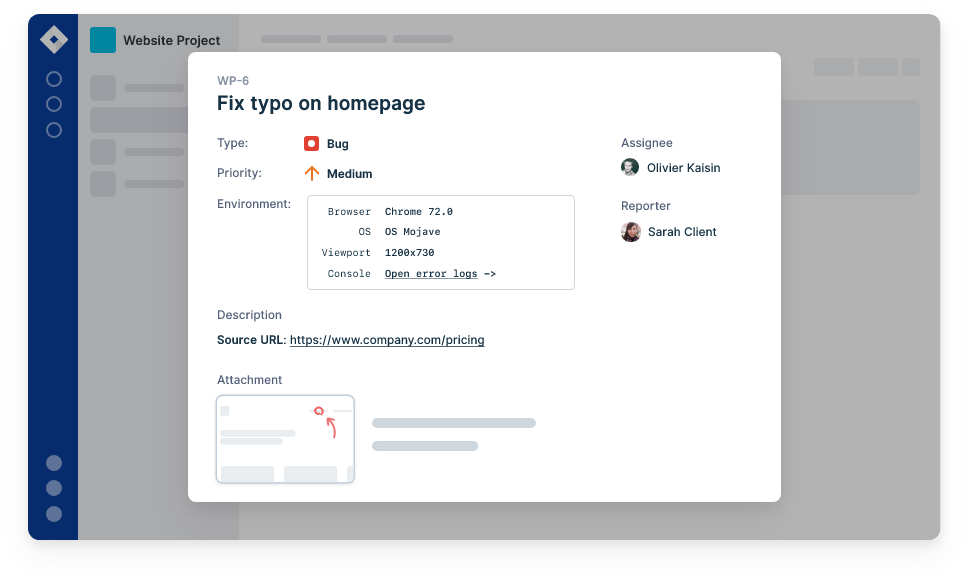
A detailed bug report is an important artifact produced during testing. It helps the team members with:

* Understand the problem,
* Steps to reproduce the problem,
* The environment and the specific conditions under which it happens, and
* The resolution if/when the developers fix the problem.

Here are a few bits of information that a good bug report should contain. *Image Source: Bugzilla*

| **Field** | **Description** |
| --- | --- |
| **Title** | A short headline that summarizes the problem. It shouldn’t be too long but just to give just the right information to the reader. It should be specific and accurate. |
| **Description** | The description should answer all the questions that are not explained by the title. It contains a detailed summary of the bug, its severity, and impact, steps to reproduce, expected results vs. the actual output. |
| **Version** | A lot of time can be wasted in trying to reproduce a bug in the wrong version of the product. Knowing the exact product version or the build number on which this bug was found is very useful to the developer in reproducing the bug. |
| **Status** | At any point, a bug can be either ‘Active’, ‘Ready for Testing’, or ‘Closed’. A bug becomes active when it is found, is ready for testing once the developer fixes it. A tester can mark it closed if the developer fixed it, or active if not. |
| **Steps to Reproduce** | Though the steps to reproduce the problem can be provided in the description, sometimes having a distinct field force the tester to think about them. They include each step one must take to successfully reproduce the problem. |
| **Assigned To** | Name of the developer or the tester to whom this bug is assigned. |
| **Resolution** | When a developer fixes the bug, they should include the cause for the bug and its resolution. It helps the team in the future when a similar bug resurfaces. |

For example, here is a picture of a bug reported on Jira, a popular bug-tracking software.



### 9. What is non-functional testing?

Non-functional testing tests the system's non-functional requirements, which refer to an attribute or quality of the system explicitly requested by the client. These include performance, security, scalability, and usability.

Non-functional testing comes after functional testing. It tests the general characteristics unrelated to the functional requirements of the software. Non-functional testing ensures that the software is secure, scalable, high-performance, and won't crash under heavy load.

### 10. What are some important testing metrics?

Testing metrics provide a high-level overview to the management or the developers on how the project is going and the next action steps.



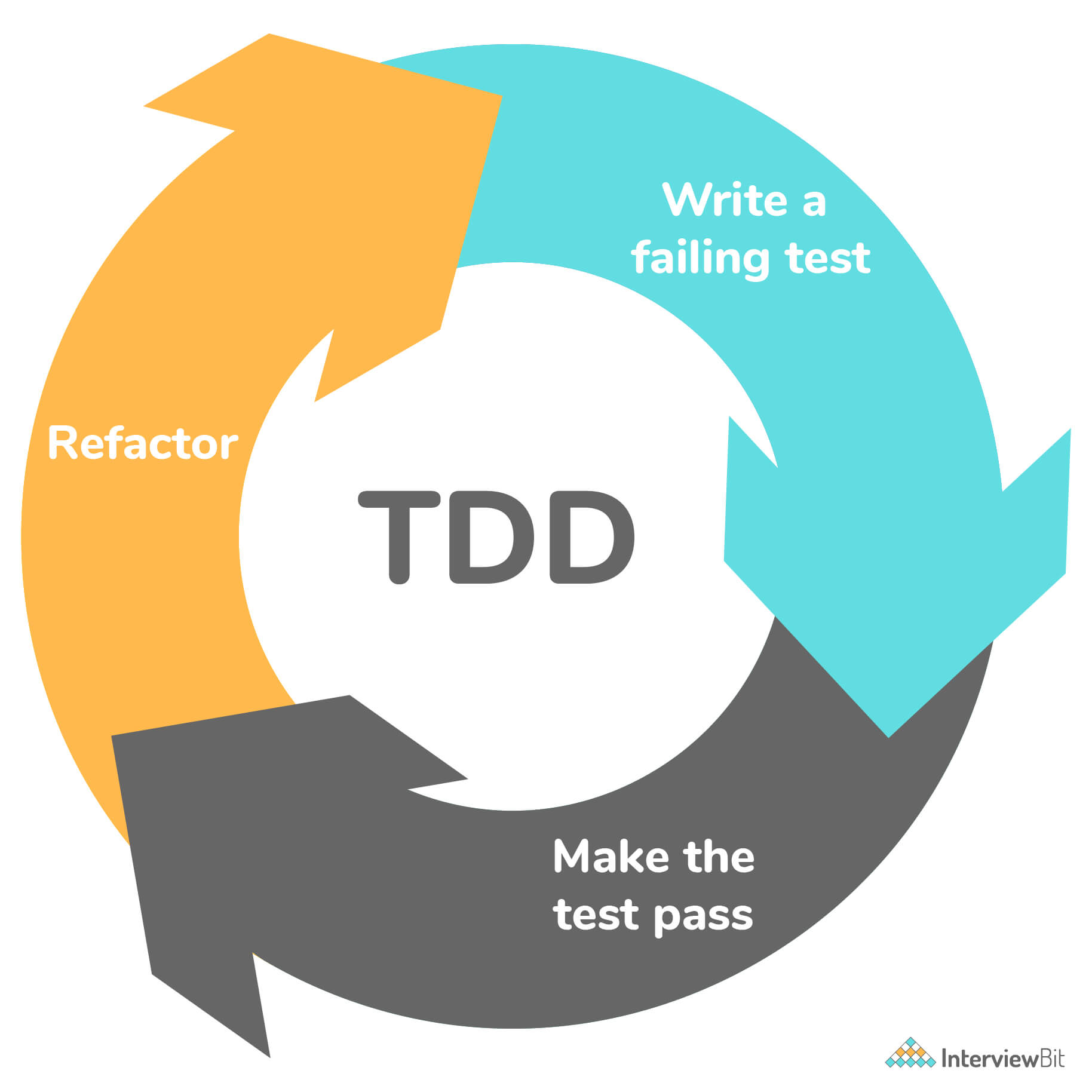
Here are some of the metrics derived from a record of the tests and failures:

* Total number of defects found, ordered by their severity
* Total number of bugs fixed
* Total number of problems caused by an error in the source code vs. configuration or external environmental factors
* Bug find and fix rate over time
* Bugs by produce/feature area
* The average time is taken by a bug since it’s found and fixed.
* Total time spent on new feature development vs. time spent on resolving bugs and failures
* Number of outstanding bugs before a release
* Bugs/failures reported by the customers vs. those found by the testers

### 11. What is Test-Driven-Development?

Test-Driven-Development (TDD) is a popular software development technique, first introduced by Kent Beck in his book with the same name, published in 1999.

In TDD, a developer working on a feature first writes a failing test, then writes just enough code to make that test pass. Once they have a passing test, they add another failing test and then write just enough code to pass the failing test. This cycle repeats until the developer has the fully working feature. If the code under the test has external dependencies such as database, files, or network, you can mock them to isolate the code.



**Benefits of TDD:**

* Writing tests first forces you to think about the feature you are trying to build, helping you produce better code.
* As you always have a working set of tests at hand, a failing test indicates that the problem is with the code you just added, reducing the time spent in debugging.
* Writing tests help the developer to clarify the requirements and specification. It’s challenging to write good tests for a poor set of requirements.
* It’s tough to produce high-quality software unless you can test the software after each new change. You can never be sure that your new code didn’t break the working software. TDD gives you the confidence to add new code, as you already have a test in place.

### 12. What is manual testing?

In manual testing, a tester manually verifies the functionality of the software. The tester has a comprehensive list of all the test cases they should test, along with the test data. They go through each case, one by one. They launch the software as an end-user would, enter the input, and manually verify the output.



It may seem that manual testing is inefficient when compared to automated testing. It is slow, not repeatable in a consistent manner, and prone to human misjudgment.

However, manual testing allows the tester to realistically test the software, using actual user data in a natural user environment, subject to similar external conditions. Only a human, not a computer, can evaluate the usability and accessibility of the application and how it looks and feels to the end-user. It also gives a broader perspective of the system. Finally, some test scenarios just can't be automated and need to be manually tested.

You should always test the software manually before trying to automate the testing.

### 13. What is cross-browser testing?

All web applications run in browsers such as Google Chrome, Mozilla Firefox, Internet Explorer, Safari, etc. Though they all work primarily the same in implementing the web standards, there are subtle differences in all of them. When building the software, it’s not always possible for the software developer to meticulously test the feature on multiple browsers, noticing the subtle inconsistencies.



In cross-browser testing, a software tester launches the web application in all the supported browsers and tries to test the same functionality on all of them. They note any unexpected behavior in a browser that doesn’t work as expected or looks different; note the behavior and the browser name and version in the test report. This helps the programmer to fix the behavior in all the browsers where it doesn't work as intended.

### 14. What is automated testing?

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

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

Once a test is automated, you can run it as often as you want, to check if any new code has broken it. It enables you to spend your time on other high-value tests, such as exploratory testing that help find bugs that an automated test would miss.

Automated testing is beneficial for repetitive testing with inputs that don’t change frequently. Humans get tired and bored from conducting the same tests repeatedly and seeing the same results. It’s easy to make mistakes when you are testing a feature for the twentieth time. Software is much better at doing repetitive tasks without getting tired or making mistakes than a human operator would.

### 15. What is the difference between Quality Control(QC) and Quality Assurance(QA)?

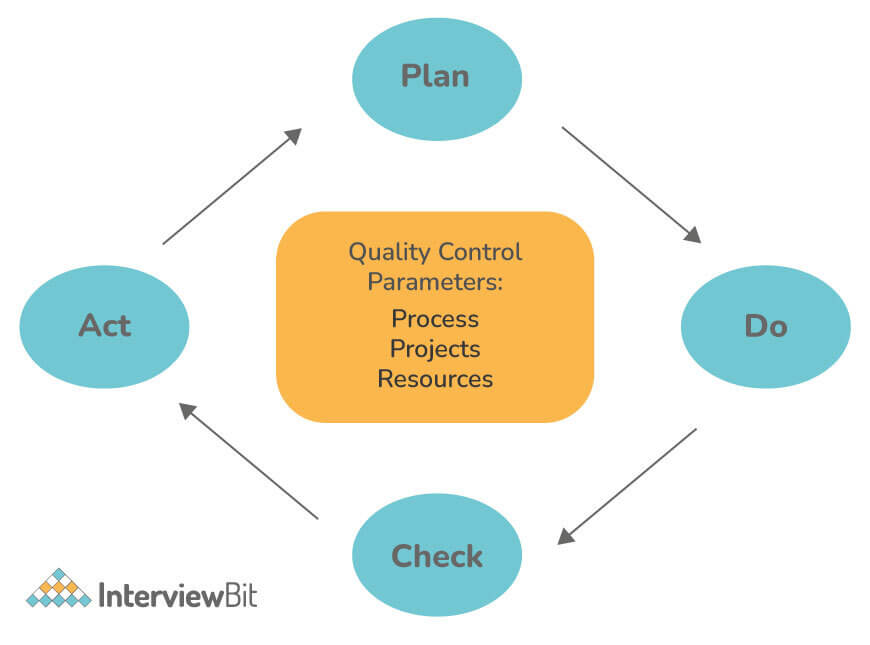
QA stands for **Quality Assurance.** In a software development team, a QA ensures that the software is thoroughly tested before releasing it to the end-users.



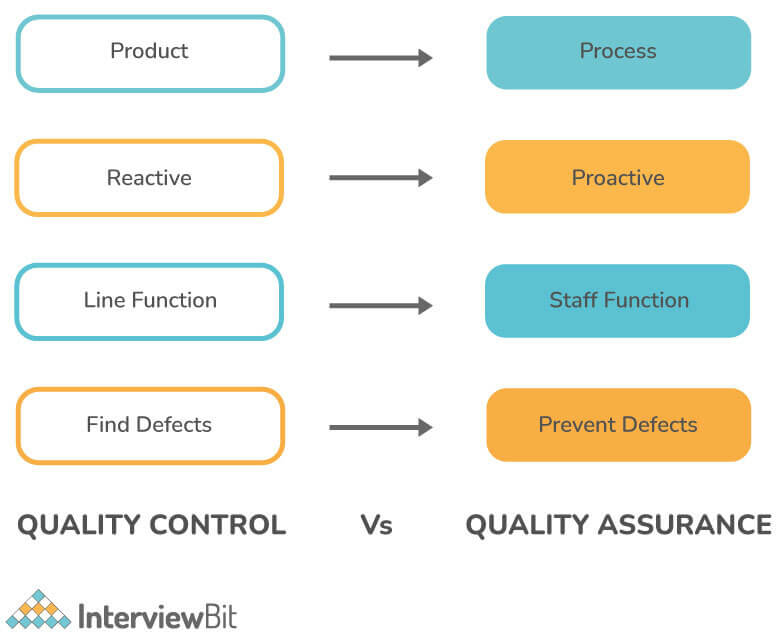
QA activities are generally performed while the product is being developed and focuses on improving the software development process.

In many software organizations, a tester and a QA can be the same person, but they can be different depending on the organization's size. The goal of the QA is to ensure quality in the shipped software.

**Quality Control**: QC stands for Quality Control. Its main aim is to ensure that the developed products meet the required standards or not.



QC is a process in software engineering that is used to ensure Software product quality by testing and reviewing its functional and non-functional requirements. QC activities are generally performed after the product is developed as it examines the quality of the end products and the final outcome.



### 16. What is a software bug?

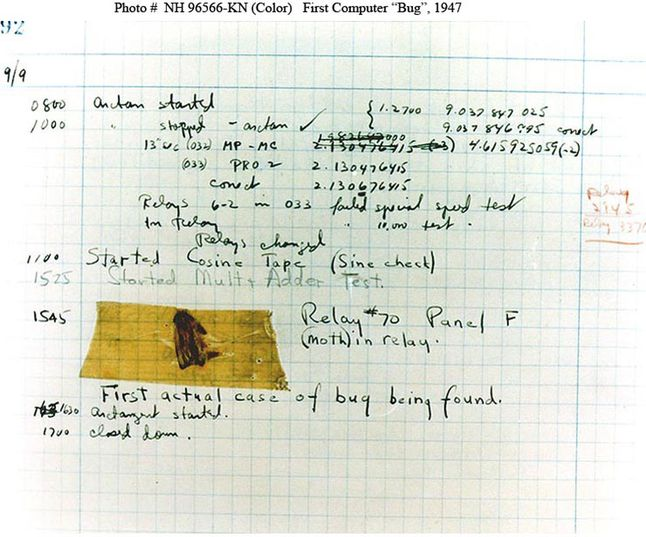
A software bug is an error in the software that produces wrong results. A software tester tests the software to find bugs in it.

There are many causes for the bugs—for example, poor design, sloppy programming, lack of version control, or miscommunication. Throughout development, developers introduce hundreds or thousands of bugs in the system. The goal of the tester is to uncover those bugs.

You can find a bug in many different ways, regardless of your role. When building the software, the software developer might notice the bug in another module, written by another developer or by themselves. The tester actively tries to find the bugs as part of a routine testing process. Finally, the users could see the bugs when the software is in production.

All bugs, no matter how they are found, are recorded into a bug-tracking system. A triage team triages the bugs and assigns a priority to the bug, and assigns the bug to a software developer to fix it. Once the developer resolves the problem, they check in the code and mark that bug as ready for testing. Once a bug is ready for testing, it goes to the tester, who tests the software to verify if it’s indeed fixed. If it is, then it’s closed. If not, they assign it to the same developer with a description of the exact steps to reproduce the bug. Some examples of popular bug-tracking systems include BugZilla, FogBugz, etc.

**Trivia:**  
The first software bug was discovered by Admiral Grace Hopper, on September 9, 1947. After they opened a malfunctioning piece of hardware, they found an insect stuck in the relay. *Image Source:*[*Link*](https://www.computerhistory.org/tdih/september/9/)

First Software Bug

### 17. Write some common mistakes that lead to major issues.

Some common mistakes include:

* Poor Scheduling
* Underestimating
* Ignoring small issues
* Not following the exact process
* Improper resource allocation

### 18. What is a user story?

All software has a target user. A user story describes the user's motivations and what they are trying to accomplish by using the software. Finally, it shows how the user uses the application. It ignores the design and implementation details.

A user story aims to focus on the value provided to the end-user instead of the exact inputs they might enter and the expected output.

In a user story, the tester creates user personas with real names and characteristics and tries to simulate a real-life interaction with the software. A user story often helps fish out hidden problems that are often not revealed by more formal testing processes.

### 19. What is a test environment?

A test environment consists of a server/computer on which a tester runs their tests. It is different from a development machine and tries to represent the actual hardware on which the software will run; once it’s in production.

Whenever a new build of the software is released, the tester updates the test environment with the latest build and runs the regression tests suite. Once it passes, the tester moves on to testing new functionality.

### 20. List some of the popular testing tools/frameworks, providing a brief description of each.

1. Selenium: a web browser automation tool that automates the test suites you need to run on a web browser.[1]
2. Protractor: An end-to-end test framework for Angular and AngularJS applications. Protractor runs tests against your application running in a real browser, interacting with it as a user would.[2]
3. Cypress: A modern front-end testing tool built for the modern web. Though it’s similar to Selenium and Protractor, it’s architecturally different from them.[3]
4. Jasmine: This is an open-source JavaScript testing framework that allows you to write behavior-driven tests.[4]
5. JUnit and NUnit: These are unit testing frameworks for Java and C# programming languages, respectively.

### 21. What are the different types of severity you can assign to a bug?

Though it varies depending on the size and structure of the software development teams, typically, a bug can be assigned the following types of severities, going from low to high:

* Low
  + User Interface bugs
  + Accessibility issues
* Medium
  + Leaky abstractions
  + Software hangs
  + Users unable to perform a specific action
  + Boundary conditions
* High
  + Crashing under high load
  + Business logic and/or calculation errors
  + Any user action that causes the software to crash
  + Exposing sensitive user data
  + Security problems
  + Loss of data

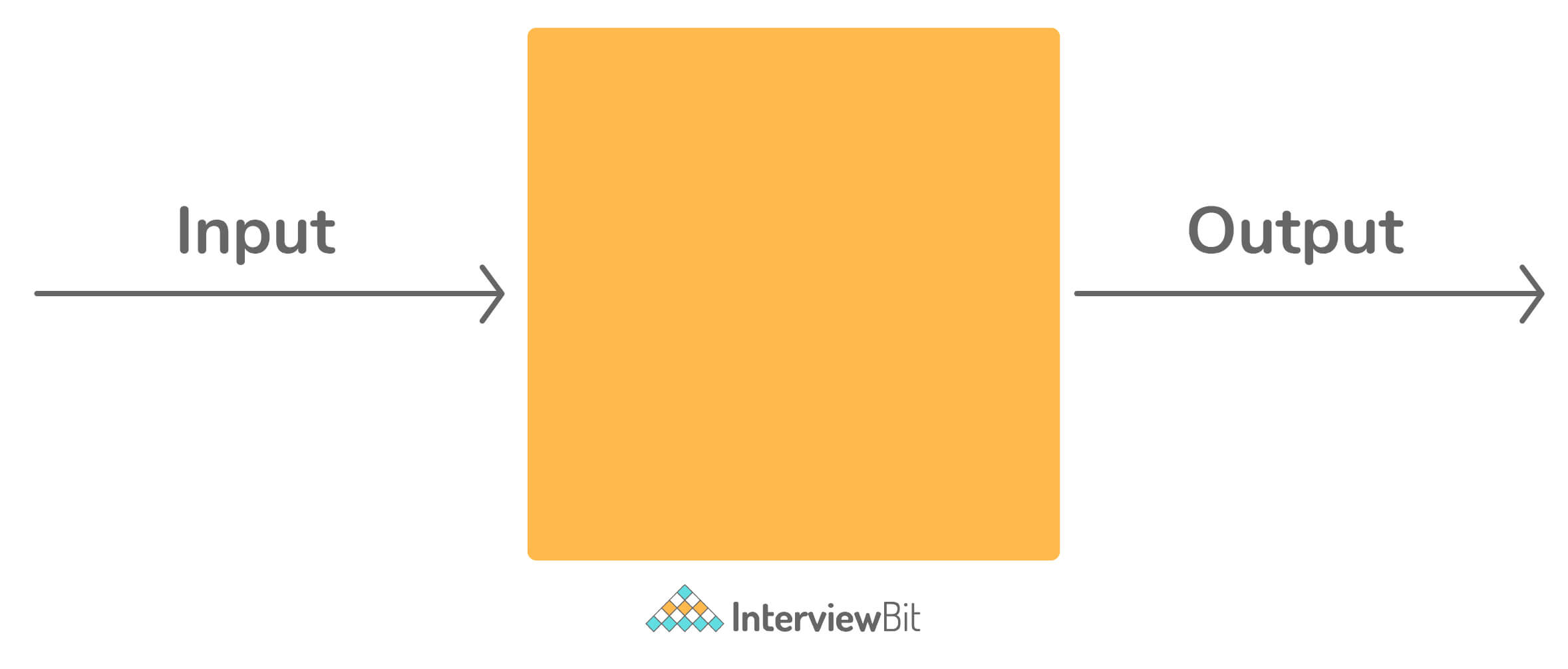
## Manual Testing Interview Questions for Freshers

### 22. What is black-box testing?

In black-box testing, the tester views the software as a black box, ignoring all the internal structure and behavior. Their only concern is the input provided to the system and the generated output. Black-box testing verifies the program’s behavior against the specified requirements.

During black-box testing, the test conditions are created based upon the software’s functionality but are unaware of how the software works internally. The software is tested from the end user’s perspective and gives a broader picture of the whole system.

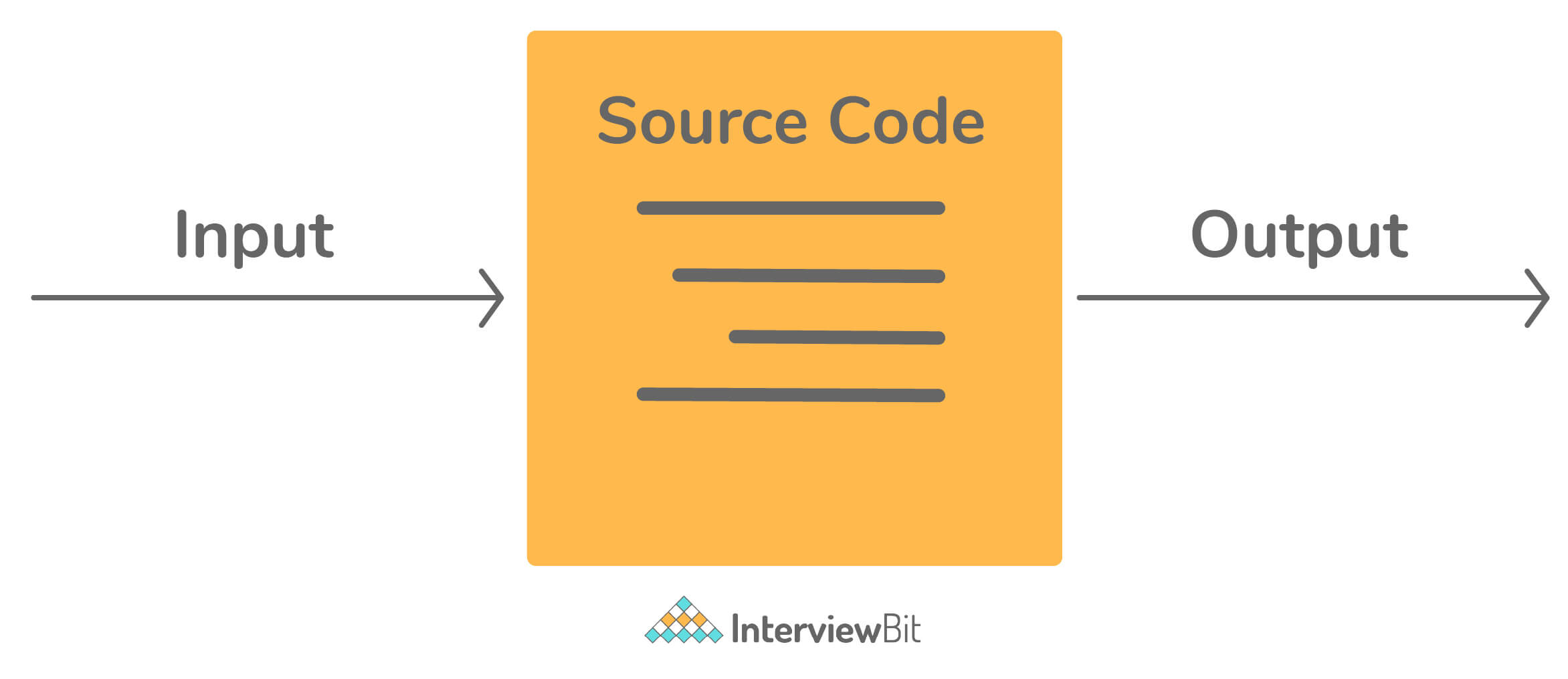
Given that the users are only concerned with whether the software works according to their needs and don’t care how it works, black-box testing helps test software usability and anticipate how the customer will use the product.



### 23. What is white-box testing?

White-box testing is an alternative strategy to black-box testing, in which a tester views the system as a transparent box. They are allowed to observe the internal implementation of the system, which guides the test. Typically, the software developers perform the white-box testing during the development phase.

In white-box testing, we assume that the tester has some programming knowledge. They try to test each possible branch a program could take in a running system. Knowing what’s inside the box, i.e., taking a look at the source code and the implementation details, it’s possible to test the system more thoroughly.



### 24. What are the differences between manual and automated testing?

| **Manual Testing** | **Automated Testing** |
| --- | --- |
| A human tester tests the software by manually running the test cases and observing and comparing the actual and expected outputs. | A tester or a programmer uses scripts and tools that execute the software and compares the actual and expected outputs. |
| Manual testing is not reproducible and repeatable. | Since it is programmed, automated testing is consistently reproducible and repeatable. It can be executed as many times as the tester wants. |
| For new features, a tester can quickly test the feature manually, without much configuration and setup. | To set up automated testing, there’s the initial investment required to write the tests and prepare an environment to run those tests on. |
| Manual testing is useful for finding bugs in the user interface or accessibility issues. | Automated testing is more suitable for catching bugs that a human tester would miss, such as programming bugs, business logic errors. |
| Manual testing is prone to human errors and is slow. | As there is no human participation involved (other than writing tests), automated testing is more reliable. It is much faster than manual testing. |

### 25. What is alpha testing?

Before you ship the software to the customers, the internal testing team performs alpha testing. Alpha testing is part of the user acceptance testing. Its goal is to identify bugs before the customers start using the software.

### 26. What is beta testing?

Once you ship the software to the customers after alpha testing, the software's actual users perform the beta testing in a real production environment. It is one of the final components of user acceptance testing. Beta testing is helpful to get feedback from real people using your software in real environments.

### 27. What is exploratory testing?

Imagine a tourist in a foreign city. There are two ways in which they can explore the city.

* Follow a map, itinerary, or a list of places they should visit
* Explore randomly, following the streets as they lead them to new places

With the first approach, the tourist follows a predetermined plan and executes it. Though they may visit famous spots, they might miss out on hidden, more exciting places in the city. With the second approach, the tourist wanders around the city and might encounter strange and exotic places that the itinerary would have missed.

Both approaches have their pros and cons.

A tester is similar to a tourist when they are testing software. They can follow a strict set of test cases and test the software according to them, with the provided inputs and outputs, or they can explore the software.

When a tester doesn't use the test scripts or a predefined test plan and randomly tests the software, it is called exploratory testing. As the name suggests, the tester is exploring the software as an end-user would. It's a form of black-box testing.

In exploratory testing, the tester interacts with the software in whatever manner they want and follows the software's instructions to navigate various paths and functionality. They don't have a strict plan at hand.

Exploratory testing primarily focuses on behavioral testing. It is effective for getting familiar with new software features. It also provides a high-level overview of the system that helps evaluate and quickly learn the software.

Though it seems random, exploratory testing can be powerful in an experienced and skilled tester's hands. As it's performed without any preconceived notions of what software should and shouldn't do, it allows greater flexibility to the tester to discover hidden paths and problems along those paths.

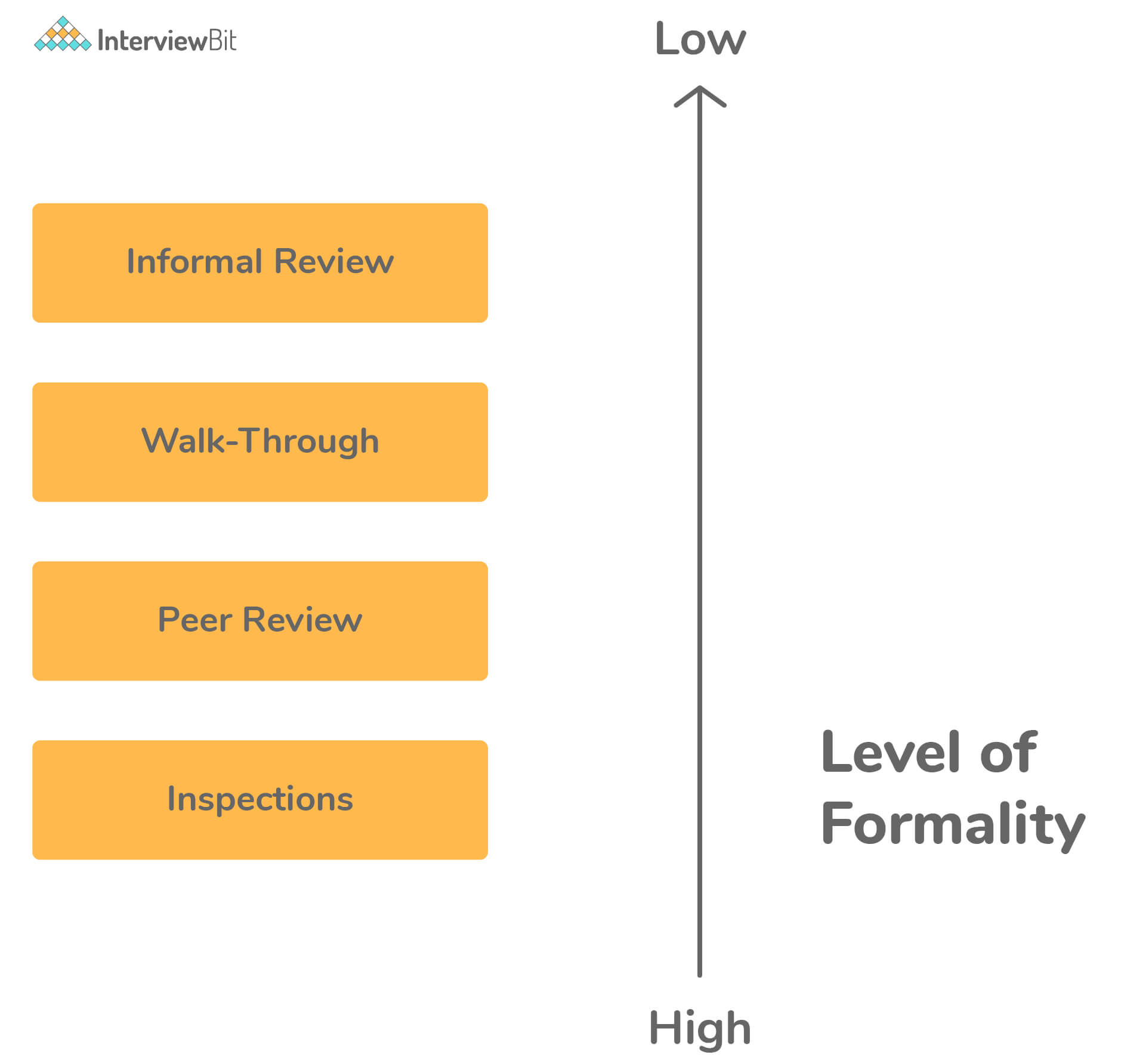
### 28. What is end-to-end testing?

End to End testing is the process of testing a software system from start to finish. The tester tests the software just like an end-user would. For example, to test a desktop software, the tester would install the software as the user would, open it, use the application as intended, and verify the behavior. Same for a web application.

There is an important difference between end-to-end testing vs. other forms of testing that are more isolated, such as unit testing. In end-to-end testing, the software is tested along with all its dependencies and integrations, such as databases, networks, file systems, and other external services.

### 29. What is static software testing?

Static testing is a technique in which you test the software without actually executing it. It involves doing code walkthroughs, code reviews, peer-reviews, or using sophisticated tools such as eslint, StyleCop to perform static analysis of the source code. Static testing is typically performed during software development.



### 30. What is dynamic software testing?

In contrast to static testing, dynamic software testing tests the software when it’s executing. The tester runs the software in a test environment and goes through all the steps involved, entering the inputs and verifying the actual output with the expected result.

### 31. What is API testing?

API testing ensures that the APIs that the software is using work as expected. The tester writes code that makes an API request to the server that provides the API, provides the required inputs, collects the output from the response, and matches the actual output with the expected output.

API testing primarily concerns the business logic of the software that’s exposing the API. It does not involve the look and feel, accessibility, or usability of the software. API testing can be automated to make it repeatable and reproducible each time they run.

### 32. What is meant by Code Coverage?

Code coverage is one of the important testing metrics. It indicates the ratio of the codebase under unit tests to the entire codebase. Code coverage of 50% means that the unit tests cover half of the codebase.

It is crucial to keep in mind that 100% code coverage doesn’t mean that the software is error-free or there are no bugs. It just means that the unit tests cover all the code. However, it’s still possible that the tests don’t test all branches that code could take or the problems with business logic execution.

### 33. What are the benefits of Manual Testing?

There are several advantages of manual testing as given below:

* Preferable for products with a short life cycle
* Saves time, money, and resources
* Ensure the error-free product
* Useful in ad-hoc testing, exploratory testing, and usability testing
* No need to change the entire code to make minor changes
* Get accurate user interface feedback
* Ability to handle difficult use case situations in a better way
* GUI testing can be done accurately
* Highly reliable
* Make user-friendliness better
* Easy to learn for new testers

### 34. What are the drawbacks of Manual Testing?

There are several drawbacks of manual testing as given below:

* Not suitable for time-bounded projects and large organizations
* More prone to human errors and mistakes
* Less efficient as the choice of recording the testing process is not available
* Less Reliable
* Regression testing is time-consuming
* Does not cover all the aspects of testing
* Load testing and performance testing can be performed manually
* More expensive in the long run process

### 35. What is the procedure for manual testing?

The process of manual testing includes the following:

* Planning and control
* Analysis and Design
* Implementation and Execution
* Evaluating exit criteria and Reporting
* Test closure activities

### 36. Write different types of Manual Testing?

Different types of manual testing include:

* Black Box Testing
* White Box Testing
* Unit Testing
* System Testing
* Integration Testing
* Acceptance Testing
* Performance Testing
* Load Testing

### 37. Name some of the manual testing tools.

Some of the top manual testing tools include:

* Postman
* Message queue monitors
* DB tools, etc.

### 38. When will you choose automated testing over manual testing?

Benefits of automated testing over manual testing

* When test cases need to be run repeatedly for a long duration of time.
* When one needs to perform regression testing, performance testing, or load testing.
* When one wants to record the testing process
* When one has less time to complete the testing phase
* When tests are needed to be executed in a standard runtime environment
* When tests include repetitive steps
* When there are multiple and quick deployments for the product, manual becomes very time taking and redundant

### 39. When will you choose manual testing over automated testing?

Benefits of manual testing over automated testing:

* When test cases need to be run for a short duration of time (once or twice).
* When one needs to perform ad-hoc testing, exploratory testing, or usability testing
* When one wants to ensure the user-friendliness of an application
* When flexibility is required
* When one wants to better handle the complex scenarios

### 40. Name some methods that can be used in code coverage.

There are several methods that can be used in code coverage includes:

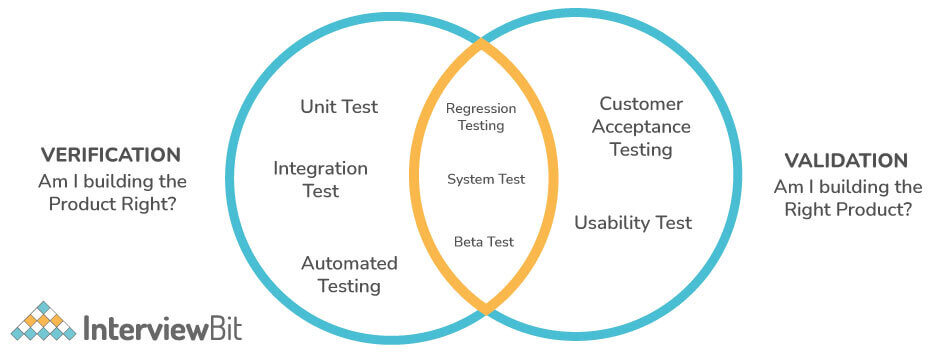
* Statement Coverage
* Decision Coverage
* Branch Coverage
* Toggle Coverage

### 41. Define Latent Defect.

Latent defect, as the name suggests, is a type of defect or bug which has been in the software system for a long time but is discovered now. A latent defect is an existing defect that can be found effectively with inspections. It usually remains hidden or dormant and is a low-priority defect.

### 42. Write difference between Validation and Verification?

**Validation:** It is defined as a process that involves dynamic testing of software products by running it. This process validates whether we are building the right software that meets that customer requirement or not. It involves various activities like system testing, integration testing, user acceptance testing, and unit testing.  
  
**Verification:** It is defined as a process that involves analyzing the documents. This process verifies whether the software conforms to specifications or not.  Its ultimate goal is to ensure the quality of software products, design, architecture, etc.

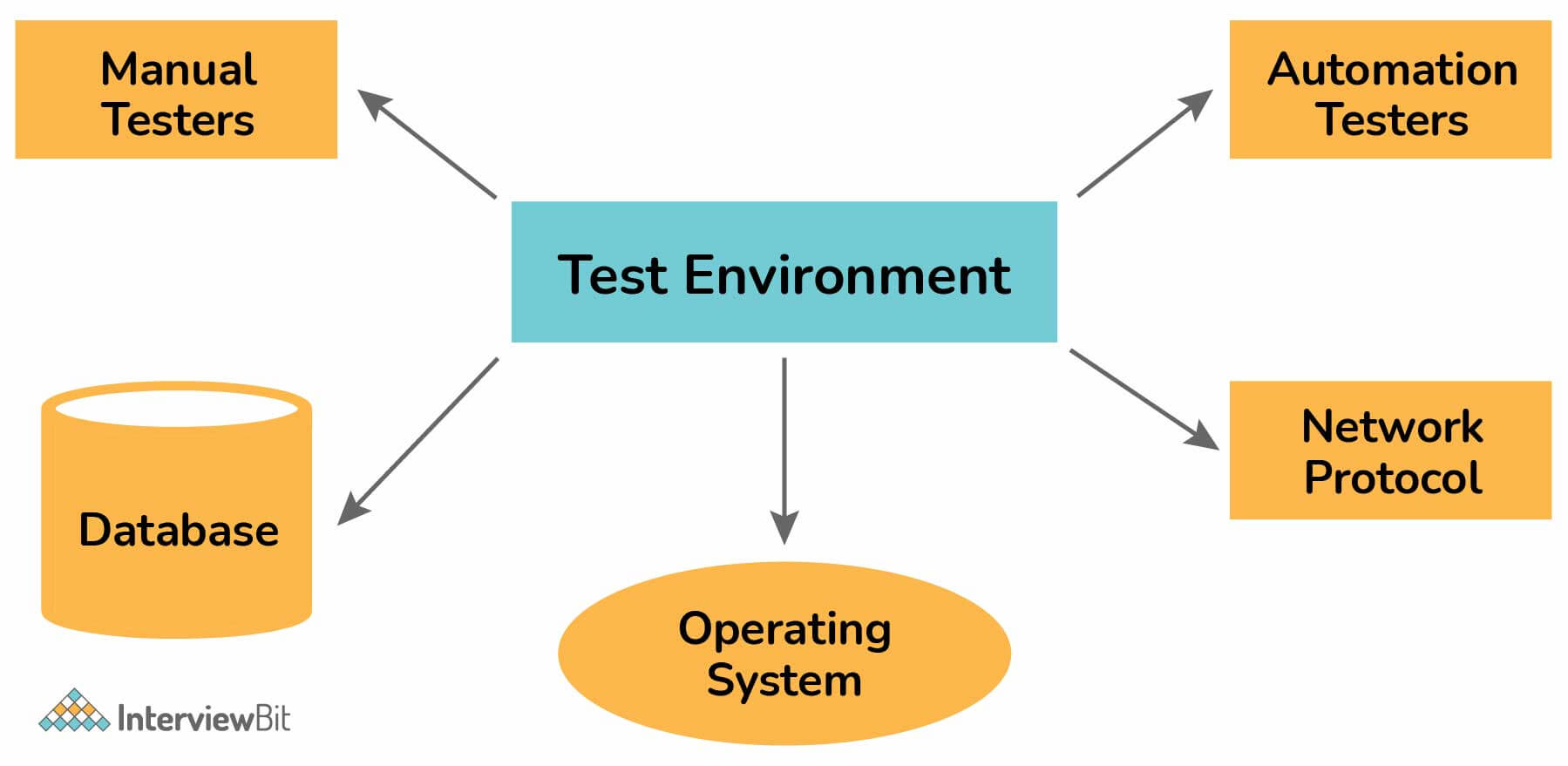


**Verification Vs Validation:**

| **Verification** | **Validation** |
| --- | --- |
| It checks whether the software meets the specification or not. | It checks whether the specification captures the customer’s needs or not. |
| It is a type of static testing. | It is a type of dynamic testing. |
| There is no requirement of executing the code. | There is a requirement for executing the code. |
| This process is performed by the QA team to make sure that the software is built as per the specifications in the SRS document. | This process is performed with the involvement of the testing team. |
| Reviews, walkthroughs, inspections, and desk-checking are some methods that can be used in verification. | Black box testing, white box testing, and non-functional testing are some methods that can be used during validation. |
| It identifies the bugs or errors early in the development process. | It can identify the bugs or errors that the verification process cannot catch. |
| It is performed before the validation process. | It is performed after the verification process. |

### 43. Explain the term testbed.

Testbed is generally referred to as a digital platform that is used for testing an application. It includes an operating system, hardware, network configuration, database, software application under test, and all other software-related issues.



### 44. What is the role of documentation in manual testing?

Some of the commonly applied documentation artifacts that are associated with software testing include:

* Test Plan
* Test Scenario
* Test Case
* Traceability Matrix

### 45. What do you mean by Test case?

Test case is basically a document that includes a set of test data, preconditions, expected results, and postconditions. This document is specially developed for a specific test scenario to ensure whether the software product meets the specific requirement or not.  In manual testing, test cases are executed manually by a tester without using any of the automated tools. One can easily identify loopholes in the specifications while developing test cases.

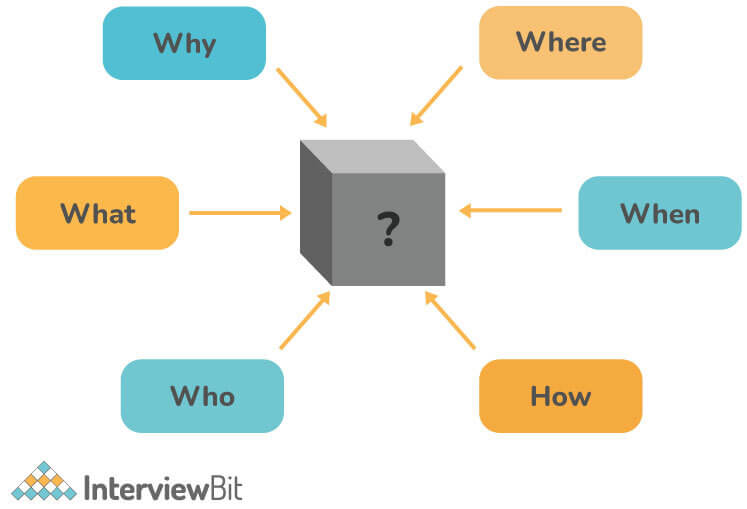
### 46. Name some attributes of the test case.

There are various attributes of test cases that make them more reliable, clear, and concise, avoiding any sort of redundancy. Some of them are given below:

* **Test Case Id**: Unique identifier of test case
* **Test Summary**: One-liner summary of test case
* **Description**: Detailed description of test case
* **Prerequisite or pre-condition**: Set of conditions to be followed before implementing the test steps
* **Test steps**: Detailed steps for performing test case
* **Test Data**: Test data value used in test case
* **Expected Result:** Estimated result to pass the test
* **Actual Result:** Actual result after executing the test steps.
* **Test Result:** Status of the test execution (Pass or Fail)
* **Automation Status:** Identifier for automation
* **Date:** Test execution date
* **Executed By:** Person name executing the test case

### 47. What is a Test Plan? What does it include?

A test plan is basically a dynamic document monitored and controlled by the testing manager. The success of a testing project totally depends upon a well-written test plan document that describes software testing scope and activities. It basically serves as a blueprint that outlines the what, when, how, and more of the entire test process.



A test plan must include the following details:

* Test Strategy
* Test Objective
* Test Scope
* Reason for Testing
* Exit/Suspension Criteria
* Resource Planning
* Test Deliverables.

### 48. What is a Test Report? What does it include?

Test report is basically a document that includes a total summary of testing objectives, activities, and results. It is very much required to reflect testing results and gives an opportunity to estimate testing results quickly. It helps us to decide whether the product is ready for release or not. It also helps us determine the current status of the project and the quality of the product. A test report must include the following details:

* Test Objective
* Project Information
* Defect
* Test Summary

### 49. What do you mean by Test Deliverables?

Test deliverables, also known as test artifacts, are basically a list of all of the documents, tools, and other components that are given to the stakeholders of a software project during the SDLC. Test deliverables are maintained and developed in support of the test. At every phase of SDLC, there are different deliverables as given below:



**Before Testing Phase**

* Test plans document.
* Test cases documents
* Test Design specifications.

**During Testing Phase**

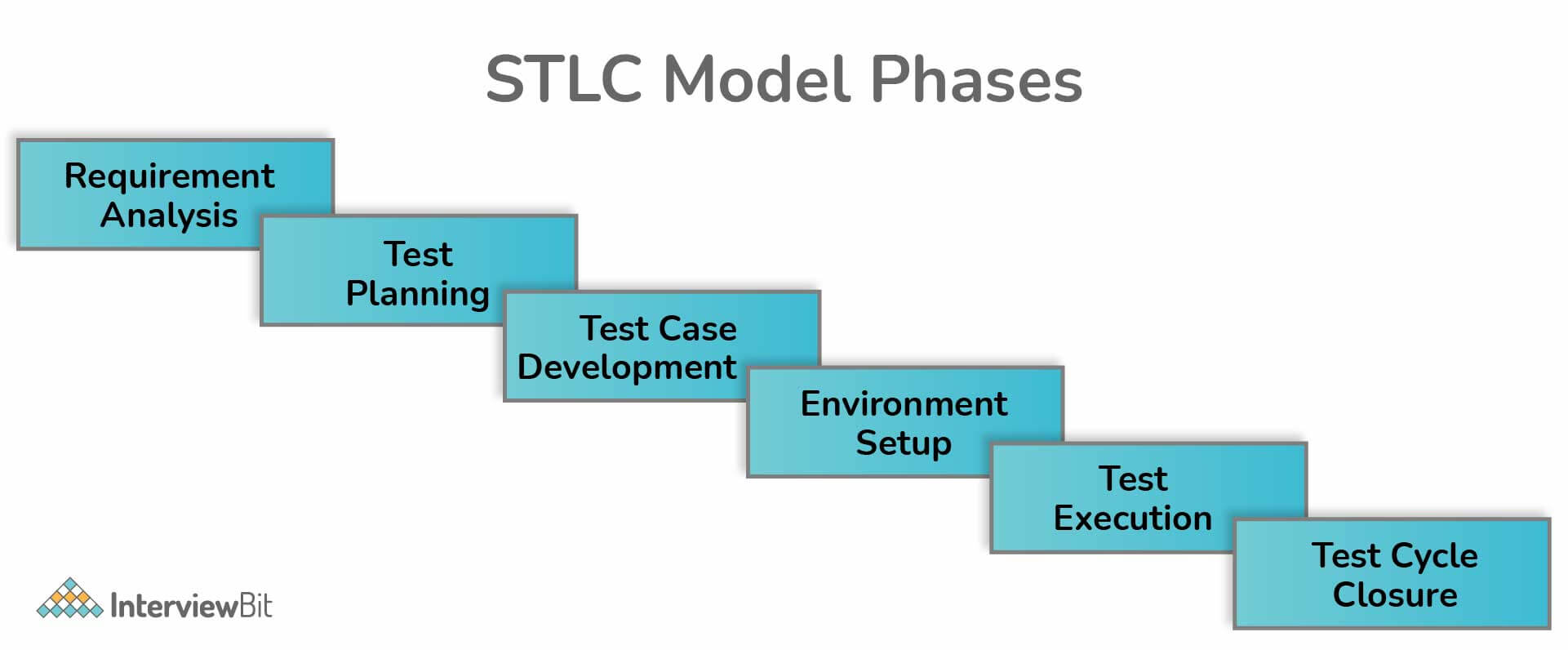
* Test Scripts
* Simulators.
* Test Data
* Test Traceability Matrix
* Error logs and execution logs

**After testing Phase**

* Test Results/reports
* Defect Report
* Installation/ Test procedures guidelines
* Release notes

### 50. Explain STLC.

STLC (Software Testing Life Cycle) is a fundamental part of SDLC which is used to test software and ensure that the quality standards are met. It generally involves both verification activities and validation activities. In this, different activities are executed in a specific order throughout the software testing process. There are basically six different phases in STLC Model as shown below:



* Requirement Analysis
* Test Planning
* Test Case Development
* Test Environment Setup
* Test Execution
* Test Cycle Closure

### 51. Write difference between bug, defect, and error.

**Error**: It is defined as a programming mistake in coding because of which we can't compile or run a program.    
  
**Defect**: It is defined as the variation or difference between the actual result and the expected result founded by a tester or developer. The defect is basically detected after the product goes into production and is resolved in the development phase only.    
  
**Bug**: It is defined as a fault or mismatch in a software system that is detected during the testing phase. It has an impact on software functionality and performance.

### 52. Explain use-case testing.

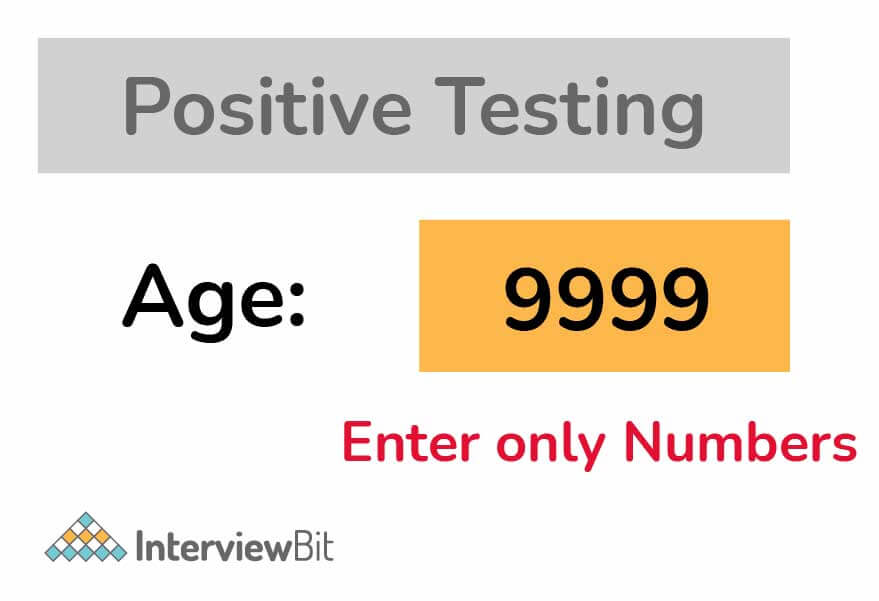
Use case testing is basically defined as a technique that helps developers and testers to identify test cases that exercise the whole system on each transaction basis right from start to finish. It is a part of black-box testing that is used widely in developing tests or systems for acceptable levels.

### 53. What do you mean by Test Matrix and Traceability Matrix?

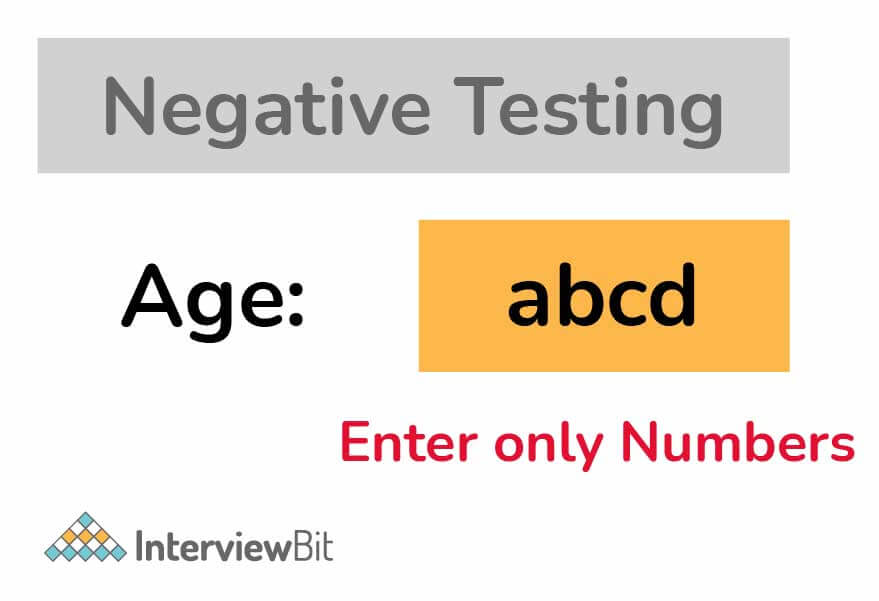
**Test Matrix:** It is referred to as a testing tool that is used to capture actual quality, effort, resources, plan, and time required to capture all the phases of software testing. It only covers the testing phase of the life cycle.    
  
**Requirement Traceability Matrix (RTM):** It is referred to as a document, usually present in the form table, that is used to trace and demonstrate the relationship between the requirements and other artifacts of the project right from start to end. In simple words, it maps between test cases and customer requirements.

### 54. What is Positive and Negative Testing?

**Positive Testing**: It is a type of testing process where the software application is validated against the valid data sets as an input. It is simply used to check whether the application does what it is supposed to do or not.



**Negative Testing**: It is a type of testing process where the software application is validated against invalid data sets as an input. It is simply used to check whether the system shows an error when it is supposed to do or not. In test case execution, negative testing is considered a very crucial factor.



**Positive vs Negative Testing:**

| **Positive Testing** | **Negative Testing** |
| --- | --- |
| It tests the application or system by giving valid data. | It tests the application or system by giving invalid data. |
| It accepts all the numeric and alphabetic values. | It does not accept any special character. |
| This type of testing is performed to identify a known set of test conditions. | This type of testing is performed to identify an unknown set of test conditions. |
| It is usually performed on each and every application. | It is usually performed where the chances of unexpected conditions or errors are more. |
| It requires less time and can be performed by people having less knowledge. | It requires more time and can only be performed by professionals. |
| It makes sure that the software application is normal. | It makes sure that the software applications are 100% detect-free. |
| It does not encompass all the possible cases. | It encompasses all the possible cases. |
| It is less significant or vital than negative testing. | It is more significant and vital than positive testing. |

### 55. What do you mean by Critical bug?

A critical bug is referred to as a bug that affects the majority of the functionality of the given application. It does not disturb or block the testing of other parts of the given application. A bug is only classified as critical when it halts the core operations of the software.

### 56. What is UAT (User Acceptance Testing)?

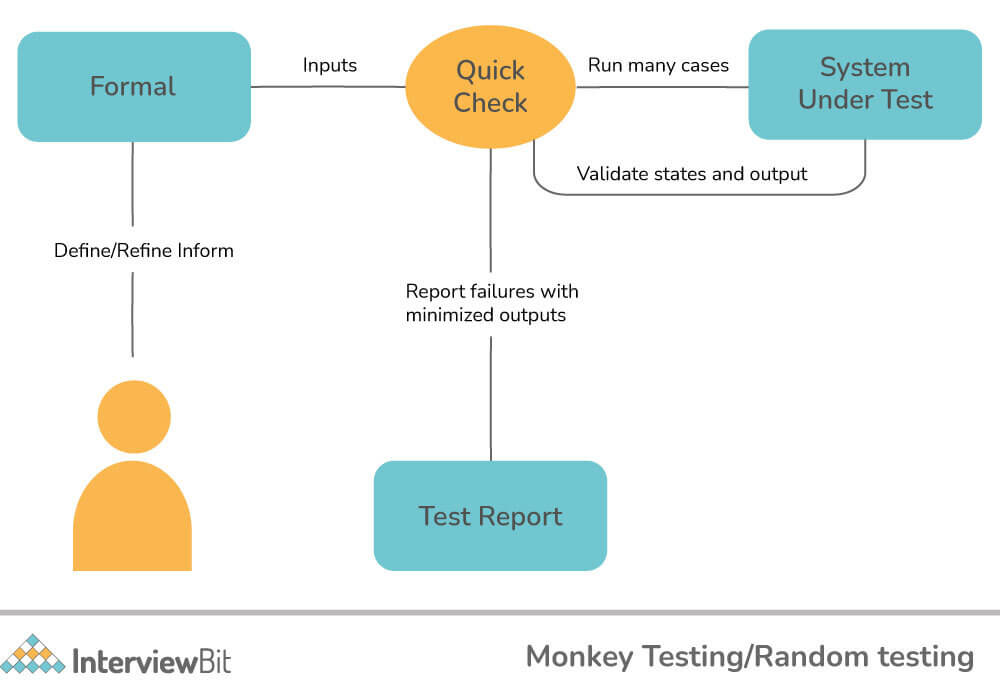
UAT, also known as end-user testing, is basically a testing methodology performed by the end-users or clients for the approval of the production release. It is one of the last stages of the SDLC and is only performed after the software has been tested thoroughly. Its main purpose is to validate end-to-end business flow. It makes sure that the developed software meets all the requirements to float into the market or not.

### 57. System testing can be done at any stage?

All the components of the software are generally tested as a whole to make sure that the overall product fulfills the requirements that are specified. So, no. system testing cannot be done at any stage instead system testing must start only when all modules or units work correctly and are in place, but it needs to be performed before UAT.

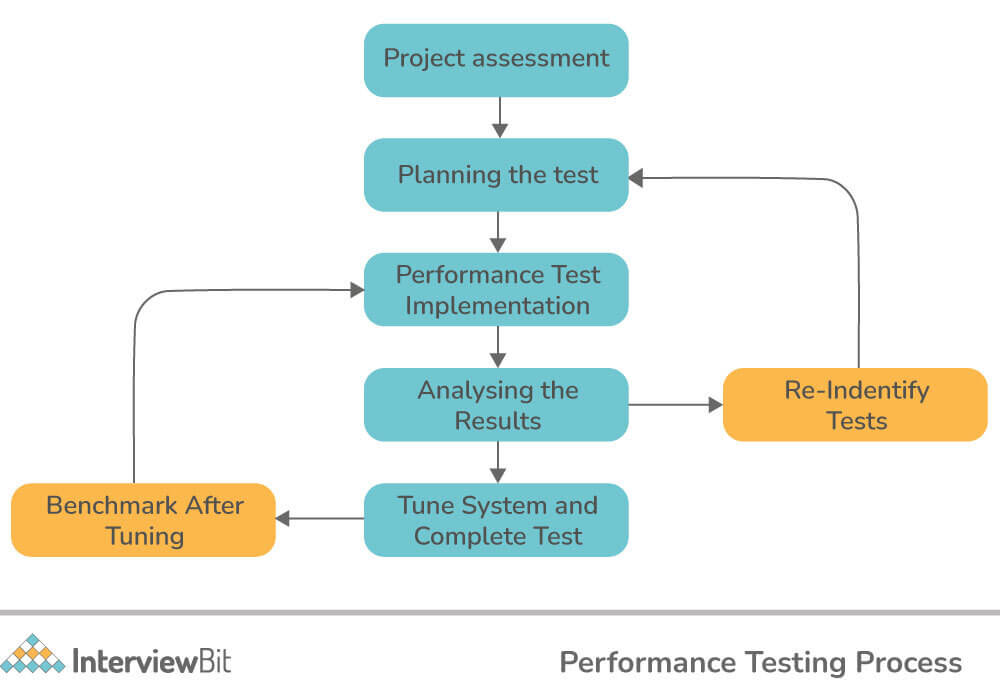
### 58. Explain Monkey Testing and Performance Testing.

**Monkey Testing**: Monkey testing, also known as Random Testing, is a type of software testing technique in which data is generated randomly using a tool or some automated mechanism. The system is tested using this randomly generated input, and results are analyzed accordingly. There are no such rules in this type of testing.



**Performance Testing**: It is a type of non-functional software testing technique that is used to determine the system parameters like speed, scalability, and stability under different workload conditions. Its main purpose is to eliminate performance bottlenecks, not to find bugs. Some of the key parameters of performance testing include:

* CPU Utilization
* Memory Utilization
* QPS/TPS (Transaction per second)
* Average load time
* System throughput, etc.



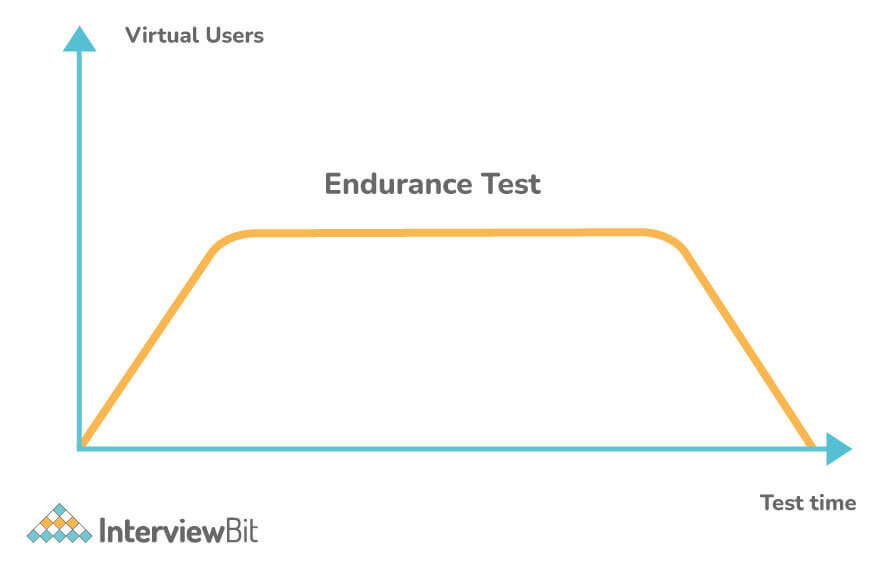
### 59. Write the difference between Test Stub and Test Driver.

Test driver and test stub, both are types of test harness that are used to provide a simulation environment for testing a module or component. They both are dummy modules specially created for test purposes.   
  
**Test stubs**: Test stubs are used in a top-down testing approach and allow testing of the upper levels of the code when the lower levels of the code are not developed yet. It is used as ‘called programs’ when subprograms are under construction.   
  
**Test drivers**: Test drivers are used in a bottom-up testing approach and allow testing of the lower levels of the code when the upper levels of the code are not developed yet. It is used as ‘called programs’ when main programs are under construction.

### 60. Explain Endurance Testing or Soak Testing?

Endurance testing, also known as Soak testing, is a type of performance testing usually performed to check the performance of the system that is under constant use. Its main purpose is to determine whether a system can sustain a continuous high load or not. Memory utilization is also monitored to identify potential leaks during this testing. Some of the endurance testing tools include:

* WebLOAD
* LoadUI
* OpenSTA
* LoadComplete
* Apache JMeter, etc.

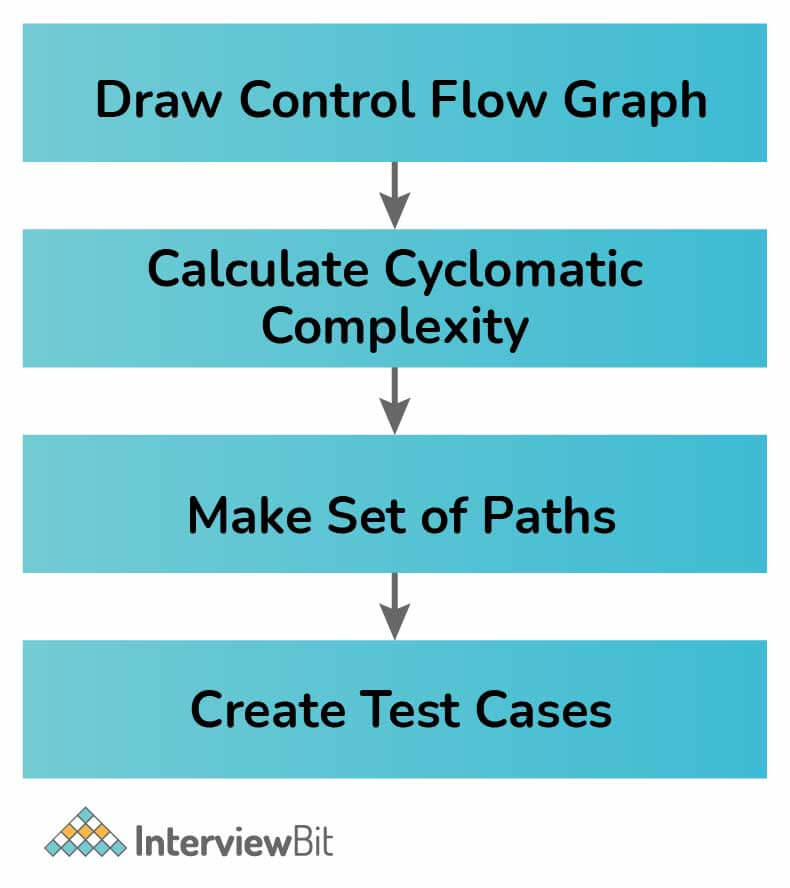


### 61. What is the importance of Localization Testing?

Localization testing is a type of testing that is performed to ensure whether the software product offers full functionality and usability in a particular locale or not. It is simply used to verify the accuracy and suitability of the content. It is not only about linguistics, but also about traditions, common herd behavior, and other similar factors. It generally deals with the functionality and GUI of the application.

### 62. Explain Path testing.

Path testing is a type of testing specially used to design test cases. In this type of testing, the control flow graph of a program is specially designed to identify a set of linearly independent paths of execution. Its main objective is to ensure that each path is covered and executed well. It also reduces or minimizes the occurrence of redundant tests.

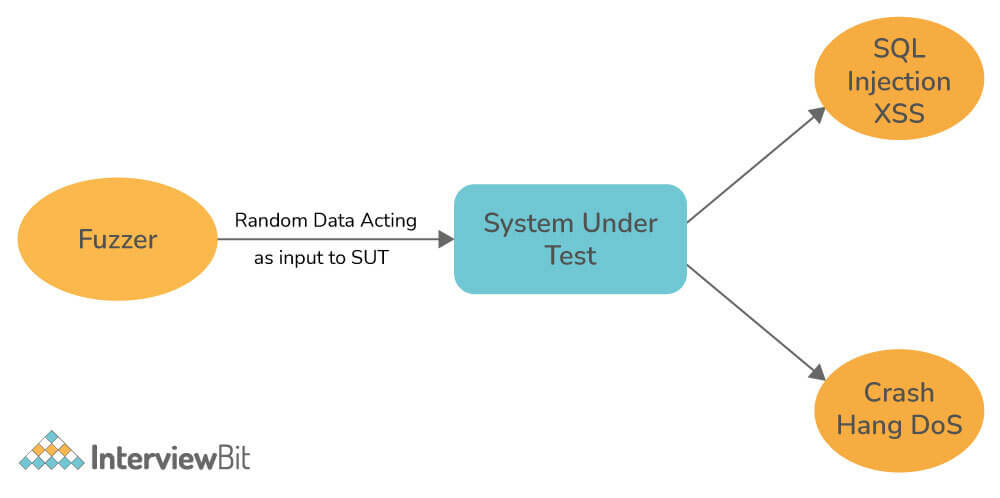


### 63. What do you mean by Baseline Testing and Benchmark testing?

**Baseline Testing**: It is a type of non-functional testing in which a set of tests are run to capture performance information. Using this gathered information, we can make required changes in the application and ultimately improve the performance and capabilities of the application. In general, it refers to a benchmark that usually forms the base of any new creation. During this testing, many errors are discovered and resolved.   
  
**Benchmark Testing**: It is a type of testing that involves both the developers and DBAs (Database Administrators) to determine current performance information. Using this information, one can improve the performance of the same by matching it with the benchmarks (industry standards). Its main objective is to compare the present and future software releases with their specific benchmark.

### 64. Explain Fuzz Testing and its importance.

Fuzz testing is a software testing technique in which a large amount of random data, called fuzz, is provided as input to the application simply to detect or discover security loopholes and coding errors in the software application. It is more useful for bigger projects but it only detects a serious fault or defect. It is simply used to check the vulnerability of software and gives more effective results when used with beta testing, black box testing, etc.



### 65. What do you mean by Data flow testing?

Data flow testing is a type of structural testing that is used to analyze the flow of data in the program. In this, a programmer can perform various tests on data values and variables. Using this testing, one can determine the variables that are used at every stage of the program’s control flow. It helps us in the following ways:

* Eliminate or remove variables that are never used after being declared
* Pinpoint variables that are used but never declared
* Deallocate variable before it is used
* Pinpoint variables that are defined multiple times before it is used

### 66. What is the importance of agile testing?

Agile testing is basically a software testing process that uses agile methodologies i.e., follow test-first design paradigm. It evaluates the software from the customer's point of view. It does not require any development team to complete coding for starting QA, instead of testing and coding can go hand in hand. Features are tested as they are developed. Some of its advantages include:

* Boosts the performance as it allows all the testers and developers to work together
* Features are tested as they are developed
* Ensure successful delivery of the high-grade product
* Saves time and money
* Flexible and highly adaptable to changes
* Improves product quality and allows developers to release software as early as possible

### 67. What are different categories of debugging?

Different categories of debugging include:

* Brute force debugging
* Backtracking
* Cause elimination
* Program slicing
* Fault tree analysis

## Manual Testing Interview Questions For Experienced

### 68. What is Selenium? What are its benefits?

[Selenium](https://www.interviewbit.com/selenium-interview-questions/) is a web browser automation tool that automates the test suits you need to run on a web browser.

Some of the benefits of Selenium include:

* It is open-source software, eliminating licensing costs.
* It supports all the major languages, such as Java, C#, Python, Ruby, etc.
* It supports all the major web browsers, e.g., Google Chrome, Firefox, Safari, etc.
* You can integrate it with other testing frameworks and tools to build a comprehensive test suite for your software.

### 69. What is boundary value analysis?

In software, many errors occur near the edges of the range of the data values. For example, when the programmer uses the greater-than operator (>) instead of the greater-than-or-equal-to (>=) operator, it causes the off-by-one indexing error.

Typically, developers miss these boundary cases because they follow a happy path when developing and testing. Boundary value analysis helps to discover the errors caused by extreme values. The tester chooses the test data at and immediately above and below the boundaries of the input domain of the data.

For example, if an input field expects a string of 20 characters long, the tester tests it with strings of lengths 19, 20, and 21.

### 70. What is regression testing?

The dictionary definition of regression is the act of going back to a previous place or state. In software, regression implies that a feature that used to work suddenly stopped working after a developer added a new code or functionality to the software.

Regression problems are pervasive in the software industry, as new features are getting added all the time. Developers don't build these features in isolation, separate from the existing code. Instead, the new code interacts with the legacy code and modifies it in various ways, introducing side effects, whether intended or not.

As a result, there is always a chance that introducing new changes may negatively impact a working feature. It's important to keep in mind that even a small change has the potential to cause regression.

Regression testing helps ensure that the new code or modifications to the existing code don't break the present behavior. It allows the tester to verify that the new code plays well with the legacy code.

### 71. What is unit testing?

Unit testing is the process of testing a single unit of code in an isolated manner. The unit of code can be a method, a class, or a module. Unit testing aims to focus on the smallest building blocks of code to get confidence to combine them later to produce fully functioning software.

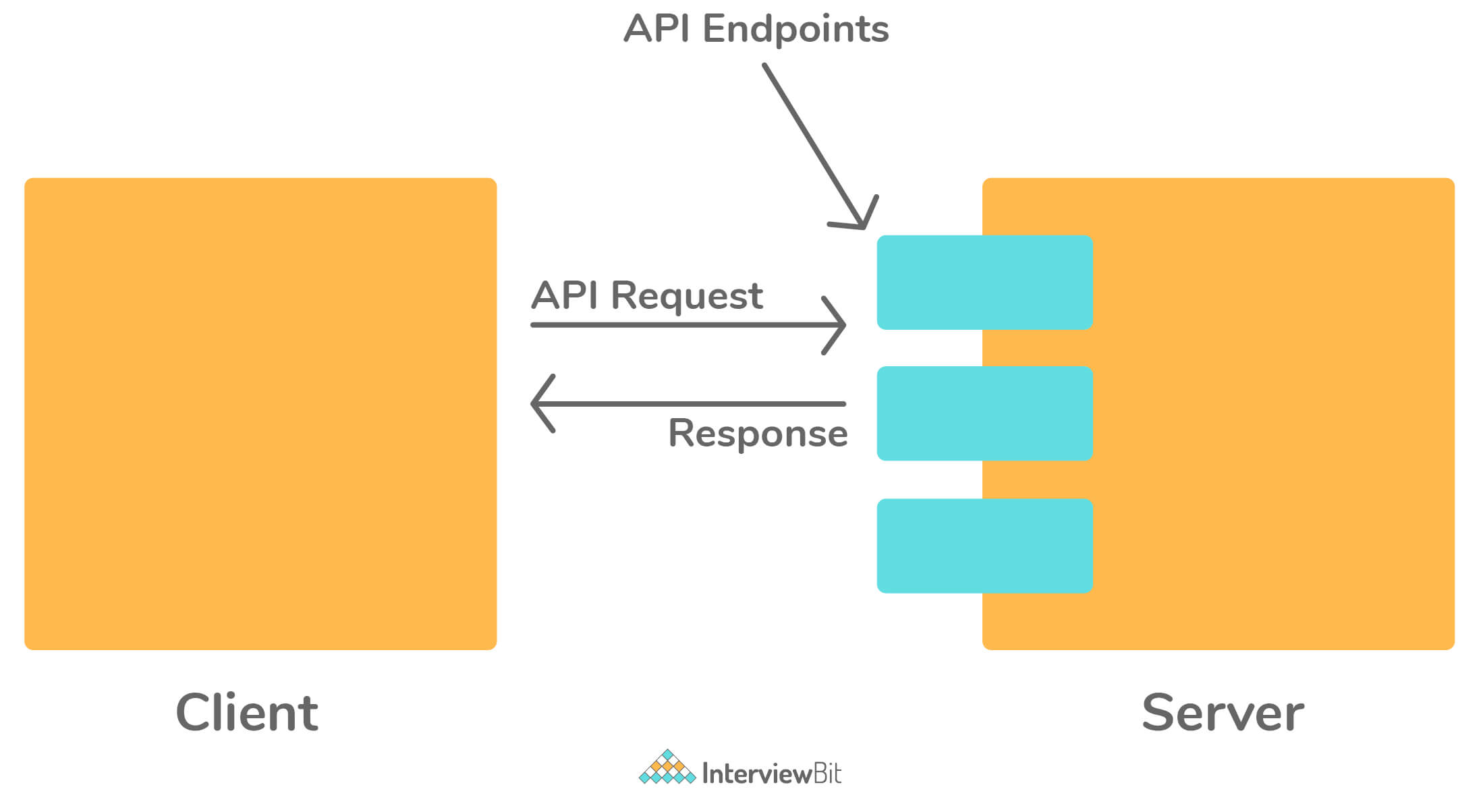
A unit test invokes the code and verifies the result with the expected result. If the expected and actual outcomes match, then the unit test passes. Otherwise, it fails.

A good unit test has the following characteristics:

1. It should test a single piece of functionality.
2. It is fully automated and repeatable.
3. It should run quickly and provide immediate feedback.
4. It should be isolated and shouldn’t interact with external dependencies such as network, database, or file system unless needed. You can use the mocking technique to simulate the external dependencies and isolate the code under test.

### 72. What is an API?

API stands for Application Programming Interface. It is a means of communication between two software components. An API abstracts the internal workings and complexity of a software program and allows the user of that API to solely focus on the inputs and outputs required to use it.

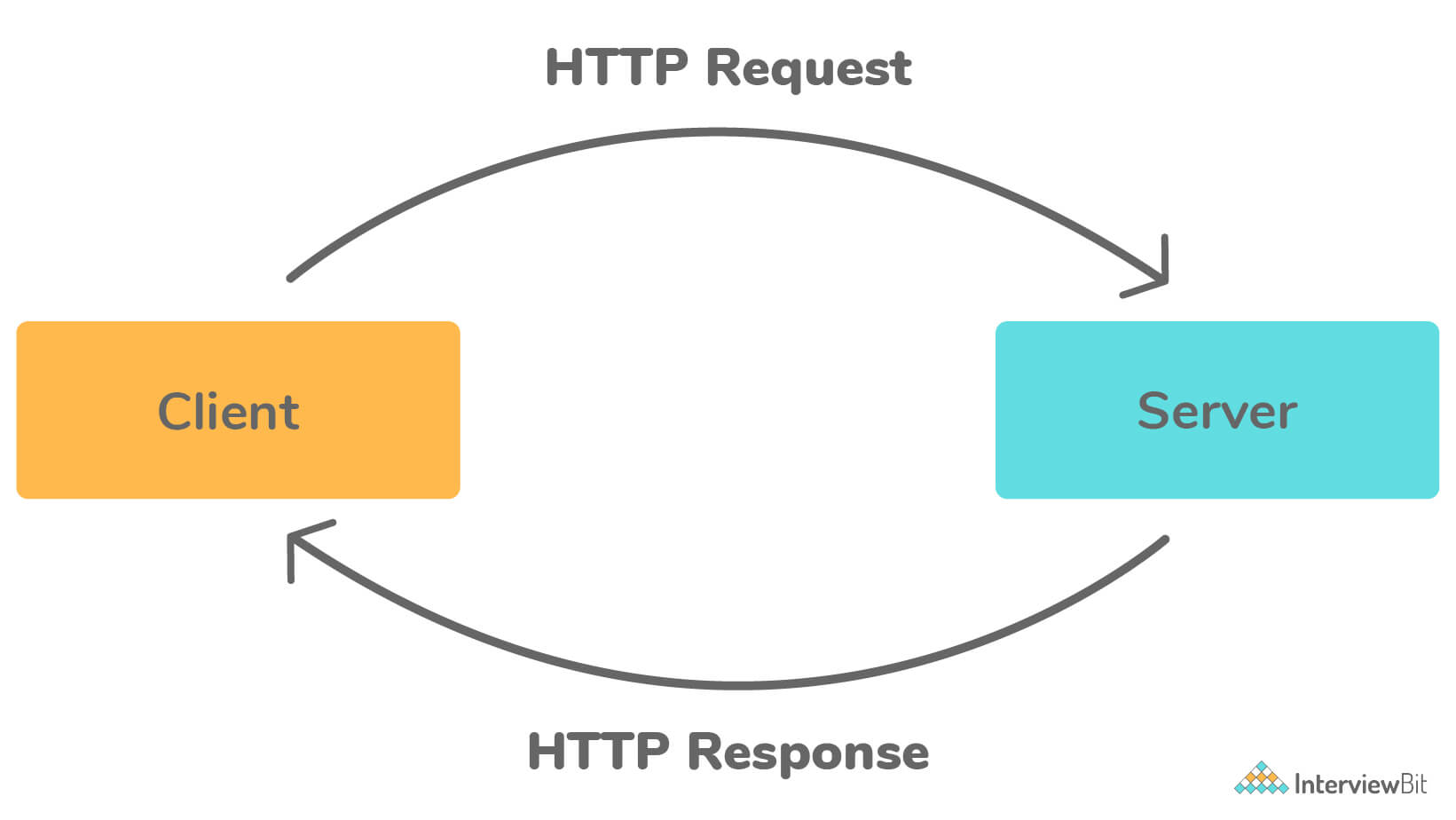


When building software, developers rarely write software from scratch and make use of other third-party libraries. An API allows two software components to talk to each other by providing an interface that they can understand.

Another use of an API is to provide data required by an application. Let's say you are building a weather application that displays the temperature. Instead of building the technology to collect the temperature yourself, you'd access the API provided by the meteorological institute.

### 73. What are the different HTTP status codes that a server can return?

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



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

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

### 74. What is test coverage?

Test coverage is a metric that indicates how much of the source code is covered by the tests, allowing the tester to verify the quality of their testing. It helps the tester figure out whether they are testing everything they're supposed to test.

Test coverage can mean different things to different people, depending on the particulars of their testing approaches.

1. Product: It means looking at test coverage to answer the question: Which features or the areas of the software does your tests cover?
2. Requirements: The software might work well, but it's not useful to the customer if it doesn't satisfy their needs. Requirements coverage indicates how many of the requirements are tested.
3. Source Code: This is usually a developer's domain and is a white-box testing technique. The developer can check how much of their source code is covered by the unit tests.

### 75. What is meant by browser automation?

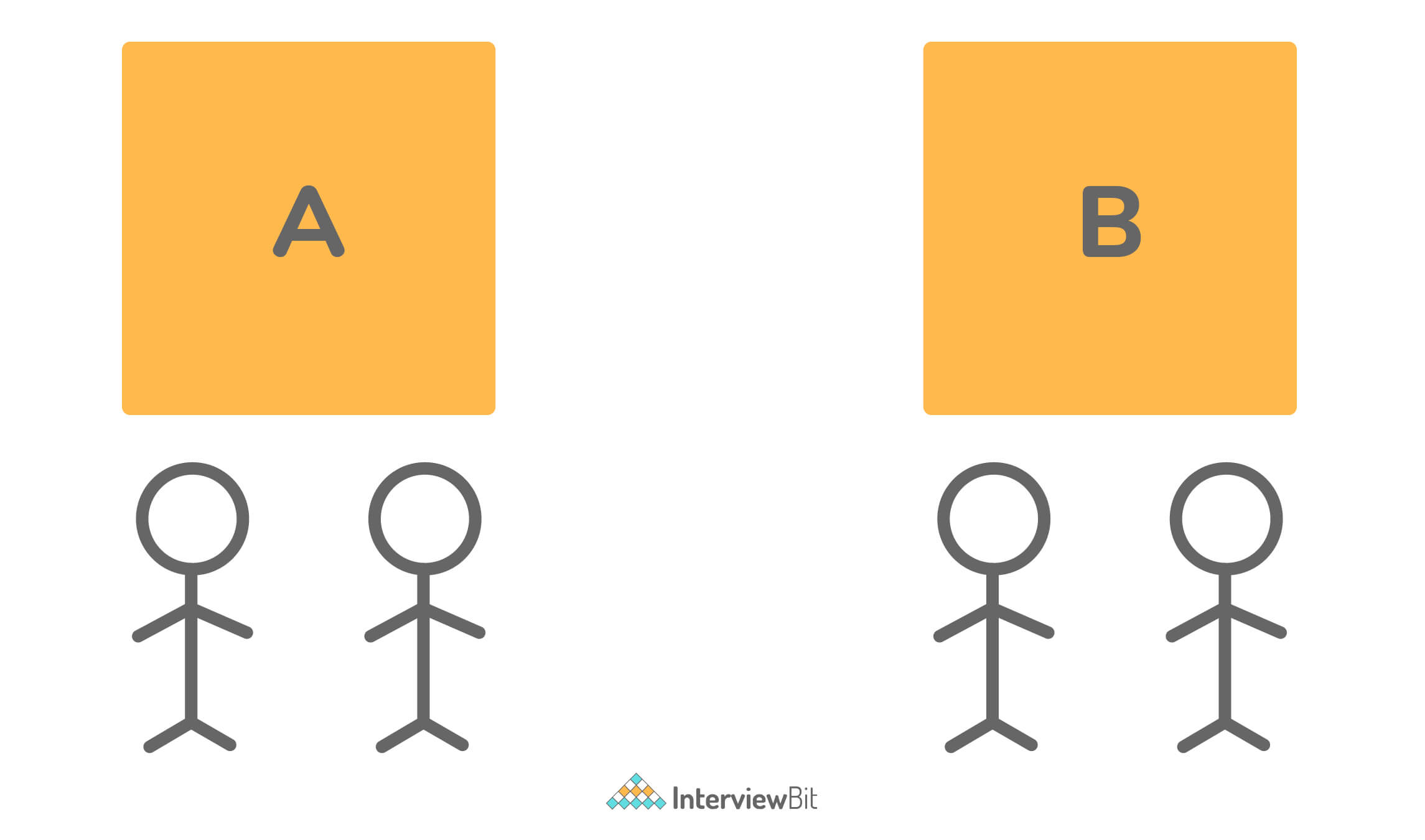
It’s a process of automatically testing a web application’s functionality in a browser, where a program launches the browser, navigates to the application, and interacts with the user interface by clicking buttons or links, just like an average user would.

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

### 76. What is A/B testing?

A/B testing is the process of testing two or more different versions of your software with users to assess which performs better. It is a low-risk way of testing variations of a new or existing functionality.

You can choose a part of your users to use feature A. The other group uses feature B. Then user feedback and response are evaluated using statistical testing to decide the final version of the feature.



Typically, A/B testing is used to test the user experience of different interfaces. This allows the team to quickly gather feedback and test their initial hypothesis.

### 77. What is the difference between Retesting and Regression Testing?

**Regression Testing**: Regression testing, also known as generic testing, revolves around re-running functional and non-functional tests. It is especially done to ensure whether previously developed and tested software still performs the same after a change or not. It can be performed either manually or using automated tests.  
  
**Re-testing:** Re-testing, also known as planned testing, is used for specific bugs after it has been fixed by the developers. Re-testing is performed to check the scenario under the same environmental conditions after detection has been fixed.   
  
**Regression vs Retesting:**

| **Regression** | **Retesting** |
| --- | --- |
| It is performed to make sure that the changes haven't affected the unchanged part. | It is performed to make sure that the test cases that were filed in the last execution are passed after the detects are fixed by developers. |
| It is not carried out for specific detect fixes. | It is usually carried out based on defect fixes. |
| It is only the previous version functionality-centric. | It is current or previous version functionality-centric. |
| It can be performed parallel with retesting. | It is needed to perform before regression testing. |
| It does not include the verification of bugs. | It includes the verification of bugs. |
| In this type of testing, test cases can be automated and the testing style is generic. | In this type of testing, test cases cannot be automated and the testing is done in a planned manner. |
| It is only used for passed test cases. | It is only used for failed test cases. |

### 78. What is System testing and Unit Testing? Write the difference between them.

**System Testing:** It is a typical black box testing technique that is performed in a complete and fully integrated system to evaluate the system’s compliance with its specific requirements. It must investigate both functional requirements and non-functional requirements. Generally, it is performed by both testers and developers.  
  
**Unit Testing:** In unit testing, each component of the software is individually tested. Generally, unit testing is performed by developers. Those systems that have a lot of interdependencies between their modules cannot be tested by unit testing.   
  
**System vs Unit Testing:**

| **System Testing** | **Unit Testing** |
| --- | --- |
| The system testing method involves treating each module as a separate target for testing, and integrating the modules after each has been tested. | The purpose of unit testing is to test only one module at a time, rather than the integrated version of the application. |
| Generally, when it comes to unit testing, a single module testing approach is taken. | For System test cases, it includes both top-down approach testing and bottom-up approach testing with all modules in integrated mode. |
| It focuses on system validation. | It focuses on functional verification. |
| It usually follows the requirements specification. | It usually follows the specification of modules. |
| It is also known as black-box testing. | It is also known as white-box testing. |
| It is a low-level test as compared to unit testing. | It is a high-level test as compared to system testing. |

### 79. What are the types of Integration Testing?

* Big bang testing
* Bottom-Up Testing
* Top-Down Testing

### 80. Name some of the most popular integration testing tools.

Some of the most popular integration testing tools include:

* DBUnit
* Greenmail
* Mockito
* REST-Assured
* JUnit 5
* H2 Database, etc.

### 81. What is Test Harness and Test Closure?

**Test Harness**: Test harness, also known as the automated test framework, is a collection of software and test data required to unit test software modules during development. It is mostly used by the developers and helps in the automation and execution of unit test cases. It generally includes two main parts as given below:

* Test execution engine
* Test script repository

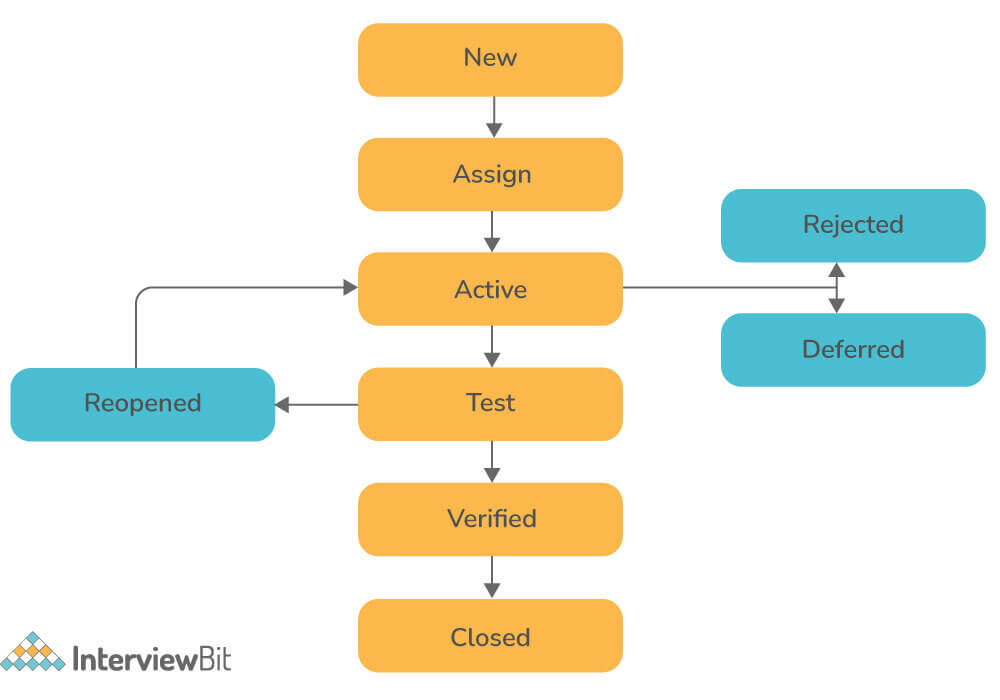
**Test Closure**: Test closure is basically a document that provides the summary of all the tests that are performed during SDLC. It gives full detailed analysis reports of the bugs that are discovered and removed. It is usually performed prior to the end of the testing process.

### 82. Explain Test Scenario.

Test scenario also called a scenario test, is defined as high-level detailed documentation of test cases or use cases. In this, the tester tests the software application from an end-user perspective. It usually can serve as the basis for lower-level test cases or use case creation. Test scenario is also known as test condition or test possibility. It gives you an idea of what we need to test.

### 83. What is the defect life cycle?

Defect life cycle, also known as a bug life cycle, is a life cycle of various stages through which a defect goes during its whole lifetime. This life cycle starts as soon as the defect is discovered or reported by the tester and ends when the tester ensures that the defect is resolved and it won't occur again. The defect life cycle includes the steps as shown below:



### 84. Explain Experienced-based testing techniques.

The experience-based testing technique is a type of testing that is based on the tester’s experience with testing to understand the essential areas of a system. This type of testing is generally used in a low-risk system. Individual’s information, abilities, and foundation knowledge are prime supporters of the test conditions and experiments in experienced-based techniques. There are four different experienced-based testing techniques as shown below:



### 85. Write the difference between smoke testing and sanity testing.

**Smoke Testing:** It is a type of testing performed to ensure that the acute functionalities of the program are working well. It acts as a confirmation of whether the quality assurance team can further proceed with testing or not.    
**Sanity Testing:** It is an unscripted form of testing performed to ensure that the code changes that are made are working well. It is performed by the test team for some basic tests. This testing focuses on one or a few areas of functionality and is usually narrow and deep.   
  
**Smoke vs Sanity Testing:**

| **Smoke Testing** | **Sanity Testing** |
| --- | --- |
| Its main objective is to measure the stability of the system so that the team can proceed with more rigorous testing. | Its main objective is to measure the rationality of the system so that the team can proceed with more rigorous testing. |
| It is usually performed by the developers or testers. | It is usually performed by testers. |
| It is a subset of regression testing. | It is a subset of acceptance testing. |
| It can be executed both manually or by using automation tools. | It can only be executed manually, not by using automation tools. |
| It is generally performed when a new product is developed. | It is generally performed after regression testing. |
| It is documented and is used to test the end-to-end functions of the application. | It is not documented and is used to test only modified or defect-fixed functions. |
| It is also considered as a subset of acceptance testing. | It is also considered a subset of regression testing. |

### 86. What do you mean by pesticide paradox?

Pesticide paradox is basically a phenomenon where the more one tests the software, the more it becomes immune to its tests. To overcome this, testers should always find new strategies, approaches, and test cases, so that they can identify bugs and resolve them.

### 87. Explain Configuration Testing.

Configuration testing is a software testing technique that is used to evaluate the configurational requirements of the software. It discovers the optimal configuration of the system under which the application performs at its best, therefore configuration testing is considered important. It also helps in identifying and resolving any compatibility issues.

### 88. Name two parameters that can be useful to check the quality of test execution.

Two parameters required to check the quality of test execution includes:

* **Defect reject ratio**: Ratio of total rejection to total production.
* **Defect leakage ratio**: Ratio of the total possibility of rejection occurrence to the total production.

## Conclusion

### 89. Conclusion

Software testing is an important activity that ensures quality, giving the confidence to release the software to customers. This article explained the testing process and its importance in software development. It also covers important concepts on manual testing and will guide you to master the field of manual testing.

However, testing is only a single component of a good software development strategy. A development team should use high coding standards, best practices, and patterns to reduce the bug count. As a long-term strategy, the best way to improve the testing process is to test frequently, measure the results, gather feedback and use it to get better.

# Guru99: 159Q

### 1. What is Exploratory Testing?

Exploratory testing is a hands-on approach in which testers are involved in minimum planning and maximum test execution. The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used. The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts. This does not mean that other, more formal testing techniques will not be used. For example, the tester may decide to use boundary value analysis but will think through and test the most important boundary values without necessarily writing them down. Some notes will be written during the exploratory-testing session so that a report can be produced afterward.

### 2. What is “use case testing”?

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

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

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

### 4. What is traceability matrix?

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

### 5. What is Equivalence partitioning testing?

Equivalence partitioning testing is a software testing technique which divides the application input test data into each partition at least once of equivalent data from which test cases can be derived. By this testing method, it reduces the time required for software testing.

### 6. What is white box testing and list the types of white box testing?

White box testing technique involves selection of test cases based on an analysis of the internal structure (Code coverage, branches coverage, paths coverage, condition coverage, etc.) of a component or system. It is also known as Code-Based testing or Structural testing. Different types of white box testing are

1. Statement Coverage
2. Decision Coverage

### 7. In white box testing, what do you verify?

In white box testing following steps are verified.

1. Verify the security holes in the code
2. Verify the incomplete or broken paths in the code
3. Verify the flow of structure according to the document specification
4. Verify the expected outputs
5. Verify all conditional loops in the code to check the complete functionality of the application
6. Verify the line by line coding and cover 100% testing

### 8. What is black box testing? What are the different black box testing techniques?

Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program. This testing is usually done to check the functionality of an application. The different black box testing techniques are

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

### 9. What is the difference between static and dynamic testing?

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

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

### 10. What are verification and validation?

Verification is a process of evaluating software at the development phase. It helps you to decide whether the product of a given application satisfies the specified requirements. Validation is the process of evaluating software at the after the development process and to check whether it meets the customer requirements.

### 11. What are the different test levels?

There are four test levels

1. Unit/component/program/module testing
2. Integration testing
3. System testing
4. Acceptance testing

### 12. What is Integration testing?

[Integration testing](https://www.guru99.com/integration-testing.html)is a level of software testing process, where individual units of an application are combined and tested. It is usually performed after unit and functional testing.

### 13. What Test Plans consists of?

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

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

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

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

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

### 15. Mention the difference between Data Driven Testing and Retesting?

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

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

## Advanced Manual Software Testing Interview Questions for 3/5/10 Years Experience

### 16. What are the valuable steps to resolve issues while testing?

* Record: Log and handle any problems which have happened
* Report: Report the issues to higher level manager
* Control: Define the issue management process

### 17. What is the difference between test scenarios, test cases, and test script?

Difference between test scenarios and test cases is that

**Test Scenarios:** A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.

**Test Cases:** It is a document that contains the steps that have to be executed; it has been planned earlier.

**Test Script:**It is written in a programming language and it’s a short program used to test part of the functionality of the software system. In other words a written set of steps that should be performed manually.

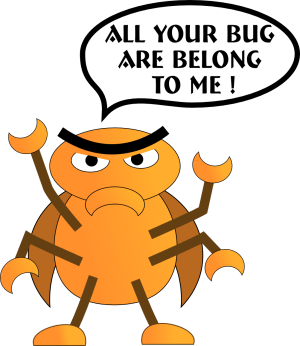
### 18. What is Latent defect?

**Latent defect:**This defect is an existing defect in the system which does not cause any failure as the exact set of conditions has never been met

### 19. What are the two parameters which can be useful to know the quality of test execution?

To know the quality of test execution, we can use two parameters

* Defect reject ratio
* Defect leakage ratio



Parameters for quality of test execution

### 20. What is the function of the software testing tool “phantom”?

Phantom is a freeware and is used for windows GUI automation scripting language. It allows us to take control of windows and functions automatically. It can simulate any combination of keystrokes and mouse clicks as well as menus, lists and more.

### 21. Explain what Test Deliverables is?

Test Deliverables are a set of documents, tools and other components that have to be developed and maintained in support of testing.

There are different test deliverables at every phase of the software development lifecycle

* Before Testing
* During Testing
* After the Testing

### 22. What is mutation testing?

Mutation testing is a technique to identify if a set of test data or test case is useful by intentionally introducing various code changes (bugs) and retesting with original test data/ cases to determine if the bugs are detected.

### 23. What all things you should consider before selecting automation tools for the AUT?

* Technical Feasibility
* Complexity level
* Application stability
* Test data
* Application size
* Re-usability of automated scripts
* Execution across environment

### 24. How will you conduct Risk Analysis?

For the risk analysis following steps need to be implemented

1. Finding the score of the risk
2. Making a profile for the risk
3. Changing the risk properties
4. Deploy the resources of that test risk
5. Making a database of risk

### 25. What are the categories of debugging?

Categories for debugging

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

### 26. What is fault masking explain with example?

When the presence of one defect hides the presence of another defect in the system, it is known as fault masking.

Example: If the “Negative Value” cause a firing of unhandled system exception, the developer will prevent the negative values input. This will resolve the issue and hide the defect of unhandled exception firing.

### 27. Explain what Test Plan is? What is the information that should be covered in Test Plan?

A test plan can be defined as a document describing the scope, approach, resources, and schedule of testing activities and a test plan should cover the following details.

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

### 28. How can you eliminate the product risk in your project?

It helps you to eliminate product risk in your project, and there is a simple yet crucial step that can reduce the product risk in your project.

* Investigate the specification documents
* Have discussions about the project with all stakeholders including the developer
* As a real user walk around the website

### 29. What is the common risk that leads to project failure?

The common risk that leads to a project failure are

* Not having enough human resource
* Testing Environment may not be set up properly
* Limited Budget
* Time Limitations

### 30. On what basis you can arrive at an estimation for your project?

To estimate your project, you have to consider the following points

* Divide the whole project into the smallest tasks
* Allocate each task to team members
* Estimate the effort required to complete each task
* Validate the estimation

### 31. Explain how you would allocate a task to team members?

|  |  |
| --- | --- |
| **Task** | **Member** |
| * Analyze software requirement specification | * All the members |
| * Create the test specification | * Tester/Test Analyst |
| * Build up the test environment | * Test administrator |
| * Execute the test cases | * Tester, a Test administrator |
| * Report defects | * Tester |

### 32. Explain what is testing type and what are the commonly used testing type?

To get an expected test outcome, a standard procedure is followed which is referred to as Testing Type.

Commonly used testing types are

* Unit Testing: Test the smallest code of an application
* API Testing: Testing API created for the application
* Integration Testing: Individual software modules are combined and tested
* System Testing: Complete testing of the system
* Install/UnInstall Testing: Testing done from the point of client/customer view
* Agile Testing: Testing through Agile technique

### 33. While monitoring your project what all things you have to consider?

The things that have to be taken in considerations are

* Is your project on schedule
* Are you over budget
* Are you working towards the same career goal
* Have you got enough resources
* Are there any warning signs of impending problems
* Is there any pressure from management to complete the project sooner

### 34. What are the common mistakes which create issues?

* Matching resources to wrong projects
* Test manager lack of skills
* Not listening to others
* Poor Scheduling
* Underestimating
* Ignoring the small problems
* Not following the process

### 35. What does a typical test report contain? What are the benefits of test reports?

A test report contains the following things:

* Project Information
* Test Objective
* Test Summary
* Defect

The benefits of test reports are:

* Current status of project and quality of product are informed
* If required, stakeholder and customer can take corrective action
* A final document helps to decide whether the product is ready for release

### 36. What is test management review and why it is important?

Management review is also referred to as [Software Quality Assurance](https://www.guru99.com/software-quality-assurance-test-audit-review-makes-your-life-easy.html) or SQA. SQA focusses more on the software process rather than the software work products. It is a set of activities designed to make sure that the project manager follows the standard process. SQA helps test manager to benchmark the project against the set standards.

### 37. What are the best practices for software quality assurance?

The best practices for an effective SQA implementation is

* Continuous Improvement
* Documentation
* Tool Usage
* Metrics
* Responsibility by team members
* Experienced SQA auditors

### 38. When is RTM (Requirement Traceability Matrix) prepared?

RTM is prepared before test case designing. Requirements should be traceable from review activities.

### 39. What is the difference between Test matrix and Traceability matrix?

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

**Traceability Matrix**: Mapping between test cases and customer requirements is known as Traceability Matrix

### 40. In manual testing what are stubs and drivers?

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

### 41. What is the step you would follow once you find the defect?

Once a defect is found you would follow the step

a) Recreate the defect

b) Attach the screenshot

c) Log the defect

### 42. Explain what is “Test Plan Driven” or “Key Word Driven” method of testing?

This technique uses the actual test case document developed by testers using a spreadsheet containing special “key Words”. The key words control the processing.

### 43. What is the DFD (Data Flow Diagram)?

When a “flow of data” through an information system is graphically represented, then it is known as Data Flow Diagram. It is also used for the visualization of data processing.

### 44. Explain what LCSAJ is?

LCSAJ stands for ‘linear code sequence and jump.’ It consists of the following three items

a) Start of the linear sequence of executable statements

b) End of the linear sequence

c) The target line to which control flow is transferred at the end of the linear sequence

### 45. Explain what N+1 testing is?

The variation of regression testing is represented as N+1. In this technique, the testing is performed in multiple cycles in which errors found in test cycle ‘N’ are resolved and re-tested in test cycle N+1. The cycle is repeated unless there are no errors found.

### 46. What is Fuzz testing and when it is used?

Fuzz testing is used to detect security loopholes and coding errors in software. In this technique, random data is added to the system in an attempt to crash the system. If vulnerability persists, a tool called fuzz tester is used to determine potential causes. This technique is more useful for bigger projects but only detects a major fault.

### 47. Mention what the main advantages of statement coverage metric of software testing are?

The benefit of statement coverage metric is that

a) It does not require processing source code and can be applied directly to object code

b) Bugs are distributed evenly through the code, due to which percentage of executable statements covered reflects the percentage of faults discovered

### 48. How to generate test cases for “replace a string” method?

a) If characters in new string > characters in the previous string. None of the characters should get truncated

b) If characters in new string< characters in the previous string. Junk characters should not be added

c) Spaces after and before the string should not be deleted

d) String should be replaced only for the first occurrence of the string

### 49. How will you handle a conflict amongst your team members?

* I will talk individually to each person and note their concerns
* I will find a solution to the common problems raised by team members
* I will hold a team meeting, reveal the solution and ask people to co-operate

### 50. Mention what are the categories of defects?

Mainly there are three defect categories

* **Wrong**: When a requirement is implemented incorrectly
* **Missing**: It is a variance from the specification, an indication that a specification was not implemented or a requirement of the customer is not met
* **Extra**: A requirement incorporated into the product that was not given by the end customer. It is considered as a defect because it is a variance from the existing requirements

### 51. Explain how does a test coverage tool work?

The code coverage testing tool runs parallel while performing testing on the actual product. The code coverage tool monitors the executed statements of the source code. When the final testing is done, we get a complete report of the pending statements and also get the coverage percentage.

### 52. Mention what the difference between a “defect” and a “failure” in software testing is?

In simple terms when a defect reaches the end customer, it is called a failure while the defect is identified internally and resolved; then it is referred to as a defect.

### 53. Explain how to test documents in a project that span across the software development lifecycle?

The project span across the software development lifecycle in the following manner

* Central/Project test plan: It is the main test plan that outlines the complete test strategy of the project. This plan is used till the end of the software development lifecycle
* Acceptance test plan: This document begins during the [requirement phase](https://www.guru99.com/lifecycle-of-requirement.html) and is completed at the final delivery
* System test plan: This plan starts during the design plan and proceeds until the end of the project
* Integration and Unit test plan: Both these test plans start during the execution phase and last until the final delivery

### 54. Explain which test cases are written first black boxes or white boxes?

Black box test cases are written first as to write black box test cases; it requires project plan and requirement document all these documents are easily available at the beginning of the project. While writing white box test cases requires more architectural understanding and is not available at the start of the project.

### 55. Explain what the difference between latent and masked defects is?

* **Latent defect:** A latent defect is an existing defect that has not caused a failure because the sets of conditions were never met
* **Masked defect:** It is an existing defect that has not caused a failure because another defect has prevented that part of the code from being executed

### 56. Mention what bottom-up testing is?

Bottom-up testing is an approach to integration testing, where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

### 57. Mention what the different types of test coverage techniques are?

Different types of test coverage techniques include

* **Statement Coverage:** It verifies that each line of source code has been executed and tested
* **Decision Coverage:** It ensures that every decision in the source code is executed and tested
* **Path Coverage:** It ensures that every possible route through a given part of the code is executed and tested

### 58. Mention what the meaning of breath testing is?

Breath testing is a test suite that exercises the full functionality of a product but does not test features in detail

### 59. Explain what the meaning of Code Walk Through is?

Code Walk Through is the informal analysis of the program source code to find defects and verify coding techniques

### 60. Mention what the basic components of defect report format are?

The essential components of defect report format include

* Project Name
* Module Name
* Defect detected on
* Defect detected by
* Defect ID and Name
* Snapshot of the defect
* Priority and Severity status
* Defect resolved by
* Defect resolved on

### 61. Mention what the purpose behind doing end-to-end testing is?

End-to-end testing is done after functional testing. The purpose behind doing end-to-end testing is that

* To validate the [software requirements](https://www.guru99.com/learn-software-requirements-analysis-with-case-study.html) and integration with external interfaces
* Testing application in real-world environment scenario
* Testing of interaction between application and database

### 62. Explain what it means by test harness?

A test harness is configuring a set of tools and test data to test an application in various conditions, and it involves monitoring the output with expected output for correctness.

### 63. Explain in a testing project what testing activities would you automate?

In testing project testing activities, you would automate are

* Tests that need to be run for every build of the application
* Tests that use multiple data for the same set of actions
* Identical tests that need to be executed using different browsers
* Mission critical pages
* A transaction with pages that do not change in a short time

### 64. What is the MAIN benefit of designing tests early in the life cycle?

It helps prevent defects from being introduced into the code.

### 65. What is risk-based testing?

Risk-based[Testing](https://www.guru99.com/software-testing.html)is the term used for an approach to creating a[Test Strategy](https://www.guru99.com/how-to-create-test-strategy-document.html)that is based on prioritizing tests by risk. The basis of the approach is a detailed risk analysis and prioritizing of risks by risk level. Tests to address each risk are then specified, starting with the highest risk first.

### 66. What is the KEY difference between preventative and reactive approaches to testing?

Preventative tests are designed early; reactive tests are designed after the software has been produced.

### 67. What is the purpose of exit criteria?

The purpose of exit criteria is to define when a test level is completed.

### 68. What determines the level of risk?

The likelihood of an adverse event and the impact of the event determine the level of risk.

### 69. When is used Decision table testing?

Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table, the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.

Learn More About Decision Table Testing Technique in the Video Tutorial [**here**](https://www.guru99.com/decision-table-testing.html)

### 70. Why we use decision tables?

The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs. However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface. The other two specification-based techniques, decision tables, and state transition testing are more focused on business logic or business rules. A decision table is a good way to deal with combinations of things (e.g., inputs). This technique is sometimes also referred to as a ’cause-effect’ table. The reason for this is that there is an associated logic diagramming technique called ’cause-effect graphing’ which was sometimes used to help derive the decision table

### 71. What is the MAIN objective when reviewing a software deliverable?

To identify defects in any software work product.

### 72. Which of the following defines the expected results of a test? Test case specification or test design specification.

Test case specification defines the expected results of a test.

### 73. What is the benefit of test independence?

It avoids author bias in defining effective tests.

### 74. As part of which test process do you determine the exit criteria?

The exit criteria are determined on the bases of ‘Test Planning’.

### 75. What is Alpha testing?

Pre-release testing by end user representatives at the developer’s site.

### 76. What is beta testing?

Testing performed by potential customers at their own locations.

### 77. Mention what the difference between Pilot and Beta testing is?

The difference between a pilot and beta testing is that pilot testing is actually done using the product by the group of users before the final deployment, and in beta testing, we do not input real data, but it is installed at the end customer to validate if the product can be used in production.

### 78. Given the following fragment of code, how many tests are required for 100% decision coverage?

if width > length

thenbiggest\_dimension = width

if height > width

thenbiggest\_dimension = height

end\_if

elsebiggest\_dimension = length

if height > length

thenbiggest\_dimension = height

end\_if

end\_if

4

### 79. You have designed test cases to provide 100% statement and 100% decision coverage for the following fragment of code. if width > length then biggest\_dimension = width else biggest\_dimension = length end\_if The following has been added to the bottom of the code fragment above. print “Biggest dimension is ” &biggest\_dimensionprint “Width: ” & width print “Length: ” & length How many more test cases are required?

None, existing test cases can be used.

### 80. What is the difference between Testing Techniques and Testing Tools?

Testing technique: – Is a process for ensuring that some aspects of the application system or unit functions properly there may be few techniques but many tools.

Testing Tools: – Is a vehicle for performing a test process. The tool is a resource to the tester, but itself is insufficient to conduct testing

Learn More About Testing Tools [**here**](https://www.guru99.com/testing-tools.html)

### 81. We use the output of the requirement analysis, the requirement specification as the input for writing …

User Acceptance Test Cases

### 82. Repeated Testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software component:

Regression Testing

### 83. A wholesaler sells printer cartridges. The minimum order quantity is 5. There is a 20% discount for orders of 100 or more printer cartridges. You have been asked to prepare test cases using various values for the number of printer cartridges ordered. Which of the following groups contain three test inputs that would be generated using Boundary Value Analysis?

4, 5, 99

### 84. What is component testing?

Component testing, also known as unit, module, and program testing, searches for defects in and verifies the functioning of software (e.g., modules, programs, objects, classes, etc.) that are separately testable. Component testing may be done in isolation from the rest of the system depending on the context of the development life cycle and the system. Most often stubs and drivers are used to replace the missing software and simulate the interface between the software components simply. A stub is called from the software component to be tested; a driver calls a component to be tested.

Here is an awesome video on [**Unit Testing**](https://www.guru99.com/unit-testing-guide.html)

### 85. What is functional system testing?

Testing the end to end functionality of the system as a whole is defined as a functional system testing.

### 86. What are the benefits of Independent Testing?

Independent testers are unbiased and identify different defects at the same time.

### 87. In a REACTIVE approach to testing when would you expect the bulk of the test design work to be begun?

The bulk of the test design work begun after the software or system has been produced.

### 88. What are the different Methodologies in Agile Development Model?

There are currently seven different agile methodologies that I am aware of:

1. Extreme Programming (XP)
2. Scrum
3. Lean Software Development
4. Feature-Driven Development
5. Agile Unified Process
6. Crystal
7. Dynamic Systems Development Model (DSDM)

### 89. Which activity in the fundamental test process includes evaluation of the testability of the requirements and system?

A ‘Test Analysis’ and ‘Design’ includes evaluation of the testability of the requirements and system.

### 90. What is typically the MOST important reason to use risk to drive testing efforts?

Because testing everything is not feasible.

### 91. What is random/monkey testing? When is it used?

Random testing is often known as monkey testing. In such type of testing data is generated randomly often using a tool or automated mechanism. With this randomly generated input, the system is tested, and results are analyzed accordingly. These testing are less reliable; hence it is normally used by the beginners and to see whether the system will hold up under adverse effects.

### 92. Which of the following are valid objectives for incident reports?

1. Provide developers and other parties with feedback about the problem to enable identification, isolation, and correction as necessary.
2. Provide ideas for test process improvement.
3. Provide a vehicle for assessing tester competence.
4. Provide testers with a means of tracking the quality of the system under test.

### 93. Consider the following techniques. Which are static and which are dynamic techniques?

1. Equivalence Partitioning.
2. Use Case Testing.
3. Data Flow Analysis.
4. Exploratory Testing.
5. Decision Testing.
6. Inspections.

Data Flow Analysis and Inspections are static; Equivalence Partitioning, Use Case Testing, Exploratory Testing and Decision Testing are dynamic.

### 94. Why are static testing and dynamic testing described as complementary?

Because they share the aim of identifying defects but differ in the types of defect they find.

### 95. What are the phases of a formal review?

In contrast to informal reviews, formal reviews follow a formal process. A typical formal review process consists of six main steps:

1. Planning
2. Kick-off
3. Preparation
4. Review meeting
5. Rework
6. Follow-up.

### 96. What is the role of moderator in the review process?

The moderator (or review leader) leads the review process. He or she determines, in co-operation with the author, the type of review, approach and the composition of the review team. The moderator performs the entry check and the follow-up on the rework, in order to control the quality of the input and output of the review process. The moderator also schedules the meeting, disseminates documents before the meeting, coaches other team members, paces the meeting, leads possible discussions and stores the data that is collected.

Learn More about Review process in Video Tutorial [**here**](https://www.guru99.com/testing-review.html)

### 97. What is an equivalence partition (also known as an equivalence class)?

An input or output ranges of values such that only one value in the range becomes a test case.

### 98. When should configuration management procedures be implemented?

During test planning.

### 99. A Type of Functional Testing, which investigates the functions relating to the detection of threats, such as virus from malicious outsiders?

Security Testing

### 100. Testing wherein we subject the target of the test, to varying workloads to measure and evaluate the performance behaviors and the ability of the target and the test to continue to function properly under these different workloads?

Load Testing

### 101. Testing activity which is performed to expose defects in the interfaces and in the interaction between integrated components is?

Integration Level Testing

### 102. What are the Structure-based (white-box) testing techniques?

Structure-based testing techniques (which are also dynamic rather than static) use the internal structure of the software to derive test cases. They are commonly called ‘white-box’ or ‘glass-box’ techniques (implying you can see into the system) since they require knowledge of how the software is implemented, that is, how it works. For example, a structural technique may be concerned with exercising loops in the software. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.

### 103. When should “Regression Testing” be performed?

After the software has changed or when the environment has changed [Regression testing](https://www.guru99.com/regression-testing.html) should be performed.

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

A negative test is when you put in an invalid input and receives errors. While positive testing is when you put in a valid input and expect some action to be completed in accordance with the specification.

### 105. What is the purpose of a test completion criterion?

The purpose of test completion criterion is to determine when to stop testing

### 106. What can static analysis NOT find?

For example memory leaks.

### 107. What is the difference between re-testing and regression testing?

Re-testing ensures the original fault has been removed; regression testing looks for unexpected side effects.

### 108. What are the Experience-based testing techniques?

In experience-based techniques, people’s knowledge, skills, and background are a prime contributor to the test conditions and test cases. The experience of both technical and business people is important, as they bring different perspectives to the test analysis and design process. Due to previous experience with similar systems, they may have insights into what could go wrong, which is very useful for testing.

### 109. What type of review requires formal entry and exit criteria, including metrics?

Inspection

### 110. Could reviews or inspections be considered part of testing?

Yes, because both help detects faults and improves quality.

### 111. An input field takes the year of birth between 1900 and 2004 what the boundary values for testing this field are?

1899,1900,2004,2005

### 112. Which of the following tools would be involved in the automation of regression test? a. Data tester b. Boundary tester c. Capture/Playback d. Output comparator.

d. Output comparator

### 113. To test a function, what has to write a programmer, which calls the function to be tested and pass test data.

Driver

### 114. What is the one Key reason why developers have difficulty testing their own work?

Lack of Objectivity

### 115. “How much testing is enough?”

The answer depends on the risk for your industry, contract and special requirements.

### 116. When should testing be stopped?

It depends on the risks for the system being tested. 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

### 117. Which of the following is the primary purpose of the integration strategy for integration testing in the small?

The primary purpose of the integration strategy is to specify which modules to combine when and how many at once.

### 118. What are semi-random test cases?

Semi-random test cases are nothing, but when we perform random test cases and do equivalence partitioning to those test cases, it removes redundant test cases, thus giving us semi-random test cases.

### 119. Given the following code, which statement is true about the minimum number of test cases required for full statement and branch coverage?

**Read p**

**Read q**

**IF p+q> 100**

**THEN Print “Large”**

**ENDIF**

**IF p > 50**

**THEN Print “p Large”**

**ENDIF**

1 test for statement coverage, 2 for branch coverage

### 120. Which review is normally used to evaluate a product to determine its suitability for the intended use and to identify discrepancies?

Technical Review.

### 121. Faults found should be originally documented by whom?

By testers.

### 122. Which is the current formal world-wide recognized documentation standard?

There isn’t one.

### 123. Which of the following is the review participant who has created the item to be reviewed?

Author

### 124. A number of critical bugs are fixed in software. All the bugs are in one module, related to reports. The test manager decides to do regression testing only on the reports module.

Regression testing should be done on other modules as well because fixing one module may affect other modules.

### 125. Why does the boundary value analysis provide good test cases?

Because errors are frequently made during programming of the different cases near the ‘edges’ of the range of values.

### 126. What makes an inspection different from other review types?

It is led by a trained leader, uses formal entry and exit criteria and checklists.

### 127. Why can be tester dependent on configuration management?

Because configuration management assures that we know the exact version of the testware and the test object.

### 128. What is V-Model?

A software development model that illustrates how testing activities integrate with software development phases

### 129. What is maintenance testing?

Triggered by modifications, migration or retirement of existing software

### 130. What is test coverage?

Test coverage measures in some specific way the amount of testing performed by a set of tests (derived in some other way, e.g., using specification-based techniques). Wherever we can count things and can tell whether or not each of those things has been tested by some test, then we can measure coverage.

### 131. Why is incremental integration preferred over “big bang” integration?

Because incremental integration has better early defects screening and isolation ability

### 132. What is called the process starting with the terminal modules?

Bottom-up integration

### 133. During which test activity could fault be found most cost-effectively?

During test planning

### 134. The purpose of the requirement phase is

To freeze requirements, to understand user needs, to define the scope of testing

### 135. Why we split testing into distinct stages?

We split testing into distinct stages because of the following reasons,

1. Each test stage has a different purpose
2. It is easier to manage to test in stages
3. We can run different test into different environments
4. Performance and quality of the testing is improved using phased testing

### 136. What is DRE?

In order to measure test effectiveness, a powerful metric is used to measure test effectiveness known as DRE (Defect Removal Efficiency) From this metric we would know how many bugs we have found from the set of test cases. The formula for calculating DRE is

DRE=Number of bugs while a testing/number of bugs while testing + number of bugs found by a user

### 137. Which of the following is likely to benefit most from the use of test tools providing test capture and replay facilities? a) Regression testing b) Integration testing c) System testing d) User acceptance testing

Regression testing

### 138. How would you estimate the amount of re-testing likely to be required?

Metrics from previous similar projects and discussions with the development team

### 139. What studies data flow analysis?

The use of data on paths through the code.

### 140. What is failure?

Failure is a departure from specified behavior.

### 141. What are Test comparators?

Is it really a test if you put some inputs into some software, but never look to see whether the software produces the correct result? The essence of testing is to check whether the software produces the correct result and to do that, and we must compare what the software produces to what it should produce. A test comparator helps to automate aspects of that comparison.

### 142. Who is responsible for document all the issues, problems and open point that were identified during the review meeting

Scribe

### 143. What is the main purpose of Informal review

An inexpensive way to get some benefit

### 144. What is the purpose of test design technique?

Identifying test conditions and Identifying test cases

### 145. When testing a grade calculation system, a tester determines that all scores from 90 to 100 will yield a grade of A, but scores below 90 will not. This analysis is known as:

Equivalence partitioning

### 146. A test manager wants to use the resources available for the automated testing of a web application. The best choice is

Tester, test automater, web specialist, DBA

### 147. During the testing of a module tester, ‘X’ found a bug and assigned it to a developer. But developer rejects the same, saying that it’s not a bug. What ‘X’ should do?

Send the detailed information of the bug encountered and check the reproducibility

### 148. A type of integration testing in which software elements, hardware elements, or both are combined all at once into a component or an overall system, rather than in stages.

Big-Bang Testing

### 149. In practice, which Life Cycle model may have more, fewer or different levels of development and testing, depending on the project and the software product. For example, there may be component integration testing after component testing, and system integration testing after system testing.

V-Model

### 150. Which technique can be used to achieve input and output coverage? It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.

Equivalence partitioning

### 151. “This life cycle model is driven by schedule and budget risks” This statement is best suited for.

V-Model

### 152. In which order should tests be run?

The most important one must be tested first

### 153. The later in the development life cycle a fault is discovered, the more expensive it is to fix. Why?

The fault has been built into more documentation, code, tests, etc

### 154. What is Coverage measurement?

It is a partial measure of test thoroughness.

### 155. What is Boundary value testing?

Test boundary conditions on, below and above the edges of input and output equivalence classes. For instance, let say a bank application where you can withdraw maximum Rs.20,000 and a minimum of Rs.100, so in boundary value testing we test only the exact boundaries, rather than hitting in the middle. That means we test above the maximum limit and below the minimum limit.

### 156. What does COTS represent?

Commercial Off The Shelf.

### 157. The purpose of which is to allow specific tests to be carried out on a system or network that resembles as closely as possible the environment where the item under test will be used upon release?

Test Environment

### 158. What can be thought of as being based on the project plan, but with greater amounts of detail?

Phase Test Plan

### 159. What is Rapid Application Development?

Rapid Application Development (RAD) is formally a parallel development of functions and subsequent integration. Components/functions are developed in parallel as if they were mini projects, the developments are time-boxed, delivered, and then assembled into a working prototype. This can very quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Rapid change and development of the product are possible using this methodology. However the product specification will need to be developed for the product at some point, and the project will need to be placed under more formal controls before going into production.

# STH: 9Q

**#Q 1) What is the Process for Creating a Test Script?**

**Answer:**

**Step 1:** is to get a thorough understanding of the AUT:

* This could be by reading the requirement documents thoroughly.
* In the absence of docs, we could try to understand any point of reference that we have – a previous version of the application or wire-frames or screenshots

**Step 2:** After understanding the requirements, we make a list of what are the areas in this application that will have to be tested. In other words, we identify the test requirements. The focus in this step is to identify “What” to test. The outcome of this step is a list of [Test Scenarios](https://www.softwaretestinghelp.com/decision-table-test-case-design-technique/).

**Step 3:** Once we have the test scenarios, we concentrate next on “How” to test them.  This phase involves writing detailed steps about how to test a particular feature, what data to enter ([Test Data](https://www.softwaretestinghelp.com/tips-to-design-test-data-before-executing-your-test-cases/)) and what is the expected result.

Once these 3 steps are done, we are ready for testing.

**#Q 2) What are the fields in a Bug Report?**

**Answer:**Following important fields should be included in a [good Bug Report](https://www.softwaretestinghelp.com/sample-bug-report/)**:**

1. A unique ID
2. Defect Description: a short describing what the bug is.
3. Steps to Reproduce: details about how to arrive at the error, exact test data, the time at which defect was found(if applicable) environment: any information that will help re-encounter the issue
4. Module/section of the application (if applicable)
5. Severity
6. Screenshot
7. Responsible QA: in case of any follow-up questions regarding this issue

**#Q 3) How to test a customer-facing software?**

**Answer:**With any application that we test, we are trying to see if a certain set of requirements are met by the application or not. But when it comes to a user-facing site, apart from concentrating on functionality, we also have to look into a few usability features, maybe performance and security aspects also to a certain extent.

**The first level of testing is**: Does the site satisfy its functional requirements.

**For Example,** if it is a loan management site, we need to look at – are the new customer able to apply for a loan, are the existing customer able to access their loan info, is the interest % applied to the loan amount correct, etc.

**The next level of testing is**: how easy is it to use the site, do the options make a logical sense and meet the expectations of the user or not.

**For Example,** if the user has to be pass 3-4 screens to submit the basic information they are going to be annoyed, so such issues have to be addressed.

**Another** **example,** after entering username and password, the user might click on the tab- which means the control should go to “Sign in” button, instead if it’s going to cancel, the user is going to be really annoyed and the experience of using the site is going to be compromised. Such issues have to be caught.

[**Performance Testing**](https://www.softwaretestinghelp.com/introduction-to-performance-testing-loadrunner-training-tutorial-part-1/) to the complete extent might not be in scope but simple situations like, how long does the search results take to be displayed and how much time does it take for the system to retrieve a customer info at the peak hour – these are some example of the kind of things we would want to keep an eye on.

**Security** – for sites where there is a secure login to access the site, the minimum functionality around it has to be tested. **For Example,** if I leave the site idle for more than 10 minutes, is it auto logging out or not. Something as basic as that should be focused on.

**#Q 4) How to overcome the challenge of not having input documentation for testing?**

**Answer:**IF the detailed standard documentation like BRD and FSD are unavailable, the tester will have to depend on some point of reference.

* Screenshots
* A previous version of the application
* Wireframes, etc

Another factor that helps immensely, is to talk to the developers or the business analysts (when available) to get a confirmation on our understanding or clarifications in case of doubts.

When none of these situations works, we can just conceptualize the application based on our previous IT application experience and create the basic set of test scripts. When the testing phase comes up, we can set up a portion of test cycle time and do some test case management (make the already created scripts perfect) so we have the doc for the next phases.

**#Q 5) How to get**[**maximum productivity**](https://www.softwaretestinghelp.com/how-to-improve-your-testing-skills-and-beat-the-competition/)**from an offshore team?**

**Answer:**The key is to make sure that all the testers know about all the modules and that there is no knowledge concentration in one place. Involving everyone in test script peer reviews, defect meetings, and KT sessions are going to ensure that everyone is aware of the application to the best extent possible.

Also, by encouraging the concept of teamwork we can have the team members collaborate, help and aid each other for better productivity.

Regular follow up meetings also help the process very much.

**#Q 6) What are the Roles and Responsibilities of an onsite coordinator? Does he/she test too?**

**Answer:**The onsite coordinator is a point of contact for the offshore team and to the client for any information regarding the testing engagement.

**This job includes:**

* KT from and to offshore and clients
* Getting the environment to test all ready
* Sanity testing, smoke testing
* Testing – the key functionality.
* Bug review – found by the offshore team
* Bug assigning to the respective dev
* Presenting metrics
* Providing sign off

Yes, even an onsite coordinator has to test.

**#Q 7) Inconsistent bugs- Why onsite can find it, but offshore can’t and vice versa – How to handle this situation?**

**Answer:**Every bug has to be noted and analyzed – whether it is encountered at onsite or offshore, whether repeatable or not. A real value-add to a tester’s job is when we involve ourselves in the Root Cause Analysis process for a bug rather than simply reporting it.

**Some of the ways we can handle this situation are:**

* All the onsite and offshore team members should follow a guideline that screenshots had to be taken for every error that we encounter – repeatable or not.
* If there are logs, system files or anything like that, that might help us find any evidence of the issue- we should try to find it.
* Despite all these steps, if we still can’t tell why and when the problem occurs- we should report it to the developer all the same – with as much information as we can.

**#Q 8) Video/audio related testing – What does this include?**

**Answer:**How to test an application having video or audio?

**Here are the important points to consider:**

* Access levels (restricted or not – password controlled)
* Different kinds of environments
* Browser compatibility
* Screen resolutions
* Internet connection speeds
* The specific options on a video – like play, stop, mute, etc.
* Video by size
* Response to the videos – comments (limitations on the comment length and number of comments it can take)
* Video responses to the videos
* Interface with social networking sites – Interoperability
* Buffering speed
* Embedding the video

**#Q 9) Mobile Application Testing – What does it include briefly?**

**Answer: Mobile App Testing Important Test Scenarios:**

* Check if the app works well with multiple carriers and multiple devices.
* Usability of the features on a mobile screen.
* Testing it on different mobile platforms – like Android and iOS.
* Installations, uninstalling, launching the app with network and without a network, testing functionality.
* Network connections –WiFi, 2G, etc.
* Logs at iOS iPhone configuration utility for Android Monitor.bat can be used for debugging.

# SimpliLearn: 20Q

### 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 for 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

#### Free Course: Getting Started with JUnit

Learn the Basics of JUnit[ENROLL NOW](https://www.simplilearn.com/learn-junit-basics-free-course-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-skillup-course-promotion)

Free Course: Getting Started with JUnit

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

Manual testing is broken down into:

* Black Box
* White Box
* Integration
* Unit
* System
* Acceptance

### 9. 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

### 10. 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.

#### Post Graduate Program: Full Stack Web Development

in Collaboration with Caltech CTME[ENROLL NOW](https://www.simplilearn.com/pgp-full-stack-web-development-certification-training-course?source=GhPreviewCTABanner)

Post Graduate Program: Full Stack Web Development

### 11. 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.

### 12. 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.

### 13. 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.

### 14. 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.

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

This should be done during test planning.

### 16. List the four different test levels

The four levels are:

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

#### Free Course: Intro to RPA

Get closer to your dream role with the FREE course[START LEARNING](https://www.simplilearn.com/learn-rpa-basics-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-skillup-course-promotion)

Free Course: Intro to RPA

### 17. 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.

### 18. 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.

### 19. What’s GUI testing?

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

### 20. 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

# ArtOfTesting: 112Q

**Ques.1. What do you mean by Software Testing?**  
Ans. [Software testing](https://artoftesting.com/what-is-software-testing) is the process of evaluating a system to check if it satisfies its business requirements. It measures the overall quality of the system in terms of attributes. Like – correctness, completeness, usability, performance, etc.  
  
Basically, it is used for ensuring the quality of software to the stakeholders of the application.

**Ques.2. Why is testing required?**  
Ans. We need software testing for the following reasons-  
1. Testing provides an assurance to the stakeholders that the product works as intended.  
  
2. Avoidable defects leaked to the end-user/customer without proper testing adds a bad reputation to the development company.  
  
3. Defects detected earlier phase of SDLC results in lesser cost and resource utilization of correction.  
  
4. Saves development time by detecting issues in an earlier phase of development.  
  
5. The testing team adds another dimension to the software development by providing a different viewpoint to the product development process.

**Ques.3. When should we stop testing?**  
Ans. Testing (both manual and automated) can be stopped when one or more of the following conditions are met-  
  
1. **After test case execution** – The testing phase can be stopped when one complete cycle of test cases is executed after the last known bug fix with the agreed-upon value of pass-percentage  
  
2. **Once the testing deadline is met** – Testing can be stopped after deadlines get met with no high priority issues left in the system.  
  
3. **Based on Mean Time Between Failure (MTBF)** – MTBF is the time interval between two inherent failures. Based on stakeholder’s decisions, if the MTBF is quite large, one can stop the testing phase.  
  
4. **Based on code coverage value** – The testing phase can be stopped when the automated code coverage reaches a specific threshold value with sufficient pass-percentage and no critical bug.

**Ques.4. What is Quality Assurance and what are the different activities involved in Quality assurance?**  
Ans. [Quality assurance](https://artoftesting.com/quality-assurance) is a process-driven approach that checks if the process of developing the product is correct and conforming to all the standards. It is considered a preventive measure. This is because it identifies the weakness in the process to build software. It involves activities like document review, test case review, walk-throughs, inspection, etc.

**Ques.5. What is Quality Control and what are the different types of testing involved in QC?**  
Ans. Quality control is a product-driven approach that checks that the developed product conforms to all the specified requirements. It is considered a corrective measure as it tests the built product to find the defects. It involves different types of testing like functional testing, performance testing, usability testing, etc.

**Ques.6. What is the difference between Verification and Validation?**  
Ans. Following are the major differences between verification and validation-

|  |  |  |
| --- | --- | --- |
| **#** | **Verification** | **Validation** |
| 1. | Verification is the process of evaluating the different artifacts as well as the process of software development.  This is done in order to ensure that the product being developed will comply with the standards. | Validation is the process of validating that the developed software product conforms to the specified business requirements. |
| 2. | It is a static process of analyzing the documents and not the actual end product. | It involves dynamic testing of a software product by running it. |
| 3. | Verification is a process-oriented approach. | Validation is a product-oriented approach. |
| 4. | Answers the question – “Are we building the product right?” | Answers the question – “Are we building the right product?” |
| 5. | Errors found during verification require lesser cost/resources to get fixed as compared to be found during the validation phase. | Errors found during validation require more cost/resources. Later the error is discovered higher is the cost to fix it. |

**Ques.7. What is SDLC?**  
Ans. [SDLC](https://artoftesting.com/software-development-life-cycle-sdlc) stands for Software Development Life Cycle. It refers to all the activities performed during software development – requirement gathering, requirement analysis, designing, coding or implementation, testing, deployment, and maintenance.

**Ques.8. Explain the STLC – Software Testing life cycle.**  
Ans. The [software testing life cycle](https://artoftesting.com/software-testing-life-cycle-stlc) refers to all the activities performed during testing of a software product. The phases include-

* **Requirement analysis and validation** – In this phase, the requirements documents are analyzed and validated and the scope of testing is defined.
* **Test planning** – In this phase, test plan strategy is defined, estimation of test effort is defined along with automation strategy, and tool selection is done.
* **Test Design and Analysis** – Here, test cases are designed, test data is prepared and automation scripts are implemented.
* **Test environment setup** – A test environment closely simulating the real-world environment is prepared.
* **Test execution** – The test cases are prepared, bugs are reported and retested once resolved.
* **Test closure and reporting –**A test closure report is prepared to have the final test results summary, learning, and test metrics.

**Ques.9. What are the different types of testing?**  
Ans. Testing can broadly be defined into two types-

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

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

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

**Ques.10. What is manual testing?**  
Ans. [Manual testing](https://artoftesting.com/manual-testing) a type of testing that involves validation of the requirements of the application by executing a predefined set of test cases manually without the use of any automation tool.

**Ques.11. What is automation testing?**  
Ans. [Automation testing](https://artoftesting.com/automation-testing) is a type of [software testing](https://artoftesting.com/what-is-software-testing) that involves automated test case execution using an automation tool. It helps in reducing the test execution time as the test scripts written once, can be run automatically any number of times without any human intervention.

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

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

**Ques.13. What are some disadvantages of automation testing?**  
Ans. Some disadvantages of automation testing are-

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

**Ques.14. What is performance testing?**  
Ans. [Performance testing](https://artoftesting.com/performance-testing) is a type of non-functional testing in which the performance of the system is evaluated under expected or higher load. The various performance parameters evaluated during performance testing are – response time, reliability, resource usage, scalability, etc. The different types of performance testing are – Load, Stress, Endurance, Spike, and Volume Testing.

**Ques.15. What is a test bed?**  
Ans. A test bed is a test environment used for testing an application. A test bed configuration can consist of the hardware and software requirement of the application under test including – operating system, hardware configurations, software configurations, tomcat, database, etc.

**Ques.16. What is a test plan?**  
Ans. A [test plan](https://artoftesting.com/test-plan-document-template) is a formal document describing the scope of testing, the approach to be used, resources required and time estimate of carrying out the testing process. It is derived from the requirement documents (Software Requirement Specifications).

**Ques.17. What is a test scenario?**  
Ans. A [test scenario](https://artoftesting.com/test-scenario-examples) is derived from a use case. It is used for end to end testing of a feature of an application. A single test scenario can cater to multiple test cases. The scenario testing is particularly useful when there is time constraint while testing.

**Ques.18. What is a Test case?**  
Ans. A [test case](https://artoftesting.com/test-case) is used to test the conformance of an application with its requirement specifications. It is a set of conditions with pre-requisites, input values and expected results in a documented form.

**Ques.19. What are some attributes of a test case?**  
Ans. A test case can have the following attributes-

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

**Ques.20. What is Test data?**  
Ans. Test data is data that is used to test the software with different inputs and helps to check whether the corresponding output is as per the expected result or not. This data is created based on the business requirements.

**Ques.21. What is a Test script?**  
Ans. A test script is an automated test case written in any programming or scripting language. These are basically a set of instructions to evaluate the functioning of an application.

**Ques.22. What is Error in Software Testing?**Ans  Since we all are humans so it is obvious to make a mistake. Likewise, error is a similar case that happens in software testing due to some missing scenario in the requirements, some issues in design or some mistakes in the implementation.

**Ques.23. What is a Bug?**  
Ans. A bug is a fault in a software product **detected at the time of testing**, causing it to function in an unanticipated manner.

**Ques.24. What is a defect?**  
Ans. A defect is non-conformance with the requirement of the product **detected in production** (after the product goes live).

**Ques.25. What are some defect reporting attributes?**  
Ans. Some of the attributes of a Defect report are-

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

**Ques.26. What are some of the bug or defect management tools?**  
Ans. Some of the most widely used Defect Management tools are – Jira, Bugzilla, Redmine, Mantis, Quality Center, etc.

**Ques.27. What is defect density?**  
Ans. Defect density is the measure of the density of the defects in the system. It can be calculated by dividing the number of defects identified by the total number of lines of code(or methods or classes) in the application or program.

**Ques.28. What is defect priority?**  
Ans. A defect priority is the urgency of fixing the defect. Normally the defect priority is set on a scale of P0 to P3 with P0 defect having the most urgency to fix.

**Ques.29. What is defect severity?**  
Ans. Defect severity is the severity of the defect impacting the functionality. Based on the organization, we can have different levels of defect severity ranging from minor to critical or show stopper.

**Ques.30. Give an example of Low priority-Low severity, Low priority-High severity, High priority-Low severity, High priority-High severity defects.**  
Ans. Below are the examples for different combinations of priority and severity-

1. **Low priority-Low severity** – A spelling mistake in a page not frequently navigated by users.
2. **Low priority-High severity** – Application crashing in some very corner case.
3. **High priority-Low severity** – Slight change in logo color or spelling mistake in the company name.
4. **High priority-High severity** – Issue with login functionality.  
     
   For details, check – [Priority & Severity with Examples](https://artoftesting.com/priority-severity-bugs-defects)

**Ques.31. What is a blocker?**  
Ans. A blocker is a bug of high priority and high severity. It prevents or blocks testing of some other major portion of the application as well.

**Ques.32. What is a critical bug?**  
Ans. A critical bug is a bug that impacts a major functionality of the application and the application cannot be delivered without fixing the bug. It is different from the blocker bug as it doesn’t affect or blocks the testing of other parts of the application.

**Ques.33. Explain the bug life cycle or the different states of a bug.**  
Ans. A bug goes through the following phases in software development-

* New – A bug or defect when detected is in New state.
* Assigned – The newly detected bug when assigned to the corresponding developer is in the Assigned state.
* Open – When the developer works on the bug, the bug lies in the Open state.
* Rejected/Not a bug – A bug lies in rejected state in case the developer feels the bug is not genuine.
* Deferred – A deferred bug is one, fix of which is deferred for some time(for the next releases) based on the urgency and criticality of the bug.
* Fixed – When a bug is resolved by the developer it is marked as fixed.
* Test – When fixed the bug is assigned to the tester and during this time the bug is marked as in Test.
* Reopened – If the tester is not satisfied with the issue resolution the bug is moved to the Reopened state.
* Verified – After the Test phase, if the tester feels the bug is resolved, it is marked as verified.
* Closed – After the bug is verified, it is moved to Closed status.

**Ques.34. What are the different test design techniques?**  
Ans. Test design techniques are different standards of test designing that allow systematic and widely accepted test cases. The different test design techniques can be categorized as static test design techniques and dynamic test design techniques.

1. Static Test Design Techniques – The test design techniques which involves testing without executing the code. The various static test design techniques can be further divided into two parts manual and using tools-  
   * **Manual static design techniques**
     + Walkthrough
     + Informal reviews
     + Technical reviews
     + Audit
     + Inspection
     + Management review
   * **Static design techniques using tools**
     + Static analysis of code – It includes analysis of the different paths and flows in the application and different states of the test data.
     + Compliance with coding standard – This evaluates the compliance of the code with the different coding standards.
     + Analysis of code metrics – The tool used for static analysis is required to evaluate the different metrics like lines of code, complexity, code coverage, etc.
2. Dynamic Test Design Techniques – Dynamic test design techniques involve testing by running the system under test.
   * Specification-based – Specification-based test design techniques are also referred to as black-box testing. These involve testing based on the specification of the system under test without knowing its internal architecture.
   * Structure-based – Structure-based test design techniques are also referred to as white box testing. In these techniques, the knowledge of code or internal architecture of the system is required to carry out the testing.
   * Experienced-based – The experienced-based techniques are completely based on the experience or intuition of the tester. The two most common forms of experienced-based testing are – Adhoc testing and exploratory testing.

**Ques.35. What is Static Testing?**  
Ans.  Static testing is a kind of testing for reviewing the work products or documentation that are being created throughout the entire project. It allows reviewing the specifications, business requirements, documentation, processes and functional requirements in the initial phase of testing.  
So that the testers involved in it can understand the requirements in more detail before starting the testing lifecycle which intends to help in delivering the quality product.

**Ques.36. What is Dynamic Testing?**  
Ans. Testing performed by executing or running the application under test either manually or using automation.

**Ques.37. Explain the different types of specification-based test design techniques**?  
Ans. Specification-based test design techniques are also referred to as black-box testing. It involves testing based on the specification of the system under test without knowing its internal architecture. The different types of specification-based test design or black box testing techniques are-

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

**Ques.38. Explain equivalence class partitioning.**  
Ans. [Equivalence class partitioning](https://artoftesting.com/equivalence-class-partitioning) is a specification-based black-box testing technique. In equivalence class partitioning, a set of input data that defines different test conditions are partitioned into logically similar groups such that using even a single test data from the group for testing can be considered as similar to using all the other data in that group.  
  
For example, for testing a Square program (a program that prints the square of a number), the equivalence classes can be-  
Set of Negative numbers, whole numbers, decimal numbers, set of large numbers, etc.

**Ques.39. What is boundary value analysis?**  
Ans. [Boundary value analysis](https://artoftesting.com/boundary-value-analysis) is a software testing technique for designing test cases wherein the boundary values of the classes of the equivalence class partitioning are taken as input to the test cases e.g. if the test data lies in the range of 0-100, the boundary value analysis will include test data – 0,1, 99, 100.

**Ques.40. What is decision table testing?**  
Ans. Decision table testing is a type of specification-based test design technique or black-box testing technique in which testing is carried out using decision tables showing the application’s behavior based on different combinations of input values.  
  
Decision tables are particularly helpful in designing test cases for complex business scenarios involving verification of application with multiple combinations of input.

**Ques.41. What is a cause-effect** **graph?**  
Ans. A cause-effect graph testing is a black-box test design technique in which graphical representation of input i.e. cause and output i.e. effect is used for test designing. This technique uses different notations representing AND, OR, NOT, etc relations between the input conditions leading to output.

**Ques.42. What is state transition testing?**  
Ans. State transition testing is a black box test design technique based on a state machine model. State transition testing is based on the concept that a system can be defined as a collection of multiple states and the transition from one state to another happens because of some event.

**Ques.43. What is the use case testing?**  
Ans. A use case testing is a black-box testing approach in which testing is carried out using use cases. A use case scenario is seen as an interaction between the application and actors(users). These use cases are used for depicting requirements and hence can also serve as a basis for acceptance testing.

**Ques.44. What is Test Coverage?**Ans. It is a metric that measures the amount of testing performed on software while executing the test cases. Test coverage for any software can be calculated as the percentage of the number of test areas or coverage items covered with respect to the total number of test areas.  
  
The higher the test coverage, the more the part of the software gets covered by test cases and hence, the more effective will be the testing.

**Ques.45. What is structure-based testing?**Ans. Structure-based test design techniques are also referred to as white box testing. In these techniques, the knowledge of code or internal architecture of the system is required to carry out the testing. The various kinds of testing structure-based or white testing techniques are-

* **Statement testing** – A white box testing technique in which the test scripts are designed to execute the application’s code statements. Its coverage is measured as the line of code or statements executed by test scripts.
* **Decision testing/branch testing** – A testing technique in the test scripts is designed to execute the different decision-branches (e.g. if-else conditions) in the application. Its coverage is measured as the percentage of decision points out of the total decision points in the application.
* **Condition testing**– Condition testing is a testing approach in which we test the application with both True and False outcome for each condition. Hence for n conditions, we will have 2n test scripts.
* **Multiple condition testing** – In multiple condition testing, the different combinations of condition outcomes are tested at least once. Hence for 100% coverage, we will have 2^n test scripts. This is very exhaustive and very difficult to achieve 100% coverage.
* **Condition determination testing** – It is an optimized way of multiple condition testing in which the combinations which don’t affect the outcomes are discarded.
* **Path testing** – Testing the independent paths in the system(paths are executable statements from entry to exit points).

**Ques.46. What is code coverage?**  
Ans. Code coverage is the measure of the amount of code covered by the test scripts. It gives the idea of the part of the application covered by the test suite.

**Ques.47. What are Statement testing and statement coverage in white box testing?**  
Ans. Statement testing is a white box testing approach in which test scripts are designed to execute code statements.  
  
Statement coverage is the measure of the percentage of statements of code executed by the test scripts out of the total code statements in the application. The statement coverage is the least preferred metric for checking test coverage.

**Ques.48. What is decision testing or branch testing?**  
Ans. Decision testing or branch testing is a white box testing approach in which test coverage is measured by the percentage of decision points(e.g. if-else conditions) executed out of the total decision points in the application.

**Also, check –**[Top SQL Queries Asked in Interviews](https://artoftesting.com/sql-queries-for-interview)

**Ques.49. What are the different levels of testing?**Ans. [Testing can be performed at different levels](https://artoftesting.com/levels-of-software-testing) during the development process. Performing testing activities at multiple levels helps in the early identification of bugs. The different levels of testing are –

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

**Ques.50. What is unit testing?**  
Ans. [Unit testing](https://artoftesting.com/unit-testing) is the first level of testing and it involves testing individual modules of the software. It is usually performed by developers.

**Ques.51. What is integration testing?**  
Ans. [Integration testing](https://artoftesting.com/integration-testing) is performed after unit testing. In integration testing, we test the group of related modules. It aims at finding interfacing issues between the modules.

**Ques.52. What are the different types of integration testing?**  
Ans. The different type of integration testing is-

1. Big bang Integration Testing – In big bang integration testing, testing starts only after all the modules are integrated.
2. Top-down Integration Testing – In top-down integration, testing/integration starts from top modules to lower-level modules.
3. Bottom-up Integration Testing – In bottom-up integration, testing starts from lower-level modules to higher-level modules up in the hierarchy.
4. Hybrid Integration Testing – Hybrid integration testing is the combination of both Top-down and bottom-up integration testing. In this approach, the integration starts from the middle layer and testing is carried out in both the direction

For details check [Integration testing](https://artoftesting.com/integration-testing).

**Ques.53. What is a stub?**  
Ans. In the case of top-down integration testing, many times lower-level modules are not developed while beginning testing/integration with top-level modules. In those cases, Stubs or dummy modules are used that simulate the working of modules by providing a hard-coded or expected output based on the input values.

**Ques.54. What is a driver?**  
Ans. In the case of bottom-up integration testing, drivers are used to simulating the working of top-level modules in order to test the related modules lower in the hierarchy.

**Ques.55. What is system testing?**  
Ans. [System testing](https://artoftesting.com/system-testing) is the level of testing where the complete software is tested as a whole. The conformance of the application with its business requirements is checked in system testing.

**Ques.56. What is acceptance testing?**  
Ans. [Acceptance testing](https://artoftesting.com/acceptance-testing) is testing performed by the potential end-user or customers to check if the software conforms to the business requirements and can be accepted for use.

**Ques.57. What is UAT Testing?**  
Ans  [UAT testing](https://artoftesting.com/uat-testing) is the last phase of the testing lifecycle. Its main focus is to validate that software is working in accordance with business requirements. It also ensures that the application is user-friendly and can handle complex scenarios at its best before releasing the product to real-world users.

**Ques.58. What is End-To-End Testing?**  
Ans. End-to-End testing is a type of testing where the entire application undergoes testing, to test each functionality of the software is working as expected and there is no loophole remaining in it. It ensures that the application is user-friendly and meets the business requirements.

**Ques.59. What is alpha testing?**  
Ans. [Alpha testing](https://artoftesting.com/alpha-testing) is a type of acceptance testing that is performed testers or the internal employees of the organization at the developer site.

**Ques.60. What is beta testing?**  
Ans. [Beta testing](https://artoftesting.com/beta-testing) is the testing done by end-users at the end user’s site. It allows users to provide direct input about the software to the development company.

**Ques.61. What is Adhoc Testing?**  
Ans. [Adhoc testing](https://artoftesting.com/adhoc-testing) is an unstructured way of testing that is performed without any formal documentation or proper planning.

**Ques.62. What is monkey testing?**  
Ans. [Monkey testing](https://artoftesting.com/monkey-testing) is a type of testing that is performed randomly without any predefined test cases or test inputs.

**Ques.63. How is monkey testing different from Adhoc testing?**Ans. In the case of Adhoc testing although there are no predefined or documented test cases still testers have an understanding of the application. While in the case of monkey testing testers don’t have any understanding of the application.

**Ques.64. What is exploratory testing?**  
Ans. [Exploratory testing](https://artoftesting.com/exploratory-testing) is a type of testing in which new test cases are added and updated while exploring the system or executing test cases. Unlike scripted testing, test design and execution go parallelly in exploratory testing.

**Ques.65. What is load testing?**  
Ans. [Load testing](https://artoftesting.com/load-testing) is a type of performance testing which aims at finding an application’s performance under the expected workload. During load testing, we evaluate the response time, throughput, error rate, etc parameters of the application.

**Ques.66. What is stress testing?**  
Ans. [Stress testing](https://artoftesting.com/stress-testing) is a type of performance testing in which an application’s behavior is monitored under a higher workload than expected. Stress testing is done to find memory leaks and the robustness of the application.

**Ques.67. What is volume testing?**  
Ans. [Volume testing](https://artoftesting.com/volume-testing) is a type of performance testing in which the performance of the application is evaluated with a large amount of data. It checks the scalability of the application and helps in the identification of a bottleneck with a high volume of data.

**Ques.68. What is endurance testing or Soak testing?**  
Ans. [Endurance testing](https://artoftesting.com/endurance-testing) is a type of performance testing which aims at finding issues like memory leaks when an application is subjected to load test for a long period of time.

**Ques.69. What is spike testing?**  
Ans. [Spike testing](https://artoftesting.com/spike-testing) is a type of performance testing in which the application’s performance is measured while suddenly increasing the number of active users during the load test.

**Ques.70. What is UI testing?**  
Ans. UI or user interface testing is a type of testing that aims at finding Graphical User Interface defects in the application and checks that the GUI conforms to the specifications.

**Ques.71. What is usability testing?**  
Ans. [Usability testing](https://artoftesting.com/usability-testing) is the type of testing that aims at determining the ease of using the application. It aims at uncovering the usability defects in the application.

**Ques.72. What is Accessibility testing?**  
Ans. Accessibility testing is the type of testing which aims at determining the ease of use or operation of the application specifically for people with disabilities.

**Ques.73. What is compatibility testing?**  
Ans. [Compatibility testing](https://artoftesting.com/compatibility-testing) is validating software to see how compatible the software is with a particular environment – operating system, platform, or hardware.

**Ques.74. What is configuration testing?**  
Ans. Configuration testing is the type of testing used to evaluate the configurational requirements of the software along with the effect of changing the required configuration.

**Ques.75. What is localization** **testing?**  
Ans. Localization testing is a type of testing in which we evaluate the application’s customization(a localized version of the application) in a particular culture, locale or country.

**Ques.76. What is globalization testing?**  
Ans. Globalization testing is a type of testing in which application is evaluated for its functioning across the world in different cultures, languages, locales, and countries.

**Ques.77. What is negative testing?**  
Ans. [Negative testing](https://artoftesting.com/negative-testing) is a type of testing in which the application’s robustness(graceful exiting or error reporting) is evaluated when provided with invalid input or test data.

**Ques.78. What is security testing?**  
Ans. [Security testing](https://artoftesting.com/security-testing) is a type of testing which aims at evaluating the integrity, authentication, authorization, availability, confidentiality, and non-repudiation of the application under test.

**Ques.79. What is penetration testing?**  
Ans. Penetration testing or pen testing is a type of security testing in which application is evaluated(safely exploited) for different kinds of vulnerabilities that any hacker could exploit.

**Ques.80. What is robustness testing?**  
Ans. Robustness testing is a type of testing that is performed to find the robustness of the application i.e. the ability of the system to behave gracefully in case of erroneous test steps and test input.

**Ques.81. What is concurrency testing?**  
Ans. Concurrency testing is a multi-user testing in which an application is evaluated by analyzing the application’s behavior with concurrent users accessing the same functionality.

**Ques.82. What is backend testing?**  
Ans. Backend testing is a type of testing that involves testing the backend of the system which comprises testing the databases and the APIs in the application.

**Ques.83. What is A/B testing?**  
Ans. [A/B testing](https://artoftesting.com/ab-testing) is a type of testing in which the two variants of the software product are exposed to the end-users and on analyzing the user behavior on each variant, the better variant is chosen and used thereafter.

**Ques.84. What is risk analysis?**  
Ans. Risk analysis is the analysis of the risk identified and assigning an appropriate risk level to the defect based on its impact over the application.

**Ques.85. What is the difference between regression and retesting?**  
Ans. Regression testing involves testing the application to verify that a new code change doesn’t affect the other parts of the application. Whereas, in retesting, we verify if the fixed issue is resolved or not.

**Ques.86. What is the difference between black-box and white-box testing?**  
Ans. Black-box testing is a type of testing in which the internal architecture of the code is not required for testing. It is usually applicable for system and acceptance testing.  
  
Whereas white-box testing requires internal design and implementation knowledge of the application being tested. It is usually applicable for Unit and Integration testing.

**Ques.87. What is the difference between smoke and sanity testing?**  
Ans. The difference between smoke and sanity testing is-

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

**Ques.88. What is the difference between Release and Build?**  
Ans. A build is an executable file provided by the developers to the testing team for testing the application. It undergoes various iterations of fixing and testing until the application works as expected. Once the application becomes stable and ready for the end-users, it’s released in the market.  
  
Whereas, a release is an installable software provided to the end-users after it gets certified by the testing team. During the release of any software to the client, release notes are attached to it that includes a number of defects still open, covered user stories, change-requirements, and version of the release.

**Ques.89. What is the difference between bug leakage and bug release?**  
Ans. Bug leakage is when the tested software is released into the market and the end-user finds bugs in it. These include the bugs that got missed by the testing team during the testing phase.  
  
Whereas, bug release is when a specific version of the software is released in the market with some known bugs which are intended to get fixed in the later versions. These types of issues are of low priority and are mentioned in the release notes while sharing with the end-users.

**Ques.90. What do you mean by Defect Triage?**Ans. [Defect triage](https://artoftesting.com/defect-triage-process) is a process in which the defects are prioritized based on different factors like severity, risk, the time required to fix the bug, etc. The defect triage meeting includes the different stakeholders – the development team, testing team, project manager, BAs, etc, which decide the priority of fixing the defects.

**Ques.91. What is a test harness? Why do we need a test harness?**  
Ans. A test harness is a collection of test scripts and test data usually associated with the unit and integration testing. It involves stubs and drivers that are required for testing software modules and integrated components.

**Ques.92. What is all pair testing?**  
Ans. All pair testing is a type of testing in which the application is tested with all possible combinations of the values of input parameters.

**Ques.93. What is failover testing?**  
Ans. Failover testing is a type of testing that is used to verify the application’s ability to allocate more resources(more servers) in case of failure and transferring the processing part to the back-up system.

**Ques.94. What is fuzz testing?**  
Ans. Fuzz testing is a type of testing in which a large amount of random data is provided as input to the application in order to find security loopholes and other issues in the application.

**Ques.95. What is pilot testing?**  
Ans. Pilot testing is testing carried out as a trial by a limited number of users to evaluate the system and provide their feedback before the complete deployment is carried out.

**Ques.96. What is dev-box Testing?**  
Ans. In dev-box testing, a tester performs testing on the developer’s system to verify if the major functionalities of the application are stable and ready for testing.

**Ques.97. What is mutation testing?**  
Ans. Mutation testing is a type of white box testing in which the source code of the application is mutated to cause some defects in its working. After that, the test scripts are executed to check for their correctness by verifying the failures caused by the mutant code.

**Ques.98. What is the requirement traceability matrix(RTM)?**  
Ans. In software testing, a [requirement traceability matrix](https://artoftesting.com/requirements-traceability-matrix-rtm) is a table that relates the high-level requirements with either detailed requirements, test plans, or test cases. RTM helps in ensuring 100% test coverage.

**Ques.99. What is cyclomatic complexity?**  
Ans. Cyclomatic complexity is the measure of the number of independent paths in an application or program. This metric provides an indication of the amount of effort required to test complete functionality. It can be defined by the expression –  
**L – N + 2P**, where:  
L is the number of edges in the graph  
N is the number of nodes  
P is the number of disconnected parts

**Ques.100. What are the entry criteria in software testing?**  
Ans. A set of prerequisites that are required to kick-off the testing activity and that includes Test environment, Test tool, Test Data, database connectivity, and many more.

**Ques.101. What is exit criteria in software testing?**  
Ans. An exit criteria is a formal set of conditions that specify the agreed-upon features or state of the application in order to mark the completion of the process or product.

**Ques.102. What is the difference between testing and debugging?**  
Ans. Testing is primarily performed by the testing team in order to find the defects in the system. Whereas, debugging is an activity performed by the development team. In debugging the cause of the defect is located and fixed. Thus removing the defect and preventing any future occurrence of the defect as well.  
  
Another difference between the two is – testing can be done without any internal knowledge of software architecture. Whereas debugging requires knowledge of software architecture and coding.

**Ques.103. Explain the Agile methodology?**  
Ans. The [agile methodology](https://artoftesting.com/agile-methodology-advantages-and-disadvantages) of software development is based on an iterative and incremental approach. In this model, the application is broken down into smaller builds on which different cross-functional teamwork together providing rapid delivery along with adapting to changing needs at the same time.

**Also, check –**[Agile vs Waterfall](https://artoftesting.com/difference-between-agile-and-waterfall-model)**| Selecting the right model for your project**

**Ques.104. What is scrum?**  
Ans. A scrum is a process for implementing Agile methodology. In scrum, time is divided into sprints and on completion of sprints, a deliverable is shipped.

**Ques.105. What are the different roles in scrum?**  
Ans. The different roles in scrum are –

1. Product Owner – The product owner owns the whole development of the product, assigns tasks to the team and acts as an interface between the scrum team(development team) and the stakeholders.
2. Scrum Master – The scrum master monitors that scrum rules get followed in the team and conducts scrum meetings.
3. Scrum Team – A scrum team participate in the scrum meetings and perform the tasks assigned.

**Ques.106. What is a scrum meeting?**  
Ans. A scrum meeting is a daily meeting in the scrum process. This meeting is conducted by scrum master and update of the previous day’s work along with the next day’s task and context is defined in this meeting.

**Ques.107. Explain TDD (Test Driven Development).**  
Ans. Test-Driven Development is a software development methodology in which the development of the software is driven by test cases created for the functionality to be implemented. In TDD, first, the test cases are created and then code to pass the tests is written. Later the code is refactored as per the standards.

**Ques.108. What is the difference between Latent and Masked Defects?**  
Ans. A latent defect is an unidentified defect present in the current release but is not visible because the conditions in which the defect could be found have never met. These types of defects occur only when a particular event gets triggered which was concealing their presence.  
  
Whereas a masked defect is an existing defect that has not yet caused any failure because another error has masked it or prevented it from getting discovered.

**Ques.109. What is the PDCA cycle in software testing?**  
Ans.  PDCA cycle is a key for continuous process improvement in software development. It includes the following 4 steps-

* Plan – Plan the objectives, goals, initiatives which help to reach customer satisfaction.
* Do – It implements the plan into action. To serve the customer with better quality and satisfaction it is necessary to have a good plan to execute.
* Check – To check the progress of your plan which has been implemented. The result will show how accurate the planning had been done.
* Act – Acting upon the results to do further improvement which helps in achieving the planned goals.

**Ques.110. What is Defect Cascading?**  
And. Defect cascading is the triggering of a defect by another defect. It happens when a defect is not caught by the testing team and it gives rise to another defect.

**Ques.111. What is a test metric?**Ans.  Test Metric is a quantitative analysis that helps in monitoring the progress of a software project. Every project has its own timeline so to ensure the delivery of the project on time requires setting deliverables at different intervals and this aspect of measuring the progress is provided by test metrics.

**Ques.112. What is Context-driven testing?**Ans. Context-driven testing is the type of testing that involves adopting the test practices, methodologies and at times customizing them based on the context of the project.  
  
In this type of testing, instead of following best practices, we follow what works best for the project based on the skills, experience, and judgment of the testing team. For details check – [Context-driven methodology by James Bach](https://www.satisfice.com/blog/archives/74)

# STM: 100Q

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

According to ANSI/IEEE 1059 standard – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item. [Click here for more details.](https://www.softwaretestingmaterial.com/software-testing/)

#### ****2. What is the difference between SDET, Test Engineer, and Developer****

SDET Vs Test Engineer Vs Developer

| **Test Engineer** | **SDET** | **Developer** |
| --- | --- | --- |
| Test Engineer thinks only in the terms of pass or fail of a test case and how to break the software | SDET knows system functional objectives as well as quality objectives | Developer thinks how to develop a system and make a functionality work |
| Test Engineer works only for test life cycle, like design of test cases, and execution | SDET is involved in Designing, development, and testing | Developer is limited to Coding part and release to testing team |
| No coding knowledge is required | Dynamic skill sets, like knowledge of quality and testing and good in coding too | Only coding knowledge is required |
| Test Engineers know where repetitive work or simple data entry is present but they are not expected to minimize the repetitive tasks | SDET understands automation needs, they can code and provide a solution to the team where repetitive kind of work is killing the time. They can design framework which can help testing team to reduce repetitive test cycle or simple data entry task. | Developers don’t deal with such tasks |
| Test Engineers are not expected to reach up to code level and tune the performance | Well aware of Performance tuning and security threats , they can suggest and reach to the code and suggest where application is poor in performance, plus they can optimize the code | Developers are only expected to code the functionality which is expected by customer |

#### ****3. What are the best practices for writing test cases?****

* Write test cases with end-users perspective
* Write test steps in a simple way that anyone can follow them easily
* Make the test cases reusable
* Set the priority
* Provide a test case description, test data, expected result, precondition, postcondition.
* Write invalid test cases along with valid test cases
* Follow proper naming conventions
* Review the test cases regularly and update them if necessary.

#### ****4. How many test cases you can execute in a day?****

Be practical while answering these kind of real time manual testing interview questions. You can say like it totally depends on the test case complexity and size. Some test cases have few test steps and some have more test steps.

A sample answer is “In my previous project, we generally execute 30-40 simple test cases (like login functionality) per day, 10-20 medium test cases (like Assigning user roles) per day, and 5-10 complex test cases (complete purchase flow) per day.

#### ****5. How many Test cases you can write in a day or how much time is required to write a test case?****

Same strategy applies here too. It depends totally on the complexity of the functionality.

#### ****6. How many defects did you detect in your last project?****

Before saying how many defects you detected in your last project, first start saying about the type of project you worked and how many test cases you executed.  
I have worked for an ecommerce website where I have executed overall of 200 test cases and found around 45 defects.

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

Configuration management (CM) is a process of systems engineering to maintain system resources, computer systems, servers, software, and product’s performance in a consistent state. It helps to record all the changes made in the system and ensures that the system performs as expected even though changes are made over time.

#### ****8. Name some popular configuration management tools?****

Some of the popular configuration management tools are Ansible, Chef, Puppet, Terraform, Saltstack, etc.

#### ****9. What is Modification Request?****

Modification request (MR) in software development is used by clients to change the existing functionality of a software.

#### ****10. What is Enhancement report?****

Enhancement report (ER) in software development is used by clients to add a new feature in a software.

#### ****11. What if the software is so buggy it can’t really be tested at all?****

If the software is so buggy, the first thing we need to do is to report the bugs and categories them based on Severity. If the bugs are critical bugs then it severely affects schedules and indicates deeper problems in the software development process. So you need to let the manager know about the bugs with proper documentation as evidence.

#### ****12. What are Quality Assurance and Quality Control?****

**Quality Assurance:**Quality Assurance involves in process-oriented activities. It ensures the prevention of defects in the process used to make Software Applications. So the defects don’t arise when the Software Application is being developed.

**Quality Control:**Quality Control involves in product-oriented activities. It executes the program or code to identify the defects in the Software Application.

**Must read:** [Quality Assurance vs Quality Control](https://www.softwaretestingmaterial.com/quality-assurance-vs-quality-control/)

#### ****13. What is Verification in software testing?****

Verification is the process, to ensure that whether we are building the product right i.e., to verify the requirements which we have and to verify whether we are developing the product accordingly or not. Activities involved here are Inspections, Reviews, Walk-throughs. [Click here for more details.](https://www.softwaretestingmaterial.com/verification-and-validation/)

#### ****14. What is Validation in software testing?****

Validation is the process, whether we are building the right product i.e., to validate the product which we have developed is right or not. Activities involved in this is Testing the software application. [Click here for more details.](https://www.softwaretestingmaterial.com/verification-and-validation/)

***Don’t miss:***[***Software QA Interview Questions***](https://www.softwaretestingmaterial.com/software-qa-interview-questions-answers/)

#### ****15. What is Static Testing?****

Static Testing involves reviewing the documents to identify the defects in the early stages of SDLC. In static testing, we do code reviews, walkthroughs, peer reviews, and static analysis of a source code by using tools like StyleCop, ESLint, etc.,

***Read more:***[***How To Perform Static Testing***](https://www.softwaretestingmaterial.com/static-testing/)

#### ****16. What is Dynamic Testing?****

Dynamic testing involves the execution of code. It validates the output with the expected outcome.

**Read more:** [How To Perform Dynaic Testing](https://www.softwaretestingmaterial.com/dynamic-testing/) and [Static Testing vs Dynamic Testing](https://www.softwaretestingmaterial.com/static-testing-vs-dynamic-testing/)

#### ****17. What is White Box Testing?****

White Box Testing is also called as Glass Box, Clear Box, and Structural Testing. It is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually was done at the unit level. [Click here for more details.](https://www.softwaretestingmaterial.com/black-box-and-white-box-testing/)

Various white-box testing techniques are:

1. Statement Coverage
2. Decision Coverage
3. Condition Coverage
4. Multiple Condition Coverage

#### ****18. What is Black Box Testing?****

Black Box Testing is a [software testing](https://www.softwaretestingmaterial.com/software-testing/) method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing. [Click here for more details.](https://www.softwaretestingmaterial.com/black-box-and-white-box-testing/)

#### ****19. What is Grey Box Testing?****

Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

#### ****20. What is Positive and Negative Testing?****

**Positive Testing:** It is to determine what system supposed to do. It helps to check whether the application is justifying the requirements or not.

**Negative Testing:** It is to determine what system not supposed to do. It helps to find the defects from the software.

#### ****21. What is Test Strategy?****

Test Strategy is a high-level document (static document) and usually developed by the project manager. It is a document that captures the approach on how we go about testing the product and achieve the goals. It is normally derived from the Business Requirement Specification (BRS). Documents like Test Plan are prepared by keeping this document as a base. [Click here for more details.](https://www.softwaretestingmaterial.com/test-strategy/)

#### ****22. What is Test Plan and contents available in a Test Plan?****

Test plan document is a document which contains the plan for all the testing activities to be done to deliver a quality product. Test Plan document is derived from the Product Description, SRS, or Use Case documents for all future activities of the project. It is usually prepared by the Test Lead or Test Manager.

1. Test plan identifier
2. References
3. Introduction
4. Test items (functions)
5. Software risk issues
6. Features to be tested
7. Features not to be tested
8. Approach
9. Items pass/fail criteria
10. Suspension criteria and resolution requirements
11. Test deliverables
12. Remaining test tasks
13. Environmental needs
14. Staff and training needs
15. Responsibility
16. Schedule
17. Plan risks and contingencies
18. Approvals
19. Glossaries

[Click here for more details.](https://www.softwaretestingmaterial.com/test-plan-template/)

***Learn***[***Difference Between Test Plan vs Test Strategy***](https://www.softwaretestingmaterial.com/test-strategy-vs-test-plan/)

#### ****23. What is Test Suite?****

Test Suite is a collection of test cases. The test cases which are intended to test an application.

#### ****24. What is Test Scenario?****

Test Scenario gives the idea of what we have to test. Test Scenario is like a high-level test case.

#### ****25. What is Test Case?****

Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results. [Click here for more details.](https://www.softwaretestingmaterial.com/test-case-template-with-explanation/)

***Learn***[***Difference Between Test Case vs Test Scenario***](https://www.softwaretestingmaterial.com/test-scenario-vs-test-case/)

#### ****26. What is Test Bed?****

An environment configured for testing. Test bed consists of hardware, software, network configuration, an application under test, other related software.

#### ****27. What is Test Environment?****

Test Environment is the combination of hardware and software on which Test Team performs testing.

Example:

* Application Type: Web Application
* OS: Windows
* Web Server: IIS
* Web Page Design: Dot Net
* Client Side Validation: JavaScript
* Server Side Scripting: ASP Dot Net
* Database: MS SQL Server
* Browser: IE/FireFox/Chrome

#### ****28. What is Test Data?****

Test data is the data that is used by the testers to run the test cases. Whilst running the test cases, testers need to enter some input data. To do so, testers prepare test data. It can be prepared manually and also by using tools.

For example, To test a basic login functionality having a user id, password fields. We need to enter some data in the user id and password fields. So we need to collect some test data.

#### ****29. What is Test Harness?****

A test harness is the collection of software and test data configured to test a program unit by running it under varying conditions which involves monitoring the output with the expected output.

It contains the Test Execution Engine & Test Script Repository

#### ****30. What is Test Closure?****

Test Closure is the note prepared before test team formally completes the testing process. This note contains the total no. of test cases, total no. of test cases executed, total no. of defects found, total no. of defects fixed, total no. of bugs not fixed, total no of bugs rejected etc.,

#### ****31. What are the tasks of Test Closure activities in Software Testing?****

Test Closure activities fall into four major groups.

Test Completion Check: To ensure all tests should be either run or deliberately skipped and all known defects should be either fixed, deferred for a future release or accepted as a permanent restriction.

Test Artifacts handover: Tests and test environments should be handed over to those responsible for maintenance testing. Known defects accepted or deferred should be documented and communicated to those who will use and support the use of the system.

Lessons learned: Analyzing lessons learned to determine changes needed for future releases and projects. In retrospective meetings, plans are established to ensure that good  
practices can be repeated and poor practices are not repeated

Archiving results, logs, reports, and other documents and work products in the CMS (configuration management system).

#### ****32. What is test coverage?****

Test coverage helps in measuring the amount of testing performed by a set of tests.  
Test coverage can be done on both functional and non-functional activities. It assists testers to create tests that cover areas which are missing.

#### ****33. What is Code coverage?****

Code coverage is different from Test coverage. Code coverage is about unit testing practices that must target all areas of the code at least once. It is usually done by developers or unit testers.

Refer [Test Metrics](https://www.softwaretestingmaterial.com/test-metrics/).

#### ****34. List out Test Deliverables?****

1. Test Strategy
2. Test Plan
3. Effort Estimation Report
4. Test Scenarios
5. Test Cases/Scripts
6. Test Data
7. Requirement Traceability Matrix (RTM)
8. Defect Report/Bug Report
9. Test Execution Report
10. Graphs and Metrics
11. Test summary report
12. Test incident report
13. Test closure report
14. Release Note
15. Installation/configuration guide
16. User guide
17. Test status report
18. Weekly status report (Project manager to client)

[Click here for more details.](https://www.softwaretestingmaterial.com/test-deliverables/)

#### ****35. What are the most common components of a defect report?****

The most common components of a defect report format include the following

* Project Name
* Module Name
* Defect ID
* Defect detected on
* Defect detected by
* Priority
* Severity
* Defect resolved on
* Defect resolved by

### ****Manual Testing Interview Questions for freshers – 36-55:****

#### ****36. What are the levels of testing?****

In software testing, there are four testing levels.

1. Unit Testing or component level testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

#### ****37. What is Unit Testing?****

Unit Testing is also called Module Testing or Component Testing. It is done to check whether the individual unit or module of the source code is working properly. It is done by the developers in the developer’s environment. Learn more about [Unit Testing](https://www.softwaretestingmaterial.com/unit-testing/) in detail.

#### ****38. What is Integration Testing?****

Integration Testing is the process of testing the interface between the two software units. Integration testing is done in three ways. Big Bang Approach, Top-Down Approach, Bottom-Up Approach. Learn more about [Integration Testing](https://www.softwaretestingmaterial.com/integration-testing/) in detail.

[Click here for more details.](https://www.softwaretestingmaterial.com/types-of-testing/)

#### ****39. What is System Testing?****

Testing the fully integrated application to evaluate the system’s compliance with its specified requirements is called System Testing AKA End to End testing. Verifying the completed system to ensure that the application works as intended or not.

Check this post [Difference Between System Testing and Integration Testing](https://www.softwaretestingmaterial.com/integration-testing/#What-is-the-difference-between-Integration-Testing-and-System-Testing)

#### ****40. What is Big Bang Approach?****

Combining all the modules once and verifying the functionality after completion of individual module testing.

Top-down and bottom-up are carried out by using dummy modules known as Stubs and Drivers. These Stubs and Drivers are used to stand in for missing components to simulate data communication between modules.

#### ****41. What is Top-Down Approach?****

Testing takes place from top to bottom. High-level modules are tested first and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended. Stubs are used as a temporary module if a module is not ready for integration testing.

#### ****42. What is Bottom-Up Approach?****

It is a reciprocate of the Top-Down Approach. Testing takes place from bottom to up. Lowest level modules are tested first and then high-level modules and finally integrating the high-level modules to a low level to ensure the system is working as intended. Drivers are used as a temporary module for integration testing.

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

Integration Testing vs System Testing

| **INTEGRATION TESTING** | **SYSTEM TESTING** |
| --- | --- |
| It is a low level testing | It is a high level testing |
| It is followed by System Testing | It is followed by Acceptance Testing |
| It is performed after unit testing | It is performed after integration testing |
| Different types of integration testing are: • Top bottom integration testing • Bottom top integration testing • Big bang integration testing • Sandwich integration testing | Different types of system testing are: • Regression testing • Sanity testing • Usability testing • Retesting • Load testing • Performance testing • Maintenance testing |
| Testers perform functional testing to validate the interaction of two modules | Testers perform both functional as well as non-functional testing to evaluate the functionality, usability, performance testing etc., |
| Performed to test whether two different modules interact effectively with each other or not | Performed to test whether the product is performing as per user expectations and the required specifications |
| It can be performed by both testers and developers | It is performed by testers |
| Testing takes place on the interface of two individual modules | Testing takes place on complete software application |
| Here, we validate the interace between the individual modules. | Here, we validate the finished product. |
| Testers need to understand the interlinked modules and their interaction. | Testers need to understand the internal structure and programming language. |
| It covers only functional testing. | It covers both functional and non-functional testing. |

#### ****44. What is End-To-End Testing?****

In simple words, end-to-end testing is the process of testing software from start to end. Check this [End-To-End Testing](https://www.softwaretestingmaterial.com/end-to-end-testing-tutorial/) guide for more information. Also, refer [System Testing](https://www.softwaretestingmaterial.com/system-testing/) tutorial.

#### ****45. What is Functional Testing?****

In simple words, what the system actually does is functional testing. To verify that each function of the software application behaves as specified in the requirement document. Testing all the functionalities by providing appropriate input to verify whether the actual output is matching the expected output or not. It falls within the scope of black box testing and the testers need not concern about the source code of the application.

Learn more about [Functional Testing](https://www.softwaretestingmaterial.com/functional-testing/) here

#### ****46. What is Non-Functional Testing?****

In simple words, how well the system performs is non-functionality testing. Non-functional testing refers to various aspects of the software such as performance, load, stress, scalability, security, compatibility etc., Main focus is to improve the user experience on how fast the system responds to a request.

#### ****47. What is the difference between functional and non-functional testing?****

Functional Testing vs Non-functional testing

| **Functional Testing** | **Non-functional Testing** |
| --- | --- |
| What the system actually does is functional testing | How well the system performs is non-functionality testing |
| To ensure that your product meets customer and business requirements and doesn’t have any major bugs | To ensure that the product stands up to customer expectations |
| To verify the accuracy of the software against expected output | To verify the behavior of the software at various load conditions |
| It is performed before non-functional testing | It is performed after functional testing |
| Example of functional test case is to verify the login functionality | Example of non-functional test case is to check whether the homepage is loading in less than 2 seconds |
| Testing types are • Unit testing • Smoke testing • User Acceptance • Integration Testing • Regression testing • Localization • Globalization • Interoperability | Testing types are • Performance Testing • Volume Testing • Scalability • Usability Testing • Load Testing • Stress Testing • Compliance Testing • Portability Testing • Disaster Recover Testing |
| It can be performed either manual or automated way | It can be performed efficiently if automated |

#### ****48. What is Acceptance Testing?****

It is also known as pre-production testing.  This is done by the end-users along with the testers to validate the functionality of the application. After successful acceptance testing. Formal testing conducted to determine whether an application is developed as per the requirement. It allows the customer to accept or reject the application. Types of acceptance testing are Alpha, Beta & Gamma.

#### ****49. On what basis is the acceptance plan prepared?****

The acceptance test plan is prepared using the following inputs.

* **Requirement Document:** The requirement document specifies what exactly is needed and not needed in the existing project from the customer’s perspective.
* **Input from customer:** Input from the customer will be in the format of formal emails, informal talks, discussions, etc.,
* **Project plan:** Project plan document prepared by the project manager.

All the above three inputs act as good inputs to prepare the acceptance test plan.

#### ****50. What is Alpha Testing?****

Alpha testing is done by the in-house developers (who developed the software) and testers before we ship the software to the customers. Sometimes alpha testing is done by the client or outsourcing team with the presence of developers or testers. It is a part of [User Acceptance Testing](https://www.softwaretestingmaterial.com/user-acceptance-testing-uat/). The purpose of doing this is to find bugs before the customers start using the software.

#### ****51. What is Beta Testing?****

Beta testing is done by a limited number of end-users before delivery. It is done after the Alpha Testing. Usually, it is done in the client’s place. Learn more about [Beta Testing](https://www.softwaretestingmaterial.com/beta-testing-a-detailed-guide/) here.

#### ****52. What is Gamma Testing?****

Gamma testing is done when the software is ready for release with specified requirements. It is done at the client place. It is done directly by skipping all the in-house testing activities.

#### ****53. What is Smoke Testing?****

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as “Day 0” check. It is done at the “build level”. It helps not to waste the testing time to simply testing the whole application when the key features don’t work or the key bugs have not been fixed yet.

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

Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper. It is also called as a subset of [Regression testing](https://www.softwaretestingmaterial.com/regression-testing/). It is done at the “release level”. At times due to release time constraints rigorous regression testing can’t be done to the build, sanity testing does that part by checking main functionalities.

#### ****55. What is the difference between Sanity and Smoke Testing?****

Sanity vs Smoke Testing

| **SMOKE TESTING** | **SANITY TESTING** |
| --- | --- |
| Smoke Test is done to make sure if the build we received from the development team is testable or not | Sanity Test is done during the release phase to check for the main functionalities of the application without going deeper |
| Smoke Testing is performed by both Developers and Testers | Sanity Testing is performed by Testers alone |
| Smoke Testing exercises the entire application from end to end | Sanity Testing exercises only the particular component of the entire application |
| Smoke Testing, build may be either stable or unstable | Sanity Testing, build is relatively stable |
| It is done on initial builds. | It is done on stable builds. |
| It is a part of basic testing. | It is a part of regression testing. |
| Usually it is done every time there is a new build release. | It is planned when there is no enough time to do in-depth testing. |

### ****Manual Testing Interview Questions For Experienced – 56-75:****

#### ****56. What is Retesting?****

To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build. Say, Build 1.0 was released. Test team found some defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.

***Complete Guide:***[***Retesting***](https://www.softwaretestingmaterial.com/retesting/)

#### ****57. What is Regression Testing?****

Repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software components.

Usually, we do regression testing in the following cases:

1. New functionalities are added to the application
2. Change Requirement (In organizations, we call it as CR)
3. Defect Fixing
4. Performance Issue Fix
5. Environment change (E.g., Updating the DB from MySQL to Oracle)

***Read a detailed guide on***[***Regression Testing***](https://www.softwaretestingmaterial.com/regression-testing/)

#### ****58. What do you mean by regression and confirmation testing?****

**Regression Testing:** Testing team re-execute the tests against the modified application to make sure whether the modified code breaks anything which was working earlier.

**Confirmation Testing:** Usually testers report a bug when a test fails. Dev Team releases a new version of the software after the defect is fixed. Now the testing team will retest to make sure the reported bug is actually fixed or not.

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

Graphical User Interface Testing is to test the interface between the application and the end user.

#### ****60. What is Recovery Testing?****

Recovery testing is performed in order to determine how quickly the system can recover after the system crash or hardware failure. It comes under the type of non-functional testing.

#### ****61. What is Globalization Testing?****

Globalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.

#### ****62. What is Internationalization Testing (I18N Testing)?****

Refer Globalization Testing.

#### ****63. What is Localization Testing (L10N Testing)?****

Localization is a process of adapting globalization software for a specific region or language by adding local specific components.

#### ****64. What is Installation Testing?****

It is to check whether the application is successfully installed and it is working as expected after installation.

#### ****65. What is Formal Testing?****

It is a process where the testers test the application by having pre-planned procedures and proper documentation.

#### ****65. What is Risk Based Testing?****

Identify the modules or functionalities which are most likely cause failures and then testing those functionalities.

#### ****66. What is Compatibility Testing?****

It is to deploy and check whether the application is working as expected in a different combination of environmental components.

#### ****67. What is Exploratory Testing?****

Usually, this process will be carried out by domain experts. They perform testing just by exploring the functionalities of the application without having the knowledge of the requirements. Check our detailed guide on [Exploratory Testing](https://www.softwaretestingmaterial.com/exploratory-testing-tutorial/) and also don’t miss these popular [Exploratory Testing Tools](https://www.softwaretestingmaterial.com/exploratory-testing-tools/).

#### ****68. What is Monkey Testing?****

Perform abnormal action on the application deliberately in order to verify the stability of the application. Check our in-depth guide on [Monkey Testing](https://www.softwaretestingmaterial.com/monkey-testing/).

#### ****69. What is Usability Testing?****

To verify whether the application is user-friendly or not and was comfortably used by an end-user or not. The main focus in this testing is to check whether the end-user can understand and operate the application easily or not. An application should be self-exploratory and must not require training to operate it. Check this guide to [learn how to perform Usability Testing](https://www.softwaretestingmaterial.com/usability-testing/).

#### ****70. What is Security Testing?****

Security testing is a process to determine whether the system protects data and maintains functionality as intended.

#### ****71. What is Soak Testing?****

Running a system at high load for a prolonged period of time to identify the performance problems is called Soak Testing.

#### ****72. What is Endurance Testing?****

Endurance testing is a non-functional testing type. It is also known as Soak Testing. Refer Soak testing.

#### ****73. What is Performance Testing?****

This type of testing determines or validates the speed, scalability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, throughput, and resource-utilization levels that meet the performance objectives for the project or product.

Complete Tutorial: [Performance Testing](https://www.softwaretestingmaterial.com/performance-testing-tutorial/)

#### ****74. What is Load Testing?****

It is to verify that the system/application can handle the expected number of transactions and to verify the system/application behavior under both normal and peak load conditions.

#### ****75. What is Volume Testing?****

It is toverify that the system/application can handle a large amount of data

#### ****76. What is Stress Testing?****

It is to verify the behavior of the system once the load increases more than its design expectations.

#### ****77. What is Scalability Testing?****

Scalability testing is a type of non-functional testing. It is to determine how the application under test scales with increasing workload.

#### ****78. What is Concurrency Testing?****

Concurrency testing means accessing the application at the same time by multiple users to ensure the stability of the system. This is mainly used to identify deadlock issues.

#### ****79. What is Fuzz Testing?****

Fuzz testing is used to identify coding errors and security loopholes in an application. By inputting a massive amount of random data to the system in an attempt to make it crash to identify if anything breaks in the application.

***Don’t miss:***[***Fuzz Testing Guide***](https://www.softwaretestingmaterial.com/fuzz-testing/)

#### ****80. What is Adhoc Testing?****

Ad-hoc testing is quite opposite to the formal testing. It is an informal testing type. In Adhoc testing, testers randomly test the application without following any documents and test design techniques. This testing is primarily performed if the knowledge of testers in the application under test is very high. Testers randomly test the application without any test cases or any business requirement document.

### ****Software Testing Interview Questions – 81-100:****

#### ****81. What is Interface Testing?****

Interface testing is performed to evaluate whether two intended modules pass data and communicate correctly to one another.

#### ****82. What is Reliability Testing?****

Perform testing on the application continuously for long period of time in order to verify the stability of the application

#### ****83. What is Bucket Testing?****

Bucket testing is a method to compare two versions of an application against each other to determine which one performs better.

#### ****84. What is A/B Testing?****

Refer Bucket Testing.

#### ****85. What is Split Testing?****

Refer Bucket Testing.

#### ****86. What are the principles of Software Testing?****

1. Testing shows presence of defects
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide Paradox
6. Testing is context depending
7. Absence of error fallacy

[Click here for more details.](https://www.softwaretestingmaterial.com/principles-of-software-testing-2/)

#### ****87. What is Exhaustive Testing?****

Testing all the functionalities using all valid and invalid inputs and preconditions is known as Exhaustive testing.

#### ****88. What is Early Testing?****

Defects detected in early phases of SDLC are less expensive to fix. So conducting early testing reduces the cost of fixing defects.

#### ****89. What is Defect clustering?****

Defect clustering in software testing means that a small module or functionality contains most of the bugs or it has the most operational failures.

#### ****90. What is Pesticide Paradox?****

Pesticide Paradox in software testing is the process of repeating the same test cases, again and again, eventually, the same test cases will no longer find new bugs. So to overcome this Pesticide Paradox, it is necessary to review the test cases regularly and add or update them to find more defects.

#### ****91. What is Defect Cascading in Software Testing?****

Defect cascading in Software testing means triggering of other defects in an application. When a defect is not identified or goes unnoticed while testing, it invokes other defects. It leads to multiple defects in the later stages and results in an increase in a number of defects in the application.

For example, if there is a defect in an accounting system related to negative taxation then the negative taxation defect affects the ledger which in turn affects other reports such as Balance Sheet, Profit & Loss etc.,

#### ****92. What is the difference between Outsourced Testing and Crowdsourced Testing****

Outsourced Testing vs Crowdsourced Testing

| **Outsource Testing** | **Crowdsourced testing** |
| --- | --- |
| A dedicated team is present to handle your testing Needs we can say it’s a third party which is unknown to you, test your application or product with a fresh set of mind. | A completely unknown pool of testing resources test your application, you can judge the quality of your product on the basis of number of bugs reported. |
| Payment is done on the basis of hours spent in testing, this estimation is done prior to the testing cycle. As an example testing outsourcing costs around 20 to 40$ per hour. | Payment is done on the basis of bug reported, no of severe bugs and low priority bugs. For example severe bug cost is 15$ and low priority bug is 3$ whereas medium priority bugs are costing 5$. |
| Application data is kept confidential and this is one of the code of ethics of every testing provider company. | Since there are n number of testers working on your application and they are not legally bound with Crowd source provider company, they are not bound to keep application data confidential. There are chances of data leakage if crowd source testing is done and no assurance of data privacy. |
| Communication is quite easy, because there is one representative always present to handle to share testing status, quality of your product. | Communication is bit tricky, because you have to understand the product quality on the basis of bugs logged by testers, you have to understand the bug by talking to the tester individually. |
| Quality is not compromised, since the objective is to identify all the bugs, within time and within budget. Entire team works to achieve this milestone, They present potential and valid bugs and organization is confident enough to fix only those bugs and get assured about their product quality. | Since there is no team concept, here focus is more on Quantity rather than quality. There are chances that your application is tested by 1000 of testers of different experiences. They may log 5K bugs of different severity. So its organization’s responsibility to identify the real bugs and fix them. |
| Testing platform and environment is completely owned by outsourced company, they are well settled with all useful software, tools, management tools, OS and Devices. | Testing environment is totally dependent on individual tester, some testers are on testing on MAC machine or some are testing on Windows, some are testing on Android or some are testing on Apple. |
| Skilled testers are in the team, there are fixed no of testers in the team. Each tester is well skilled in a particular area like mobile testing, performance testing, automation testing, functional testing. | Huge no of testers, with different expertise and different years of experiences, so chances of quality bugs depends of expertise of testers. Which may be surprisingly good or bad too. |
| One team, one time zone, restricted deadline, and planned budget, in this way testing cycles are complete. | No team concept, Different time zones, no deadlines but bugs are reported very fast. |
| Bugs reported are generally predictive in nature because testers work within a scope of testing. They don’t touch few areas because it may not in their budget. | Here no limitations of testing scope, N no of testers, n no of directions of breaking the system, Due to this testing cycle goes through a real scenario, for example n number of users are accessing application they might get some security flaw in the application. |
| High paid in comparison to Crowd sourced testing but lesser than Inhouse Testing team. | Budget friendly, quick results some time real and unexpected issues are identified. |

#### ****93. What is Walk Through?****

A walkthrough is an informal meeting conducts to learn, gain understanding, and find defects. The author leads the meeting and clarifies the queries raised by the peers in the meeting.

#### ****94. What is Inspection?****

Inspection is a formal meeting lead by a trained moderator, certainly not by the author. The document under inspection is prepared and checked thoroughly by the reviewers before the meeting. In the inspection meeting, the defects found are logged and shared with the author for appropriate actions. Post inspection, a formal follow-up process is used to ensure a timely and corrective action.

#### ****95. Who are all involved in an inspection meeting?****

Author, Moderator, Reviewer(s), Scribe/Recorder and Manager.

#### ****96. What is a Defect?****

The variation between the actual results and expected results is known as a defect. If a developer finds an issue and corrects it by himself in the development phase then it’s called a defect. [Click here for more details.](https://www.softwaretestingmaterial.com/difference-between-defect-bug-error-and-failure/)

#### ****97. What is a Bug?****

If testers find any mismatch in the application/system in testing phase then they call it as Bug. [Click here for more details.](https://www.softwaretestingmaterial.com/difference-between-defect-bug-error-and-failure/)

#### ****98. What is an Error?****

We can’t compile or run a program due to a coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an error. [Click here for more details.](https://www.softwaretestingmaterial.com/difference-between-defect-bug-error-and-failure/)

#### ****99. What is a Failure?****

Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as a failure. [Click here for more details.](https://www.softwaretestingmaterial.com/difference-between-defect-bug-error-and-failure/)

#### ****100. What is Bug Severity?****

Bug/Defect severity can be defined as the impact of the bug on customer’s business. It can be Critical, Major or Minor. In simple words, how much effect will be there on the system because of a particular defect. [Click here for more details.](https://www.softwaretestingmaterial.com/what-is-the-difference-between-severity-and-priority-in-software-testing/)

#### ****101. What is Bug Priority?****

Defect priority can be defined as how soon the defect should be fixed. It gives the order in which a defect should be resolved. Developers decide which defect they should take up next based on the priority. It can be High, Medium or Low. Most of the times the priority status is set based on the customer requirement. [Click here for more details.](https://www.softwaretestingmaterial.com/what-is-the-difference-between-severity-and-priority-in-software-testing/)

#### ****102. Tell some examples of Bug Severity and Bug Priority?****

**High Priority & High Severity:**Submit button is not working on a login page and customers are unable to login to the application

**Low Priority & High Severity:**Crash in some functionality which is going to deliver after couple of releases

**High Priority & Low Severity:**Spelling mistake of a company name on the homepage

**Low Priority & Low Severity:**FAQ page takes a long time to load

[Click here for more details.](https://www.softwaretestingmaterial.com/what-is-the-difference-between-severity-and-priority-in-software-testing/)

#### ****103. What is a Critical Bug?****

A critical bug is a show stopper which means a large piece of functionality or major system component is completely broken and there is no workaround to move further.  
For example, Due to a bug in one module, we cannot test the other modules because that blocker bug has blocked other modules. Bugs which affects the customers business are considered as critical.

**Example:**

1. “Sign In” button is not working on Gmail App and Gmail users are blocked to login to their accounts.  
2. An error message pops up when a customer clicks on transfer money button in a Banking website.

#### ****104. What is the difference between a Standalone application, Client-Server application and Web application?****

**Standalone application:**

Standalone applications follow one-tier architecture. Presentation, Business, and Database layer are in one system for a single user.

**Client-Server Application:**

Client-server applications follow two-tier architecture. Presentation and Business layer are in a client system and Database layer on another server. It works majorly in Intranet.

**Web Application:**

Web server applications follow three-tier or n-tier architecture. The presentation layer is in a client system, a Business layer is in an application server and Database layer is in a Database server. It works both in Intranet and Internet.

#### ****105. What is Bug Life Cycle?****

**Bug life cycle** is also known as **Defect life cycle**. In Software Development process, the bug has a life cycle. The bug should go through the life cycle to be closed. Bug life cycle varies depends upon the tools (QC, JIRA etc.,) used and the process followed in the organization. [Click here for more details.](https://www.softwaretestingmaterial.com/bug-life-cycle/)

#### ****106. What are the different stages in a defect life cycle?****

The different stages in a bug life cycle are:

* New
* Assigned
* Open
* Test
* Moved to QA / Ready to test
* Verified
* Fixed
* Closed
* Retested
* Reopen
* Duplicate
* Deferred
* Rejected
* Cannot be fixed
* Not reproducible
* Need more information

#### ****107. What is Bug Leakage?****

A bug which is actually missed by the testing team while testing and the build was released to the Production. If now that bug (which was missed by the testing team) was found by the end user or customer then we call it as Bug Leakage.

#### ****108. What is Bug Release?****

Releasing the software to the Production with the known bugs then we call it as Bug Release. These known bugs should be included in the release note.

#### ****109. What is Defect Age?****

Defect age can be defined as the time interval between date of defect detection and date of defect closure.

Defect Age = Date of defect closure – Date of defect detection

Assume, a tester found a bug and reported it on 1 Jan 2016 and it was successfully fixed on 5 Jan 2016. So the defect age is 5 days.

#### ****110. What is Error Seeding?****

Error seeding is a process of adding known errors intendedly in a program to identify the rate of error detection. It helps in the process of estimating the tester skills of finding bugs and also to know the ability of the application (how well the application is working when it has errors.)

#### ****111. What is Error Guessing?****

Error guessing is also a method of test case design similar to error seeding. In error guessing, testers design test cases by guessing the possible errors that might occur in the software application. The intention is to catch the errors immediately.

#### ****112. What is Showstopper Defect?****

A showstopper defect is a defect which won’t allow a user to move further in the application. It’s almost like a crash.

Assume that login button is not working. Even though you have a valid username and valid password, you could not move further because the login button is not functioning.

#### ****113. What is HotFix?****

A hotfix is a build aimed at resolving a severe issue found in production.

At times, a build executed in the production evironment would have some critical errors and it would be rolled back. Now development team kept all their work aside and focus on fixing these errors immediately and release a new build to fix that in the production. This build is referred as a hotfix.

Patches and hotfixes are two distinct types of software updates. Patches are available to the public, while hotfixes are not.

Hotfixes are also known as quick-fix engineering updates (QFE updates)

#### ****114. What’s a bugfix?****

A bugfix is a build aimed at resolving a bug which is detected by the testers in the testing cycle.

#### ****115. What is a bug bounty?****

A bug bounty program lets an organization offer reward to a person who find errors in their software and report them.

Bug bounty is a concept that has existed since the internet was created. Companies started to understand how expensive it is for them to hire experts in penetration testing every time they want to find vulnerabilities on their website or application. So recently, bug bounty programs become mainstream.

The first company to catch on to this concept was Google . It launched its “Vulnerability Reward Program” in 2010 and has paid out over $4 million since then.

#### ****116. What are the different strategies for rollout to end-users?****

There are four strategies to be followed for the rollout of any software testing project are as follows:

* Pilot
* Gradual Implementation
* Phased Implementation
* Parallel Implementation

#### ****117. What is Boundary Value Analysis?****

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. The Behavior at the edge of each equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition. A boundary value for a valid partition is a valid boundary value. Similarly, a boundary value for an invalid partition is an invalid boundary value. [Click here for more details.](https://www.softwaretestingmaterial.com/boundary-value-analysis-testing-technique/)

#### ****118. What is Equivalence Class Partition?****

Equivalence Partitioning is also known as Equivalence Class Partitioning. In equivalence partitioning, inputs to the software or system are divided into groups that are expected to exhibit similar behavior, so they are likely to be proposed in the same way. Hence selecting one input from each group to design the test cases. [Click here for more details.](https://www.softwaretestingmaterial.com/equivalence-partitioning-testing-technique/)

#### ****119. What is Decision Table testing?****

Decision Table is aka Cause-Effect Table. This test technique is appropriate for functionalities which has logical relationships between inputs (if-else logic). In the Decision table technique, we deal with combinations of inputs. To identify the test cases with a decision table, we consider conditions and actions. We take conditions as inputs and actions as outputs. [Click here for more details.](https://www.softwaretestingmaterial.com/decision-table-test-design-technique/)

#### ****120. What is State Transition?****

Using state transition testing, we pick test cases from an application where we need to test different system transitions. We can apply this when an application gives a different output for the same input, depending on what has happened in the earlier state. [Click here for more details.](https://www.softwaretestingmaterial.com/state-transition-test-design-technique/)

#### ****121. What is an entry criteria?****

The prerequisites that must be achieved before commencing the testing process. [Click here for more details.](https://www.softwaretestingmaterial.com/entry-and-exit-criteria/)

#### ****122. What is an exit criteria?****

The conditions that must be met before testing should be concluded. [Click here for more details.](https://www.softwaretestingmaterial.com/entry-and-exit-criteria/)

#### ****123. What is SDLC?****

Software Development Life Cycle (SDLC) aims to produce a high-quality system that meets or exceeds customer expectations, works effectively and efficiently in the current and planned information technology infrastructure, and is inexpensive to maintain and cost-effective to enhance.

[Click here for more details.](https://www.softwaretestingmaterial.com/sdlc-software-development-life-cycle/)

#### ****124. What are the different available models of SDLC?****

1. [Waterfall](https://www.softwaretestingmaterial.com/waterfall-model-in-sdlc/)
2. [Spiral](https://www.softwaretestingmaterial.com/spiral-model-in-sdlc/)
3. [V Model](https://www.softwaretestingmaterial.com/v-model-in-sdlc/)
4. Prototype
5. [Agile](https://www.softwaretestingmaterial.com/agile-scrum-methodology/)

#### ****125. Can you do System testing at any stage of SDLC?****

We can do System Testing only when all the units are in place and working properly. It can only be done before User Acceptance Testing (UAT).

#### ****126. What is the procedure of manual testing?****

Manual testing is crucial for testing software applications more thoroughly. The procedure of manual testing comprises of the following.  
1. Planning and Control  
2. Analysis and Design  
3. Implementation and Execution  
4. Evaluating and Reporting  
5. Test Closure activities

Refer [Software Development Life Cycle (SDLC)](https://www.softwaretestingmaterial.com/sdlc-software-development-life-cycle/) & [Software Testing Life Cycle (STLC)](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/)

#### ****127. What is STLC (Software Testing Lifecycle)?****

STLC (Software Testing Life Cycle) identifies what test activities to carry out and when to accomplish those test activities. Even though testing differs between Organizations, there is a testing life cycle. [Click here for more details.](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/)

#### ****128. What are the stages in the software testing lifecycle?****

Following are the stages in the STLC.

* Requirement Analysis
* Test Planning
* Test Design
* Test Environment Setup
* Test Execution
* Test Closure

#### ****129. What is RTM?****

Requirements Traceability Matrix (RTM) is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled. We have to ensure that every requirement has atleast 1 test case. Requirement Traceability Matrix AKA Traceability Matrix or Cross Reference Matrix. [Click here for more details.](https://www.softwaretestingmaterial.com/requirements-traceability-matrix/)

#### ****130. What is Test Metrics?****

Software test metrics is to monitor and control process and product. It helps to drive the project towards our planned goals without deviation. Metrics answer different questions. It’s important to decide what questions you want answers to. [Click here for more details.](https://www.softwaretestingmaterial.com/test-metrics/)

#### ****131. When to stop testing? (Or) How do you decide when you have tested enough?****

There are many factors involved in real-time projects to decide when to stop testing.

1. Requirement coverage reaches a specified point
2. Testing deadlines or release deadlines
3. When the complete testing budget is exhausted
4. By reaching the decided pass percentage of test cases
5. The risk in the project is under an acceptable limit
6. All the high priority bugs, blockers are fixed
7. When acceptance criteria is met
8. After the Alpha and Beta testing period ends
9. Depends on Management decision

Don’t miss: [ISTQB Quiz](https://www.softwaretestingmaterial.com/istqb-quiz/)

#### ****132. What is API Testing?****

API testing is a type of [software testing](https://www.softwaretestingmaterial.com/software-testing/) that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application. In API Testing our main focus will be on a Business logic layer of the [software architecture](https://www.softwaretestingmaterial.com/software-architecture/). API testing can be performed on any software system which contains multiple APIs. API testing won’t concentrate on the look and feel of the application. API testing is entirely different from GUI Testing.

Learn [API Testing](https://www.softwaretestingmaterial.com/api-testing)

#### ****133. Which test cases are written first white boxes or black box?****

The simple answer is black-box test cases are written first.

Let’s see why black-box test cases are written first compared to white box test cases.  
Prerequisites to start writing black-box test cases are Requirement documents or design documents. These documents will be available before initiating a project.  
Prerequisites to start writing white box test cases are the internal architecture of the application. The internal architecture of the application will be available in the later part of the project i.e., designing.

#### ****134. What is the workbench concept in Software Testing?****

Workbench is a practice of documenting how a specific activity must be performed. It is often referred to as phases, steps, and tasks.

In every workbench there will be five tasks such as Input, Execute, Check, Output, and rework.

#### ****135. What is Random testing?****

In random testing is a form of black-box software testing technique where the application is testing by generating random data.

#### ****In Conclusion:****

Here I am going to conclude the post “Software Testing Interview Questions And Answers”. Final words, Bookmark this post “100 Software Testing Interview Questions” for future reference. After reading this Interview Questions for Manual Testing, if you find that we missed some important questions, please comment below we would try to include those with answers.

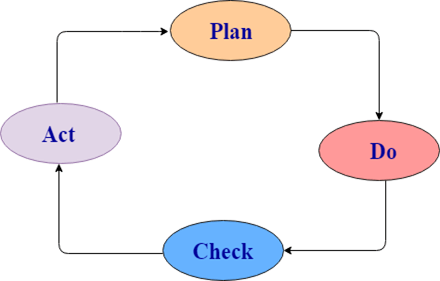
Here I have hand-picked a few posts which will help you to learn more interview related stuff along with these interview questions on manual testing.

* [Why You Choose Software Testing As A Career](https://www.softwaretestingmaterial.com/choose-software-testing-as-a-career/)
* [General Interview Questions](https://www.softwaretestingmaterial.com/6-important-interview-questions/)
* [Selenium Interview Questions](https://www.softwaretestingmaterial.com/selenium-interview-questions/)
* [Explain Test Automation Framework](https://www.softwaretestingmaterial.com/explain-test-automation-framework/)
* [Test Automation Framework Interview Questions](https://www.softwaretestingmaterial.com/test-automation-framework-interview-questions/)
* [TestNG Interview Questions](https://www.softwaretestingmaterial.com/testng-interview-questions/)
* [SQL Interview Questions](https://www.softwaretestingmaterial.com/sql-interview-questions/)
* [Manual Testing Interview Questions And Answers](https://www.softwaretestingmaterial.com/100-software-testing-interview-questions/)
* [Agile Interview Questions](https://www.softwaretestingmaterial.com/agile-testing-interview-questions/)

# 9.JavaTPoint: 50Q

### 1) What is the PDCA cycle and where testing fits in?

There are four steps in a normal software development process. In short, these steps are referred to as PDCA.



PDCA stands for Plan, Do, Check, Act.

* **Plan:** It defines the goal and the plan for achieving that goal.
* **Do/ Execute:** It depends on the plan strategy decided during the planning stage. It is done according to this phase.
* **Check:** This is the testing part of the software development phase. It is used to ensure that we are moving according to plan and getting the desired result.
* **Act:** This step is used to solve if there any issue has occurred during the check cycle. It takes appropriate action accordingly and revises the plan again.

The developers do the "planning and building" of the project while testers do the "check" part of the project.

27.4M

532

History of Java

### 2) What is the difference between the white box, black box, and gray box testing?

**Black box Testing:** The strategy of black box testing is based on requirements and specification. It requires no need of knowledge of internal path, structure or implementation of the software being tested.

**White box Testing:** White box testing is based on internal paths, code structure, and implementation of the software being tested. It requires a full and detail programming skill.

**Gray box Testing:** This is another type of testing in which we look into the box which is being tested, It is done only to understand how it has been implemented. After that, we close the box and use the black box testing.

**Following are the differences among white box, black box, and gray box testing are:**

|  |  |  |
| --- | --- | --- |
| **Black box testing** | **Gray box testing** | **White box testing** |
| Black box testing does not need the implementation knowledge of a program. | Gray box testing knows the limited knowledge of an internal program. | In white box testing, implementation details of a program are fully required. |
| It has a low granularity. | It has a medium granularity. | It has a high granularity. |
| It is also known as opaque box testing, closed box testing, input-output testing, data-driven testing, behavioral testing and functional testing. | It is also known as translucent testing. | It is also known as glass box testing, clear box testing. |
| It is a user acceptance testing, i.e., it is done by end users. | It is also a user acceptance testing. | Testers and programmers mainly do it. |
| Test cases are made by the functional specifications as internal details are not known. | Test cases are made by the internal details of a program. | Test cases are made by the internal details of a program. |

### 3)What are the advantages of designing tests early in the life cycle?

Designing tests early in the life cycle prevent defects from being in the main code.

### 4) What are the types of defects?

There are three types of defects: Wrong, missing, and extra.

**Wrong:** These defects are occurred due to requirements have been implemented incorrectly.

**Missing:** It is used to specify the missing things, i.e., a specification was not implemented, or the requirement of the customer was not appropriately noted.

**Extra:** This is an extra facility incorporated into the product that was not given by the end customer. It is always a variance from the specification but may be an attribute that was desired by the customer. However, it is considered as a defect because of the variance from the user requirements.

### 5) What is exploratory testing?

Simultaneous test design and execution against an application is called exploratory testing. In this testing, the tester uses his domain knowledge and testing experience to predict where and under what conditions the system might behave unexpectedly.

### 6) When should exploratory testing be performed?

Exploratory testing is performed as a final check before the software is released. It is a complementary activity to automated regression testing.

### 7) What are the advantages of designing tests early in the life cycle?

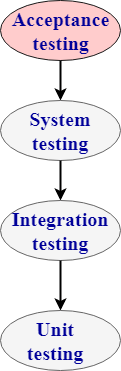
It helps you to prevent defects in the code.

### 8) Tell me about the risk-based testing.

The risk-based testing is a testing strategy that is based on prioritizing tests by risks. It is based on a detailed risk analysis approach which categorizes the risks by their priority. Highest priority risks are resolved first.

### 9) What is acceptance testing?

Acceptance testing is done to enable a user/customer to determine whether to accept a software product. It also validates whether the software follows a set of agreed acceptance criteria. In this level, the system is tested for the user acceptability.



**Types of acceptance testing are:**

1. **User acceptance testing**: It is also known as end-user testing. This type of testing is performed after the product is tested by the testers. The user acceptance testing is testing performed concerning the user needs, requirements, and business processes to determine whether the system satisfies the acceptance criteria or not.
2. **Operational acceptance testing**: An operational acceptance testing is performed before the product is released in the market. But, it is performed after the user acceptance testing.
3. **Contract and regulation acceptance testing**: In the case of contract acceptance testing, the system is tested against certain criteria and the criteria are made in a contract. In the case of regulation acceptance testing, the software application is checked whether it meets the government regulations or not.
4. **Alpha and beta testing**: Alpha testing is performed in the development environment before it is released to the customer. Input is taken from the alpha testers, and then the developer fixes the bug to improve the quality of a product. Unlike alpha testing, beta testing is performed in the customer environment. Customer performs the testing and provides the feedback, which is then implemented to improve the quality of a product.

### 10) What is accessibility testing?

Accessibility testing is used to verify whether a software product is accessible to the people having disabilities (deaf, blind, mentally disabled etc.).

### 11) What is Adhoc testing?

Ad-hoc testing is a testing phase where the tester tries to 'break' the system by randomly trying the system's functionality.

### 12) What is Agile testing?

Agile testing is a testing practice that uses agile methodologies i.e. follow test-first design paradigm.

### 13) What is API (Application Programming Interface)?

Application Programming Interface is a formalized set of software calls and routines that can be referenced by an application program to access supporting system or network services.

### 14) What do you mean by automated testing?

Testing by using software tools which execute test without manual intervention is known as automated testing. Automated testing can be used in GUI, performance, API, etc.

### 15) What is Bottom-up testing?

The Bottom-up testing is a testing approach which follows integration testing where the lowest level components are tested first, after that the higher level components are tested. The process is repeated until the testing of the top-level component.

### 16) What is Baseline Testing?

In Baseline testing, a set of tests is run to capture performance information. Baseline testing improves the performance and capabilities of the application by using the information collected and make the changes in the application. Baseline compares the present performance of the application with its previous performance.

### 17) What is Benchmark Testing?

Benchmarking testing is the process of comparing application performance with respect to the industry standard given by some other organization.

It is a standard testing which specifies where our application stands with respect to others.

### 18) Which types are testing are important for web testing?

There are two types of testing which are very important for web testing:

* **Performance testing**: Performance testing is a testing technique in which quality attributes of a system are measured such as responsiveness, speed under different load conditions and scalability. The performance testing describes which attributes need to be improved before the product is released in the market.
* **Security testing**: Security testing is a testing technique which determines that the data and resources be saved from the intruders.

### 19) What is the difference between web application and desktop application in the scenario of testing?

The difference between a web application and desktop application is that a web application is open to the world with potentially many users accessing the application simultaneously at various times, so load testing and stress testing are important. Web applications are also prone to all forms of attacks, mostly DDOS, so security testing is also very important in the case of web applications.

### 20) What is the difference between verification and validation?

Difference between verification and validation:

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Verification is Static Testing. | Validation is Dynamic Testing. |
| Verification occurs before Validation. | Validation occurs after Verification. |
| Verification evaluates plans, document, requirements and specification. | Validation evaluates products. |
| In verification, inputs are the checklist, issues list, walkthroughs, and inspection. | Invalidation testing, the actual product is tested. |
| Verification output is a set of document, plans, specification and requirement documents. | Invalidation actual product is output. |

### 21) What is the difference between Retesting and Regression Testing?

A list of differences between Retesting and Regression Testing:

|  |  |
| --- | --- |
| **Regression** | **Retesting** |
| Regression is a type of software testing that checks the code change does not affect the current features and functions of an application. | Retesting is the process of testing that checks the test cases which were failed in the final execution. |
| The main purpose of regression testing is that the changes made to the code should not affect the existing functionalities. | Retesting is applied on the defect fixes. |
| Defect verification is not an element of Regression testing. | Defect verification is an element of regression testing. |
| Automation can be performed for regression testing while manual testing could be expensive and time-consuming. | Automation cannot be performed for Retesting. |
| Regression testing is also known as generic testing. | Retesting is also known as planned testing. |
| Regression testing concern with executing test cases that was passed in earlier builds. Retesting concern with executing those test cases that are failed earlier. | Regression testing can be performed in parallel with the retesting. Priority of retesting is higher than the regression testing. |

### 22) What is the difference between preventative and reactive approaches to testing?

Preventative tests are designed earlier, and reactive tests are designed after the software has been produced.

### 23) What is the purpose of exit criteria?

The exit criteria are used to define the completion of the test level.

### 24) Why is the decision table testing used?

A decision table consists of inputs in a column with the outputs in the same column but below the inputs.

The decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combination. The reminders you get in the table explore combinations of inputs to define the output produced.

### 25) What is alpha and beta testing?

These are the key differences between alpha and beta testing:

|  |  |  |
| --- | --- | --- |
| **No.** | **Alpha Testing** | **Beta Testing** |
| 1) | It is always done by developers at the software development site. | It is always performed by customers at their site. |
| 2) | It is also performed by Independent testing team | It is not be performed by Independent testing team |
| 3) | It is not open to the market and public. | It is open to the market and public. |
| 4) | It is always performed in a virtual environment. | It is always performed in a real-time environment. |
| 5) | It is used for software applications and projects. | It is used for software products. |
| 6) | It follows the category of both white box testing and Black Box Testing. | It is only the kind of Black Box Testing. |
| 7) | It is not known by any other name. | It is also known as field testing. |

### 26) What is Random/Monkey Testing?

Random testing is also known as monkey testing. In this testing, data is generated randomly often using a tool. The data is generated either using a tool or some automated mechanism.

Random testing has some limitations:

* Most of the random tests are redundant and unrealistic.
* It needs more time to analyze results.
* It is not possible to recreate the test if you do not record what data was used for testing.

### 27) What is the negative and positive testing?

**Negative Testing:** When you put an invalid input and receive errors is known as negative testing.

**Positive Testing:** When you put in the valid input and expect some actions that are completed according to the specification is known as positive testing.

### 28) What is the benefit of test independence?

Test independence is very useful because it avoids author bias in defining effective tests.

### 29) What is the boundary value analysis/testing?

In boundary value analysis/testing, we only test the exact boundaries rather than hitting in the middle. For example: If there is a bank application where you can withdraw a maximum of 25000 and a minimum of 100. So in boundary value testing we only test above the max and below the max. This covers all scenarios.

The following figure shows the boundary value testing for the above-discussed bank application.TC1 and TC2 are sufficient to test all conditions for the bank. TC3 and TC4 are duplicate/redundant test cases which do not add any value to the testing. So by applying proper boundary value fundamentals, we can avoid duplicate test cases, which do not add value to the testing.

### 30) How would you test the login feature of a web application?

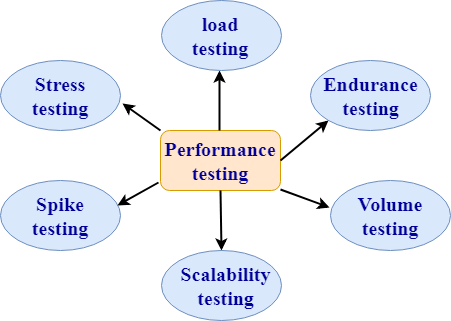
There are many ways to test the login feature of a web application:

* Sign in with valid login, Close browser and reopen and see whether you are still logged in or not.
* Sign in, then log out and then go back to the login page to see if you are truly logged out.
* Log in, then go back to the same page, do you see the login screen again?
* Session management is important. You must focus on how do we keep track of logged in users, is it via cookies or web sessions?
* Sign in from one browser, open another browser to see if you need to sign in again?
* Log in, change the password, and then log out, then see if you can log in again with the old password.

### 31) What are the types of performance testing?

**Performance testing**: Performance testing is a testing technique which determines the performance of the system such as speed, scalability, and stability under various load conditions. The product undergoes the performance testing before it gets live in the market.

**Types of software testing are:**



**1. Load testing:**

* Load testing is a testing technique in which system is tested with an increasing load until it reaches the threshold value.

#### Note: An increasing load means the increasing the number of users.

* The main purpose of load testing is to check the response time of the system with an increasing amount of load.
* Load testing is non-functional testing means that the only non-functional requirements are tested.
* Load testing is performed to make sure that the system can withstand a heavy load

**2. Stress testing:**

* Stress testing is a testing technique to check the system when hardware resources are not enough such as CPU, memory, disk space, etc.
* In case of stress testing, software is tested when the system is loaded with the number of processes and the hardware resources are less.
* The main purpose of stress testing is to check the failure of the system and to determine how to recover from this failure is known as recoverability.
* Stress testing is non-functional testing means that the only non-functional requirements are tested.

**3. Spike testing:**

* Spike testing is a subset of load testing. This type of testing checks the instability of the application when the load is varied.
* There are different cases to be considered during testing:
  + The first case is not to allow the number of users so that the system will not suffer heavy load.
  + The second case is to provide warnings to the extra joiners, and this would slow down the response time.

**4. Endurance testing:**

* Endurance testing is a subset of load testing. This type of testing checks the behavior of the system.
* Endurance testing is non-functional testing means that the only non-functional requirements are tested.
* Endurance testing is also known as Soak testing.
* Endurance testing checks the issues such as memory leak. A memory leak occurs when the program does not release its allocated memory after its use. Sometimes the application does not release its memory even after its use and this unusable memory cause memory leak. This causes an issue when the application runs for a long duration.
* Some of the main issues that are viewed during this testing are:
  + Memory leaks occurred due to an application.
  + Memory leaks occurred due to a database connection.
  + Memory leaks occurred due to a third party software.

**5. Volume testing:**

* Volume testing is a testing technique in which the system is tested when the volume of data is increased.
* Volume testing is also known as flood testing.
* Volume testing is non-functional testing means that the only non-functional requirements are tested.
* For example: If we want to apply the volume testing then we need to expand the database size, i.e., adding more data into the database table and then perform the test.

**6. Scalability testing**

* Scalability testing is a testing technique that ensures that the system works well in proportion to the growing demands of the end users.
* Following are the attributes checked during this testing:
  + Response time
  + Throughput
  + Number of users required for performance test
  + Threshold load
  + CPU usage
  + Memory usage
  + Network usage

### 32) What is the difference between functional and non-functional testing?

|  |  |  |
| --- | --- | --- |
| **Basis of comparison** | **Functional testing** | **Non-functional testing** |
| Description | Functional testing is a testing technique which checks that function of the application works under the requirement specification. | Non-functional testing checks all the non-functional aspects such as performance, usability, reliability, etc. |
| Execution | Functional testing is implemented before non-functional testing. | Non-functional testing is performed after functional testing. |
| Focus area | It depends on the customer requirements. | It depends on the customer expectations. |
| Requirement | Functional requirements can be easily defined. | Non-functional requirements cannot be easily defined. |
| Manual testing | Functional testing can be performed by manual testing. | Non-functional testing cannot be performed by manual testing. |
| Testing types | Following are the types of functional testing:   * Unit testing * Acceptance testing * Integration testing * System testing | Following are the types of non-functional testing:   * Performance testing * Load testing * Stress testing * Volume testing * Security testing * Installation testing * Recovery testing |

### 33) What is the difference between static and dynamic testing?

|  |  |
| --- | --- |
| **Static testing** | **Dynamic testing** |
| Static testing is a white box testing technique which is done at the initial stage of the software development lifecycle. | Dynamic testing is a testing process which is done at the later stage of the software development lifecycle. |
| Static testing is performed before the code deployment. | Dynamic testing is performed after the code deployment. |
| It is implemented at the verification stage. | It is implemented at the validation stage. |
| Execution of code is not done during this type of testing. | Execution of code is necessary for the dynamic testing. |
| In the case of static testing, the checklist is made for the testing process. | In the case of dynamic testing, test cases are executed. |

### 34) What is the difference between negative and positive testing?

|  |  |
| --- | --- |
| **Positive testing** | **Negative testing** |
| Positive testing means testing the application by providing valid data. | Negative testing means testing the application by providing the invalid data. |
| In case of positive testing, tester always checks the application for a valid set of values. | In the case of negative testing, tester always checks the application for the invalid set of values. |
| Positive testing is done by considering the positive point of view for example: checking the first name field by providing the value such as "Akshay". | Negative testing is done by considering the negative point of view for example: checking the first name field by providing the value such as "Akshay123". |
| It verifies the known set of test conditions. | It verifies the unknown set of conditions. |
| The positive testing checks the behavior of the system by providing the valid set of data. | The negative testing tests the behavior of the system by providing the invalid set of data. |
| The main purpose of the positive testing is to prove that the project works well according to the customer requirements. | The main purpose of the negative testing is to break the project by providing the invalid set of data. |
| The positive testing tries to prove that the project meets all the customer requirements. | The negative testing tries to prove that the project does not meet all the customer requirements. |

### 35) What are the different models available in SDLC?

There are various models available in software testing, which are the following:

* Waterfall model
* Spiral Model
* Prototype model
* Verification and validation model
* Hybrid model
* Agile model
* Rational unified process model[RUP]
* Rapid Application development [RAD]

### 36) List out the difference between smoke testing and sanity testing and dry run testing?

Following are the differences between smoke, sanity, and dry run testing:

|  |  |  |
| --- | --- | --- |
| **Smoke testing** | **Sanity testing** | **Dry-run testing** |
| It is shallow, wide and scripted testing. | It is narrow and deep and unscripted testing | A dry run testing is a process where the effects of a possible failure are internally mitigated. |
| When the builds come, we will write the automation script and execute the scripts. So it will perform automatically. | It will perform manually. | For Example, An aerospace company may conduct a Dry run of a takeoff using a new aircraft and a runway before the first test flight. |
| It will take all the essential features and perform high-level testing. | It will take some significant features and perform in-depth testing. |  |

### 37) How do we test a web application? What are the types of tests we perform on the web application?

To test any web application such as **Yahoo, Gmail**, and so on, we will perform the following testing:

* Functional testing
* Integration testing
* System testing
* Performance testing
* Compatibility testing ( test the application on the various operating systems, multiple browsers, and different version)
* Usability testing ( check whether it is user friendly)
* Ad-hoc testing
* Accessibility testing
* Smoke testing
* Regression testing
* Security testing
* Globalization testing ( only if it is developed in different languages)

### 38) Why do we need to perform compatibility testing?

We might have developed the software in one platform, and the chances are there that users might use it in the different platforms. Hence, it could be possible that they may encounter some bugs and stop using the application, and the business might get affected. Therefore, we will perform one round of Compatibility testing.

### 39) How many test cases we can write in a day?

We can tell anywhere between 2-5 test cases.

* First test case → 1st day, 2nd day.
* Second test case → 3rd day, 4th day.
* Forth test case → 5th day.
* 9-10 test cases → 19th day.

Primarily, we use to write 2-5 test cases, but in future stages we write around 6-7 because, at that time, we have the better product knowledge, we start re-using the test cases, and the experience on the product.

### 40) How many test cases can we review per day?

It would be around 7 test cases we write so that we can review 7\*3=21 test cases. And we can say that 25-30 test case per day.

### 41) How many test cases can we run in a day?

We can run around 30-55 test cases per day.

#### Note: For these types of questions (39-41), always remember the ratio: x test cases we can write, 3x test cases we can review, and 5x test cases we can execute per day.

### 42) Does the customer get a 100% bug-free product?

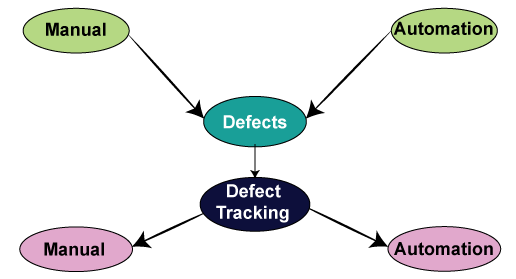
1. The testing team is not good
2. Developers are super
3. Product is old
4. All of the above

The correct answer is **testing team is not good** because sometimes the fundamentals of software testing define that no product has zero bugs.

### 43) How to track the bug manually and with the help of automation?

We can track the bug manually as:

* Identify the bug.
* Make sure that it is not duplicate (that is, check it in bug repository).
* Prepare a bug report.
* Store it in bug repository.
* Send it to the development team.
* Manage the bug life cycle (i.e., keep modifying the status).



Tracking the bug with the help of **automation** i.e., bug tracking tool:

We have various bug tracking tools available in the market, such as:

* Jira
* Bugzilla
* Mantis
* Telelogic
* Rational Clear Quest
* Bug\_track
* Quality center ( it is a test management tool, a part of it is used to track the bugs)

#### Note: Here, we have two categories of tools:

**A product based**: In the product based companies, they will use only one bug tracking tool.

**Service-based**: In service-based companies, they have many projects of different customers, and every project will have different bug tracking tools.

### 44) Why does an application have bugs?

The software can have a bug for the following reasons:

* Software complexity
* Programming errors
* If no communications are happening between the customer and the company, i.e., an application should or should not perform according to the software's needs.
* Modification in requirements
* Time pressure.

### 45) When we perform testing?

We will perform testing whenever we need to check all requirements are executed correctly or not, and to make sure that we are delivering the right quality product.

### 46) When do we stop the testing?

We can stop testing whenever we have the following:

* Once the functionality of the application is stable.
* When the time is less, then we test the necessary features, and we stop it.
* The client's budget.
* When the essential feature itself is not working correctly.

### 47) For which and all types of testing do we write test cases?

We can write test cases for the following types of testing:

|  |  |
| --- | --- |
| **Different types of testing** | Test cases |
| **Smoke testing** | In this, we will write only standard features; thus, we can pull out some test cases that have all the necessary functions. Therefore, we do not have to write a test case for smoke testing. |
| **Functional/unit testing** | Yes, we write the test case for unit testing. |
| **Integration testing** | Yes, we write the test case for integration testing. |
| **System testing** | Yes, we write the test case for system testing. |
| **Acceptance testing** | Yes, but here the customer may write the test case. |
| **Compatibility testing** | In this, we don't have to write the test case because the same test cases as above are used for testing on different platforms. |
| **Adhoc testing** | We don't write the test case for the Adhoc testing because there are some random scenarios or the ideas, which we used at the time of Adhoc time. Though, if we identify the critical bug, then we convert that scenario into a test case. |
| **Performance testing** | We might not write the test cases because we will perform this testing with the help of performance tools. |
| **Usability testing** | In this, we use the regular checklist; therefore, we don't write the test case because here we are only testing the look and feel of the application. |
| **Accessibility testing** | In accessibility testing, we also use the checklist. |
| **Reliability testing** | Here, we don't write the manual test cases as we are using the automation tool to perform reliability testing. |
| **Regression testing** | Yes, we write the test cases for functional, integration, and system testing. |
| **Recovery testing** | Yes, we write the test cases for recovery testing, and also check how the product recovers from the crash. |
| **Security testing** | Yes, we write the test case for security testing. |
| **Globalization testing:** **Localization testing** **Internationalization testing** | Yes, we write the test case for L10N testing. Yes, we write the test case for I18N testing. |

### 48) What is the difference between the traceability matrix and the test case review process?

|  |  |
| --- | --- |
| **Traceability matrix** | **Test case review** |
| In this, we will make sure that each requirement has got at least one test case. | In this, we will check whether all the scenarios are covered for the particular requirements. |

### 49) What is the difference between use case and test case?

Following are the significant differences between the use case and the test case:

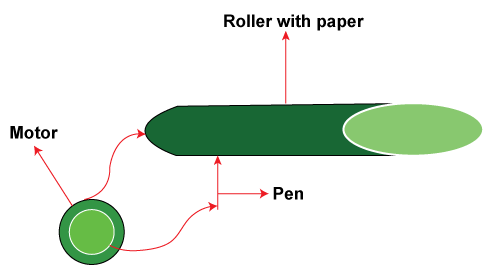
|  |  |
| --- | --- |
| **Test case** | **Use Case** |
| It is a document describing the input, action, and expected response to control whether the application is working fine based on the customer requirements. | It is a detailed description of Customer Requirements. |
| It is derived from test scenarios, Use cases, and the SRS. | It is derived from BRS/SRS. |
| While developing test cases, we can also identify loopholes in the specifications. | A business analyst or QA Lead prepares it. |

### 50) How to test a pen?

We can perform both manual and automation testing. First, we will see how we perform manual testing:

|  |  |
| --- | --- |
| **Different types of testing** | **Scenario** |
| **Smoke testing** | Checks that basic functionality is written or not. |
| **Functional/unit testing** | Check that the Refill, pen body, pen cap, and pen size as per the requirement. |
| **Integration testing** | Combine pen and cap and integrate other different sizes and see whether they work fine. |
| **Compatibility testing** | Various surfaces, multiple environments, weather conditions, and keep it in oven and then write, keep it in the freezer and write, try and write on water. |
| **Adhoc testing** | Throw the pen down and start writing, keep it vertically up and write, write on the wall. |
| **Performance testing** | Test the writing speed of the pen. |
| **Usability testing** | Check whether the pen is user friendly or not, whether we can write it for more extended periods smoothly. |
| **Accessibility testing** | Handicapped people use them. |
| **Reliability testing** | Drop it down and write, and continuously write and see whether it leaks or not |
| **Recovery testing** | Throw it down and write. |
| **Globalization testing** **Localization testing** | Price should be standard, expiry date format. |
| **Internationalize testing** | Check whether the print on the pen is as per the country language. |

Now, we will see how we perform automation testing on a pen:



For this take a roller, now put some sheets of paper on the roller, then connects the pen to the motor and switch on the motor. The pen starts writing on the paper. Once the pen has stopped writing, now observe the number of lines that it has written on each page, length of each track, and multiplying all this, so we can get for how many kilometers the pen can write.

# MindMajix: 60Q

### 1) What do you know about software testing?

Software testing refers to the validation process that helps evaluate whether the system is working according to the requirements of the business. This process helps qualify software on varying aspects, like efficiency, completeness, accuracy, and usability.

### 2) When is the ideal time to stop the process of testing?

The testing activity stops once the team has completed the following objections:

* Successful completion of the full test cycle once the final bug has been fixed
* The end date of the validation stage if no high-priority or critical defects have been left behind
* In case the team has achieved the expected level of Code Coverage (CC) ratio

|  |
| --- |
| If you want to enrich your career and become a **professional in Manual Testing**, then enroll in "[**Manual Testing Training**](https://mindmajix.com/manual-testing-training)" - This course will help you to achieve excellence in this domain. |

### 3) In software testing, what is the meaning of verification and validation?

In software testing, verification is a process that helps confirm that the development of a product is underway according to the specifications and the standard procedures of development. In this process, inspection, reviews, walkthroughs, and demos are a few stages. On the other hand, validation is all about confirming that the developed product is devoid of bugs and working as expected. In this process, functional testing and non-functional testing are two stages.

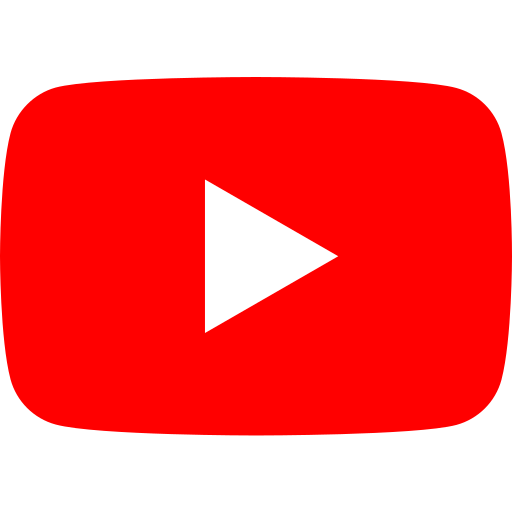
### 4) Define static testing.

Static testing is the white-box testing technique that helps direct developers to cross-check their code through a checklist to find errors.

### 5) When does static testing begin?

Developers can effortlessly begin static testing without really finalizing the program or application.

**Subscribe to explore the latest tech updates, career transformation tips, and much more.**

[**Subscribe Now**](https://bit.ly/3if9dmk)

### 6) What do you mean by black-box testing?

Black-box testing is a standard testing approach that needs testers to evaluate the software’s functionality according to the business needs. The software gets tested as a black box and gets the validation according to the end-user’s point of view.

### 7) What do you mean by a test plan?

A test plan is a plan that retains all of the potential testing activities to make sure the product is qualitative. It accumulates data from use case documents, requirements, and product descriptions.

### 8) What does the test plan include?

The test plan document comprises the following things:

* Testing objectives
* Risk factors
* Test scope
* Deliverables
* Testing the frame
* Criteria for entry and exit
* Environment
* Reasons for testing

### 9) Define test coverage?

Test coverage is referred to as the quality metric that displays the amount of testing that has been completed (in percentage). Test coverage is relevant for both non-functional and functional testing activities. Also, it is used to add those test cases that have been missing.

### 10) Can you achieve 100% testing coverage?

It is not possible to perform 100% testing of any software or product. However, can do the following steps to come closer:

* **Setting a hard limit on:**

            Percentage of test cases passed

            Number of bugs discovered

* **Setting a red flag in case**:

            There is a depletion of test budged

            There is a breach in deadlines

* **Setting a green flag in case:**

            The entire functionality is covered in test cases

            All critical and major bugs must have a ‘CLOSED’ status

### 11) Define unit testing.

Unit testing is also known with different names, such as component testing or module testing. A lot of times, developers test individual modules or units to see if they work adequately.

### 12) Define Integration testing.

Integration testing helps validate how well two or more software units will be interacting with one another. There are three different ways to validate the integration: the big bang approach, bottom-up approach, and top-down approach.

### 13) Is it possible to do system testing at any possible stage?

No, it is not possible. System testing should begin only once all of the modules are in place and are working correctly. However, it is better if performed before User Acceptance Testing (UAT).

### 14) What is the difference between a test stub and a test driver?

The test stub is referred to as a dummy program that helps integrate with an application to complete the functionality. It is relevant for such testing types that use a top-down approach. And, the test driver is a code section that calls a software component under the test. It is effective in testing that follows a bottom-up approach.

### 15) What are the different types of software testing?

Different [types of software testing](https://mindmajix.com/types-of-software-testing) include:

* Unit testing
* Functional testing
* Integration testing
* Smoke testing
* Regression testing
* Shakeout testing
* System testing
* White-box and black-box testing
* Alpha and beta testing
* Performance testing
  + Load testing
  + Stress testing
  + Endurance testing

## Software Testing Interview Questions

If you are planning to appear for a software testing interview, here are a few questions you should vary of:

### 16) Define the PDCA cycle.

In a normal software development process, there are four varying steps, referred to as PDCA. It stands for Plan, Do Check, Act.

* The plan defines the objectives and a comprehensive strategy to achieve that objective.
* Do depends upon the strategy finalized during the first stage.
* Check is the testing part of the software development stage. It is used to make sure that everything is happening as per the plan.
* The act is a step that is used to solve any issue arising during the checking cycle.

While the developers take responsibility for planning and building the project, testers handle the check part of it.

### 17) What are the benefits of designing tests early in the life cycle?

By designing tests early in the life cycle, we can effortlessly avert defects from being available in the main code.

### 18) Define exploratory testing.

Exploratory testing is referred to when design and execution take place simultaneously against an application. In this testing type, the tester uses domain knowledge and the testing experience to forecast under what conditions and where the system may behave in an unanticipated way.

### 19)How can you explain risk-based testing?

Risk-based testing is one such testing strategy that is based upon prioritizing tests by keeping risks in mind. It is based on a comprehensive risk analysis approach that further categorizes the risk by their priority. Those with the highest priority get resolved first.

### 20) What do you know about accessibility testing?

Accessibility testing comes into the picture when there is a need to verify whether the software is accessible to everybody, including those with disabilities, such as blind, deaf, mentally disabled, and more.

### 21) Define agile testing.

Agile testing is one such practice that uses agile methodologies and follows the paradigm of test-first design.

### 22) What is automated testing?

When done by software tools and executed without any manual intervention, testing is referred to as automated testing. This type of testing can be used in API, performance, GUI, and more.

### 23) Define baseline testing.

In baseline testing, a specific set of tests is run to gain information regarding the performance. Baseline testing enhances the capabilities and performance of the application with the help of collected information to make changes in the application. Baseline helps compare the application’s present performance with the previous performance.

### 24) What are the important testing types for web testing?

Two major types of testing are important for web testing, such as:

* **Performance Testing:** It is a testing technique wherein the quality attributes are evaluated, such as responsiveness, scalability, speed under varying load conditions, and more. The performance testing defines the attributes that require improvement before the launch.
* **Security Testing:**This one is a testing technique that comprehends the resources and data that must be saved from hackers or intruders.

### 25) State the difference between validation and verification****.****

|  |  |
| --- | --- |
| **Validation** | **Verification** |
| It is a dynamic testing | It is a static testing |
| Validation takes place after verification | Verification takes place before validation |
| Validation assesses products | Verification assesses documents, plans, specifications, and requirements |
| Invalidation testing, a real product is tested | In verification testing, inputs are the issues list, checklist, inspection, and walkthroughs |
| Here, the actual product is the output | Here, plans, documents, requirements, and specifications are the output. |

### 26) State the difference between reactive and preventative approaches to testing.

The reactive tests are designed once the software is developed, while the preventative tests are designed earlier than that.

### 27) What is the use of decision table testing?

A decision table comprises inputs and outputs in the same column, only that the outputs come above and inputs are placed below them. Also, decision table testing is used for the testing systems for which the specification takes the cause-effect combination or rules.

### 28) What are positive and negative testing?

Positive testing is when you put valid input and anticipate to expect certain actions that are completed as per the specifications. On the other hand, negative testing is when you put an invalid input and get an error.

### 29) What is the meaning of boundary value analysis?

In boundary value analysis, we test precise boundaries. For instance, suppose there is a bank application where you can withdraw a minimum of 100 and a maximum of 25000. Thus, we will test the above and below the maximum scenarios in boundary value analysis.

For example, if there is a bank application, you can withdraw a maximum of 25000 and a minimum of 100. So in boundary value testing, we only test above the max and below the max. This covers all scenarios.

### 30) Define the types of performance testing?

Performance testing is a technique that helps determine the system’s performance, such as stability, scalability, and speed. There are different types of software testing, such as:

* **Load Testing:** A technique wherein a system is tested with an increasing load until it attains the threshold value
* **Stress Testing:**This one is a technique that helps check the system when hardware resources are not sufficient, such as disk space, memory, CPU, and more
* **Spike Testing:**[Spike Testing](https://mindmajix.com/spike-testing) is a subset of load testing that checks the application’s instability when the load is varying
* **Endurance Testing:** This is the subset of load testing and checks the system’s behavior
* **Volume Testing:**This is a technique wherein the system gets tested when the data volume is increased
* **Scalability Testing:**It is referred to as a technique that makes sure the system is working well in proportion to the increasing demands of the users.

## Functional Testing Interview Questions

Functional Testing has various phases that must be considered while testing. We'll look at a variety of interview questions and answers in this section to help you prepare for your interview.

### 31) What is the meaning of functional testing****?****

Functional testing is a black-box testing technique where the application’s functionality is tested to come up with anticipated results by offering a specific input. The role of functional testing is not just to validate the application’s behavior but also to verify whether the application can be released or not.

### 32) Define the important steps covered in functional testing.

Some of the important steps covered in functional testing include:

* Understanding the requirement documents and resolving doubts as well as queries in the review comments
* Writing the test cases in regards to the requirement specifications by keeping every scenario in mind
* Discovering the test inputs along with requesting the test data needed to execute the test cases and check the application’s functionality
* Comprehending the real outcomes according to the tested input values
* Executing the test cases that understand whether the behavior of the application is as per the expectation or if any defects are there

### 33) What is the difference between functional and non-functional testing?

The difference between functional and non-functional testing is as follows:

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional Testing** |
| It is performed to comprehend the behavior of the system according to the client functional needs and requirements | It is the process that helps comprehend whether the performance of the system is according to the client’s requirements |
| Manual and automation testing tools are used to perform it | Effective tools are used to perform it after functional testing |
| It is quite easy to perform | It is difficult to perform |
| Functional testing covers:   * Unit Testing * Regression Testing * Smoke Testing * User Acceptance Testing * Sanity Testing * Integration testing | Non-functional testing covers:   * Performance Testing * Compatibility Testing * Load, Stress, Volume Testing * Security Testing |

### 34) What is the difference between build and release?

The build is one executable file that is regarding the application’s part handed over to the tester to test the functionalities and fix bugs. The build can also reject the testing team if it doesn’t have anything critical or major. The release is a software application that doesn’t exist in the testing phase anymore. It has been handed over to the client.

### 35) What is the bug cycle?

A bug refers to an unwanted flaw, mistake, or error occurring in the application and preventing it from generating the needed result. When any bug is discovered when testing the application, it goes through a definite cycle, from logging to resolving, known as the bug life cycle.

### 36) State some of the bug statuses.

Some of the bug statuses are:

* **New:**It is when a bug is logged for the first time
* **Assigned:** It is once a bug is logged, it will be reviewed by the team lead and assigned to the developer team
* **Open:** It is when the tester has logged the bug in the Open state, and it remains so until the developed has worked on it
* **Fixed / Resolved:**This status comes up when the developer has resolved the bug
* **Reopen:**This status comes if the tester has detected the bug again
* **Invalid / Not a Bug:**  It can be marked invalid when the developer reports an issue according to the functionality but logged because of the misinterpretation

### 37) What is data-driven testing?

Data-driven testing is a methodology where a series of test scripts, comprising test cases, gets executed consistently with the help of data sources, such as SQL database, CSV file, XML file, or Excel spreadsheet for input values. Here, the real result is compared to the anticipated one during the verification process.

### 38) What essential points should be considered when writing test cases?

Writing a test case is meant to be an essential activity for the test execution process, requiring writing skills and profound knowledge of the application to make reusable and effective test cases. Some of the essential points to be kept in mind include:

* There should be a precise understanding of the requirements before writing the test cases. There should be no assumptions or doubts.
* Every requirement should be in the form of test cases. Generally, a traceability matrix is maintained to keep track of every requirement integrated and testing completed.
* According to the requirements of document specifications, every non-functional and functional requirement should be covered.
* Test cases should be checked periodically for no redundancy or repetition.

### 39) Define automation testing.

Automation testing is one such method that uses an automation tool to execute test cases suite to improve test coverage and speed to test execution. Automation testing doesn’t need any human intervention as it executes pre-scripted tests and can compare and report results with previous test runs.

### 40) What are stress testing and load testing?

Stress testing is a performance testing type where the application goes through stress or exertion, meaning the application is executed above the break threshold to comprehend the point where it will crash. Usually, this condition comes when there is too much data or too many users.

Loading testing is a performance testing type where the application gets executed above a variety of load levels to monitor the server’s peak performance, server throughput, response time, and more.

### 41) What is volume testing?

Volume testing is a performance testing type that comprehends the response time and server throughput level when concurrent users and large data loads from the databases are put into the application or system under tests.

### 42) What are different test techniques in functional testing?

There are two different types of test techniques, such as:

* **Requirement Based Testing:**This is a functional testing form that is performed by prioritizing the needs on the basis of risk criteria. This also makes sure that all of the critical test paths get included in the process.
* **Business Process-Based Testing:**This functional testing form is performed from the perspective of the business process. The scenarios included in this are business processes' knowledge.

### 43) What is exploratory testing?

Exploratory testing means exploring or testing the application without following any procedures or schedules. When performing this testing type, testers don’t track any set patterns but use out-of-the-box thinking and varying ideas to see how the application will perform.

### 44) What are the potential login features to be tested for any web application?

The potential scenarios to perform so as to test the login feature of an application fully include:

* Checking the fields of inputs, such as username and password, for both invalid and valid values
* Entering a valid email ID with an incorrect password and an invalid email ID with a valid password
* Entering valid credentials to log into the application
* Entering the application after logging in and navigating back to the login page to check if the user is asked to log in again or not
* Sign in from a browser and open the application from another browser to check if you are still logged in
* Changing password after logging and then trying to login with the old password

### 45) What is the importance of accessibility testing?

In the present scenario, the web is dominating our lives through varying forms, including online payments, online learning, shopping, and more. Hence, to grow better, everybody should be a part of technology. With accessibility testing, disabled people can use the internet with much ease through:

* Speech recognition software
* Special keyboard
* Screen reader software
* Screen magnification software

## Manual Testing Interview Questions for 3/5/7/8 Years of Experience

If you are an experienced professional working in this domain for 3/5/7/8 years and are planning to go higher-level in your job, jotted down below are some interview questions that you should prepare.

### 46) What will be your strategy if you find a bug during testing?

When a bug is discovered, we can:

* Run additional tests to ensure the problem has a precise description
* Run tests to make sure the similar problem does not exist with  other inputs
* Once the certainty is established, can add more details and then report the bug.

### 47) How will you test if the requirements have to be freezed yet?

If the specifications haven’t been freezed for a product, can curate the test plan based on assumptions regarding the product. However, these assumptions should be well-documented in the plan.

### 48) How will you conquer the challenges that occurred because of the unavailability of documents during the testing?

In case the standard documents, such as Feature Description Document or System Requirement Specification are missing, then quality assurance will be based upon the following references, such as:

* Screenshots
* Wireframes
* A previous version of the application

Another method could be having discussions with developers and business analysts. It will help resolve doubts and open a channel for bringing transparency to the requirements.

### 49) Can you spot the difference between regression testing and retesting?

Possible differences that we can spot between regression testing and retesting are:

* Regression testing ensures the fixed bug doesn’t break other application components, while the retesting is performed to verify that the defect has been fixed.
* With regression test cases, the functionality of all or some modules is verified.
* Retesting offers higher priority in comparison to regression.

### 50) What would be the key challenges of manual testing?

As per my understanding, the following are some of the key challenges of manual testing:

* The lack of standard documentation to comprehend the application
* Testing the application entirely with the help of optimized test cases
* Unavailability of skilled testers
* Deciding the test that should be executed first
* Understanding the requirements clearly
* The ability to know when to stop the testing

### 51) Define functional and non-functional test cases?

Functional testing is the testing of the functionality of a product or an application. It helps test the software’s behavior. Based on the client's requirements, a document, known as the software specification or requirement specification, gets used to test the application. On the other hand, non-functional testing is when an application works according to the user’s expectations, regardless of the condition.

### 52) What is the meaning of STLC?

STLC is referred to as Software Testing Life Cycle. It proposes the execution of tests in a systematic and planner manner. In this model, a lot of activities take place intending to enhance the product’s quality. This method covers requirement analysis, test planning, test case development, environment setup, test execution, and test cycle closure.

### 53) What is the meaning of a fault?

 A fault is a condition that leads to the failure of software execution when performing a regarded function.

### 54) State the difference between a bug, defect, and error.

An error occurs when there is a slip in coding. This can be discovered by a manual tester when it becomes a defect. If the development team admits the defect, it becomes a bug. In case a built code misses the requirements, it turns out to be a functional failure.

### 55) What is the relationship between severity and priority?

Severity signifies the depth or gravity of a bug. It helps describe the application’s perspective. On the other hand, priority talks about the bug that must be fixed on a priority basis. It also defines the point of view of users.

### 56) List down the severity types.

The criticality or the severity of a bug can be high, medium, or low, depending on its context, such as:

* User interface defects come under low severity
* Boundary related defects and error handling defects come under medium severity
* Error handling defects, calculation defects, misinterpreted data, hardware failures, compatibility issues, control flow defects, and load conditions come under high severity

### 57) Define defect removal efficiency.

Defect Removal Efficiency (DRE) is a testing metric that indicates the development team’s efficiency in fixing errors and issues before releasing the software. It is measured as the ratio of fixed defects to the total number of discovered issues. For instance, if there were 80 discovered defects during the testing and 60 were fixed, the DRE will be 80/60 = 1.3%.

### 58) How will you perform automated testing in an environment?

Automation testing is one such process that helps execute the tests automatically. It decreases the need for human intervention significantly. Various tools can be used for it, such as WinRunner, Selenium, and QTP.  With the help of testing tools, we can speed the testing tasks. Also, these tools help us create test scripts to verify the application and generate test reports automatically.

### 59) What is the difference between software testing, quality control, and quality assurance?

Software testing is one process that ensures that the product meets the requirements of users. The objective of performing this test is to discover bugs and fix them. Hence, it helps maintain the product’s quality so that it can be delivered to the customer.

Quality Control (QC) is related to the product’s quality. It not just discovers the defects but also suggests improvements as well. Hence, the testing team is liable for QC.

Quality Assurance (QA) is a systematic and planned way of monitoring the quality of the entire process, meant to come up with a quality product. It helps track the test reports and modify the process to fulfill expectations.

### 60) State the difference between bug release and bug leakage.

Bug release is when a specific software version is released with known bugs. Generally, these bugs are of low priority or severity.

Bug leakage happens when a bug is identified by the end customer and missed by the testing team when testing the software.

# JanBask: 51Q

#### 1). What is Software Testing?

If we go by the ANSI/IEEE 1059 standards software testing is a procedure of breaking down the software to distinguish the contrasting characteristics among the existing software conditions and the required conditions (i.e. bugs and defects) and to assess the highlights of the software at hand.

#### 2). Explain the procedure for manual testing

The Manual Testing Process Comprises of the Following-

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

#### 3). Explain the tasks involved in planning and control.

Test Planning Comprises of the Following Major Tasks:

* To fix the scope and the number of risks and ascertain the goals of testing.
* To govern the test method.
* To execute the test policy and/or test tactics.

#### 4). What is Static Testing?

Static Testing includes the process of exploring the records to recognize the imperfections in the very early [stages of SDLC](https://www.janbasktraining.com/blog/what-is-sdlc-phases/).

#### 5). What is Dynamic Testing?

Dynamic testing includes the process of execution of code. It validates and approves the output with the expected results.

#### 6). What is the difference between Positive and Negative Testing?

|  |  |
| --- | --- |
| Positive Testing | Negative Testing: |
| It is done to figure out what a framework is expected to do. It checks whether the application is defending the necessities it was built for or not. | It is to figure out what framework has been tuned to not do. It finds the deformities from the product. |

#### 7). What is Use Case Testing?

The use case testing uses the use case to assess the application. So that, the tester can inspect all the functionalities of the application. Use case testing can cover a whole application.

[Read: Integration Testing Tutorial Guide for Beginner](https://www.janbasktraining.com/blog/integration-testing-tutorial/)

#### 8). What is the Test Case?

A test case is ideally used to test the conformance of a developed application in consonance with its requirement stipulations. It is a set of settings with pre-requisites, input values, and predictable results in a recognized form.

#### 9). Explain the Tasks of Test Closure Activities.

Test Closure Activities are Endowed with the following Major Tasks:

* To see which strategic deliverables are really delivered and to safeguard that all incident reports have been successfully resolved.
* To confirm and document the test ware such as writings, testing environments, etc. for future reuse.
* To deliver the test ware to the maintenance team. They will give sustenance to the software.
* To assess how the testing actually went and acquire lessons for upcoming releases and ventures.

#### 10). Can you List Down a few Characteristics of a Test Case?

A Test Case can have the Following Attributes-

* **Test Case Id –** An exceptional identifier for the test case.
* **Test Summary –** Online comments or summary for each of the test cases.
* **Description –** A Comprehensive narrative of the test case.
* **Precondition or pre-requisite –** A set of fundamentals that must be charted before implementing the test steps.
* **Test Steps –** Comprehensive steps for carrying out the test case.
* **Expected result –** The estimated result in the direction to pass the test.
* **Actual result –** The actual result received after having done the execution of the test steps.
* **Test Result –** Pass/Fail to stand on the test execution.
* **Automation Status –** Identifier for automation – whether the given application is mechanized or not.
* **Date –** The date of the test execution.
* **Executed by –** Name of the person performing the test case.

#### 11). How will you Define a Critical Bug?

A critical bug is a bug that has got the tendency to affect a majority of the functionality of the given application and the application cannot be distributed to the end client deprived of the procedure of fixing that bug. It is different from a blocker bug as it doesn’t essentially disturb or block the testing of other parts of the given application.

#### 12). What is Endurance Testing?

In this type of testing, we test the application's behavior in contrast to the load and stress put on over an application for a long period of time.

#### 13). Why do we need Localization Testing?

Localization testing generally deals with the functionality of the application and the GUI of the application.

#### 14). What is Path Testing?

Path testing is testing in which the tester guarantees that each path of the application should be affected at least once. In this testing, all the paths in the program’s source code are tested in any case once for sure.

#### 15). What is the difference between Validation and Verification?

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Progression of assessing work-products of a growth phase to control whether they fulfill the stated necessities for that stage. | The process of evaluating software during or at the end of the development process to determine whether it specified requirements. |

#### 16). Why is Software Testing Required?

Software testing is a compulsory process that ensures that the software is completely safe. And, it is good enough to be released to the market. These are the compelling reasons to show why testing is needed:

* Testing points out the errors and issues made during the developing phase.
* It reduces the coding cycles by finding issues at the beginning of the development.
* Ensures that the software has no bugs and the product meets the market standard.
* It does not result in any failures.

#### 17. What is a testbed in manual testing?

The testbed is basically an environment configured for testing. This is an environment used for testing an application and the hardware and also any software required to run the program to be tested. It includes software, hardware, network configuration, and other related software.

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

Here is the complete process of manual testing:

* Planning and Control
* Analysis and Design
* Implementation and Execution
* Evaluating exit criteria and Reporting
* Test Closure Activities

#### 19. What is API testing?

It is a kind of software testing where application programming interfaces are tested to figure out if they can meet the expectations to check the functionality, performance, reliability, and security. In simple terms, API testing is intended to reveal bugs, inconsistencies, or deviations from the expected behaviorus of an API.

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

Alpha testing is a type of software testing performance to find bugs before releasing the product to real users or the public. This is a type of user acceptance testing. Beta testing is performed by real users of the software application in a real environment.

#### 21. What’s the difference between verification and validation in testing?

Verification is a static analysis technique, here testing is done without executing the code. Validation is a dynamic analysis technique where testing is done through code execution.

#### 22. What’s the role of documentation in Manual Testing?

Documentation is a key player in effective software testing. In this testing, all the details are documented including requirement specifications, designs, inspection reports, test plans, bug reports, etc.  
Documentation helps you to estimate the testing effort you will need apart from the test coverage and tracking and tracing needs.

#### 23. What are the different types of testing?

Here is the list of different types of testing:

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

#### 24. What is the software testing life cycle?

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

* **Research for Requirement:** In the first step, the QA team understands the need in terms of what is tested and figured out the testable needs.
* **Test Planning:** This is the next step, where the test strategy is defined. Also, the ultimate project goal and scope are finalized.
* **Test Case Development:**Detailed test cases are defined and developed. The testing team also works on the test data for testing.
* **Test Environment Setup:** This is a setup of software as well as hardware for the testing teams to take care of test cases.
* **Test Execution:**This is the process of executing the code and doing a comparison of the expected and actual output.
* **Test Cycle Closure:**This last phase includes calling out the testing team member meeting and evaluating cycle completion criteria as per the test coverage, cost, quality, critical business goals, and software.

#### 25. What is functional testing?

This is a form of black-box testing focused on the software’s functional needs instead of its internal implementation. A functional need refers to required behaviour in the system, in terms of its output and input.

It validates the software against the functional needs, ignoring the non-functional attributes like usability, reliability, and performance.

### Manual Testing Interview Questions for Experienced

#### 26). What is a Test Harness?

A test harness is the gathering of software along with the test information arranged to test a program unit by running it under changing conditions which include checking the input values with the expected yield.

[Read: Unit Testing Interview Questions and Answers](https://www.janbasktraining.com/blog/unit-testing-interview-questions/)

#### 27). What is a Test Closure?

Test Closure is the note arranged before the test group formally finishes the testing procedure. This note contains the aggregate no. of experiments, total no. of experiments executed, total no. of imperfections discovered, add total no. of imperfections settled, total no. of bugs not settled, total no of bugs rejected, and so forth.

#### 28). What is Top-Down Approach?

Testing happens from top-to-bottom. High-level state modules are tested first and after that low-level modules and lastly incorporating the low-level modules to a high-level state to guarantee the framework is working as it is expected to. Stubs are utilized as an impermanent module if a module isn't prepared for integration testing.

#### 29). What is the Bottom-Up Approach?

It is the opposite of the Top-Down Approach. Testing happens from base levels to high-up levels. The lowest level modules are tried first and afterward high-level state modules and lastly coordinating the high-level state modules to a low level to guarantee the framework is filling in as it has been proposed to. Drivers are utilized as a transitory module for incorporation testing.

#### 30). Is it True That We Can do System Testing at any Stage?

No. The system testing must start only if all units are in place and are working properly. Though, it ought to happen before the ***UAT (User Acceptance testing).***

31). What are the Experience-Based Testing Techniques?

Inexperienced-based methods, individuals’ information, abilities, and foundation knowledge are prime supporters of the test conditions and experiments. The experience of both technical, as well as business, is vital, as they convey alternate points of view to the test examination and configuration process. Because of past involvement with comparable frameworks, they may have bits of knowledge into what could turn out badly, which is exceptionally valuable for testing purposes.

#### 32). When is it Ideal that the Testing is Stopped?

It depends on the level of risks associated with the system being tested. There are some criteria based on which it is ok to stop testing.

* Closing date (Testing, Release)
* The testing budget has been exhausted
* Bug rate fall below the definite level
* Test cases finished with assured percentage passed
* Alpha or beta periods for testing ends
* Reporting of code, functionality, or necessities are met to a stated point

#### 33). Explain the Concept of Semi-Random Test Cases?

Semi-random test cases are those test cases that we get when we perform arbitrary experiments and do proportionality parceling to those experiments; it evacuates repetitive experiments, along these lines giving us semi-random test cases.

#### 34). Why do We Use Decision Tables?

The techniques of equivalence dividing and boundary value analysis are regularly connected to the particular circumstances or sources of info. Nonetheless, if distinctive combinations of sources of info result in various actions being taken, this can be more difficult to indicate utilizing comparability apportioning and limiting esteem investigation, which has got a tendency to be more centered around the UI.

[Read: Manual Testing Tutorials For Beginners](https://www.janbasktraining.com/blog/manual-testing-tutorial/)

The other two determinations-based methods, choice tables, and state change testing are more centered around business rationale or business rules. A choice table is a decent method to manage blends of things (e.g. inputs). This procedure is once in a while additionally alluded to as a ’cause impact table. The purpose behind this is there is a related rationale charting system called ’cause-impact diagramming’ which was some of the time used to help determine the decision table

#### 35). Why is it That the Boundary Value Analysis Provides Good Test Cases?

This is for the reason that errors are often made during the program design of the different cases near the ‘edges of the array of values.

#### 36). What is Test Coverage?

Test coverage assesses in some specific way the quantity of testing completed by a regular set of tests (derived in some other way, e.g. using requirements-based methods). Everywhere we can tally things and can tell whether or not each of those things has been verified by some test, then we can measure coverage.

#### 37). Explain the Concept of Defect Cascading?

Defect cascading is a defect that is triggered by a different defect in the same application. In this one, the defect beseeches the other defect in the application. When a defect is extant in any stage but is not recognized, hide to other stages without getting noted. This will affect an upsurge in the number of defects.

#### 38). What is Regression Testing?

[Regression testing](https://www.janbasktraining.com/blog/regression-testing-tutorial/) checks that alteration in code has not affected the operational functionality.

If you find this blog relevant, check out Manual Testing training by Janbask Training, JanBask Training is a dynamic, highly professional, and global online training provider committed to propelling the next generation of technology learners with a whole new way of training experience.

#### 39). What is Meant by Baseline Testing?

A  baseline is an indicator of a particular benchmark that serves as a foundation of new creation.

#### 40). What is Integration Testing?

Integration testing is black-box testing. It focuses on the interface between the units to ensure that units work together to complete a specific task.

If you find this blog relevant, check out [Manual Testing training by Janbask Training](https://www.janbasktraining.com/manual-testing-training-course), JanBask Training is a dynamic, highly professional, and global online training provider committed to propelling the next generation of technology learners with a whole new way of training experience.

#### 41. What is the difference between a bug, a defect, and an error?

A bug is basically a fault in the software. It is generally found during testing time. Bugs occur due to some coding issues and lead a program to malfunction. They may also be the reason for functional issues in the product. These are fatal errors that can block the product functionality, results in a crash, or cause performance-related issues.

An error is a mistake on the software development part. Errors generally arise in software leading to a change in the overall functionality of the program.

A defect is a variance between the expected outputs and actual outputs. It is found by the developer after the product goes live. It is found after the application goes into production. A defect is also referred to as other issues with the software products, with its external behaviour, or with its internal features.

#### 42. What is regression testing?

Testing a previous test program to make sure that defects have not been introduced in unchanged areas of the software, the output of these changes made is referred to as Regression Testing.

This is a system-wide test, its main purpose is to make sure that a small change in one part of the system does not break the existing functionality.

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

When the same tests are done again and again, eventually the same tests cases will no longer find new bugs. Developers have to become extra careful in the places where testers found more defects and might not look into other areas. Here are the major methods to prevent pesticide paradox:

* To write a new set of test cases to exercise different parts of the software:
* To prepare new test cases and add them to the existing test cases.
* These are effective methods to find more defects in the area where defect numbers dropped.

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

Also known as specification-based testing, Black-Box Testing analyses the functionality of the software without knowing much about the internal structure of the item. The purpose of black-box testing is to check the functionality of the system as a whole to ensure that it is performing correctly and exceeding the user demands. Some of the techniques used in the black box testing:

* Equivalence Partitioning
* Decision Table Based Technique
* Boundary Value Analysis
* Cause-effect Graphing
* Use Case Testing

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

Here is the list of major differences between static testing and dynamic testing

* **Static Testing:**This is a white box technique that includes the process of exploring the records to recognize the imperfections.
* **Dynamic Testing:** Dynamic testing refers to the process of executing code at the later stage of the software development lifecycle. It approves the results with the expected outputs.
* **Static Testing:** It is performed before the code deployment.
* **Dynamic Testing**: It is performed after the code deployment.
* **Static Testing:** It is implemented at the verification stage.
* **Dynamic Testing:** It begins during the validation stage.

#### 46. What is ‘configuration management’?

Generally, every high-functioning organization has a master plan on the detailed description of how they are supposed to operate and accomplish tasks. Software configuration management includes the processes, policies, and tools that help in organizing, controlling, coordinating, and tracking the code, documentation, issues, designs & tools, compilers, and libraries.

#### 47. Explain STLC.

STLC stands for Software Testing Life Cycle, it is a fundamental part of SDLC used to test software as well as make sure that the quality standards are met. Generally, it includes verification activities and validation activities. In STLC, numerous activities are done in a specific order.

These are the six different phases in STLC Model:

* Requirement Analysis
* Test Planning
* Test Case Development
* Test Environment Setup
* Test Execution
* Test Cycle Closure

#### 48. Explain Endurance Testing or Soak Testing?

Endurance testing is a type of performance testing that is generally performed to check the performance of the system under constant use. The main purpose here is to determine whether a system can sustain a continuous high load or not.

#### 49. What is the function of the software testing tool “phantom”?

This freeware is used for windows, GUI automation scripting language. It allows you to take full control of windows and functions automatically. It is able to simulate any combination of keystrokes and mouse clicks and also menus, lists, and more.

#### 50. Why developers shouldn’t test the software they wrote?

Here are major reasons to show why developers are poor testers:

* Developers test the code to ensure that it works instead of testing all the ways in which it does not work.
* Because they are the ones who wrote it themselves, developers tend to be optimistic about the software and do not get the right attitude required for testing to break software.

#### 51. What are the Structure-based (white-box) testing techniques?

The structure-based testing needs a profound knowledge of the code, it needs a profound knowledge of the code as it covers testing of some structural parts of the application. This testing is focused on enhancing security, checking the flow of inputs/outputs through the application, and improving the design and usability. Some of the major white-box techniques include:

* Condition Coverage
* Statement Coverage
* Multiple Coverage
* Decision Coverage

# GreatLearning: 50Q

### ****1. How does quality control differ from quality assurance?****

**Quality Control:**Quality control is basically a product-oriented approach which is a “part of quality management focused on fulfilling quality requirements”. It also runs a program to detect if there are any defects and as well it assures that the software meets all of the requirements put forth by the stakeholders

**Quality Assurance:** Quality assurance is defined as a process-oriented approach that is a “part of quality management and focused on providing confidence that quality requirement will be fulfilled”. It mainly focuses on assuring that the methods, techniques, and processes used to create quality deliverables are applied correctly.

### ****2. What is Software Testing?****

[Software Testing](https://www.mygreatlearning.com/academy/learn-for-free/courses/software-testing-fundamentals/?gl_blog_id=36488) is a process used to verifying the correctness, identifying errors, completeness, missing requirements versus the actual requirements and quality of developed software. For that, it contains a series of activities that are conducted with an intention of finding errors in software and make them corrected before the product is released to the market. It basically checks all the requirements are working fine and available in it.

### ****3. Why is Software Testing Required?****

Software testing is a very important and mandatory process. Software testing makes sure that the software product is safe and have all that which it should have, basically it checks does the product fulfils the needs and requirements of the users and make sure that the product is good enough to be released to the market. Following are some reason which proves that software testing is important:

* It checks whether if there any defects and errors that were made during the development phases.
* It reduces the time for coding cycles by identifying issues at the initial stage of the development.
* Software testing makes sure that software application requires low maintenance and results in a more accurate, consistent and reliable manner.
* Testing ensures that the customer finds the organization reliable and satisfactory in the application performance and maintenance.
* Makes sure that software is bug-free and fulfils all the requirements and makes sure quality matches the market standards.
* Ensures that the application doesn’t stop working or face any failures in working.

### ****4. State the difference between manual testing and automation testing?****

|  |  |
| --- | --- |
| **Manual Testing** | **Automation Testing** |
| The accuracy & reliability of test cases is low in manual testing, as they are performed by humans so they are more likely to have errors in it. | Automated testing, on the other hand, is more reliable as automated tools are used to perform tests. |
| Manual testing is time-consuming as human resources perform all the tasks. | The execution in automation testing is faster than the manual as software tool execute the tests |
| Investment in manual testing is low, but Return of Investment(ROI) is low as well. | Both the Investment cost and Return of Investment, are high in automation testing. |
| Manual testing is generally preferred when the test cases run once or twice. It is also suitable for Exploratory, Usability and Adhoc Testing. | Automation testing is generally used for Regression Testing, Performance Testing, Load Testing or highly repeatable functional test cases |
| Manual testing requires human observation to find out any issues. Therefore manual testing is better in improving the customer experience. | In automation testing, there is no human observation required, that’s why there is no guarantee of a positive customer experience. |

### ****5. Write the two main categories of software testing?****

Software testing is a huge domain. It can be broadly categorized into two main categories of software testing such as:

* **Manual Testing** – This is a software testing process. The test cases are tested manually without any automated tool. It is the oldest type of software testing that can find both hidden and visible defects of the software. In this testing, the software application is tested manually by QA testers.
* **Automation Testing** – In this software testing process the test cases are tested using a special automated testing software tool. In this automation testing assistance of tools, scripts and software are used to perform test cases by repeating pre-defined actions. [Automation testing](https://www.mygreatlearning.com/academy/career-paths/automation-test-engineer/?gl_blog_id=36488) focuses on replacing manual testing which includes human activities with the systems or devices that enhance efficiency.

### ****6. What is quality control?****

Quality control is a product-oriented approach of running a program to ensure the quality of the product is maintained or improved if it has any defects, they fix them as well as making sure that the software meets all of the requirements needed by the users.

### ****7. What makes a human a good test engineer?****

A software test engineer is any professional who ensures that the product meets all the expectations and fulfil all the requirements. A software test engineer basically creates a process that would test a particular product in the software industry.

* A good test engineer should understand the priorities and should have the ability to take the point of view of the customer
* Should be passionate for quality and attention to minute details
* A good test engineer should have the ability to assert his ideas or opinions to maintain a cooperative relationship with developers
* Ability to communicate in a manner via which he can report negative things in a positive way with both technical (developers) and non-technical (customers, management) people
* Prior experience and flexibility to support whenever it is required in the software development industry is always a plus point
* Ability to take a risk whenever it is required at the time of application by judging the situations and make important decisions

### ****8. What different types of manual testing are there?****

There are six types of manual testing:

* Black Box Testing: In this level of testing, testing is done without interfering with any of its internal structure and workings.
* White Box Testing: In this level of testing the internal structure and working is tested.
* Unit Testing: In this level of testing the individual units or parts are tested.
* System Testing: In this level of testing the software product is completely tested
* Integration Testing: In this level of testing the individual units and components are combined together and are tested
* Acceptance Testing: In this level of testing the software product is tested whether it has met all requirements or not.

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

Documentation plays a crucial role in achieving effective [software testing](https://www.edureka.co/testing-with-selenium-webdriver). All the Details like requirement specifications, designs, configurations, code changes, test plans, business rules, inspection reports,  test cases, user manuals, bug reports, etc. should be documented.

Documenting the test cases will make it easier for developers to assess the testing effort developer will need along with test coverage, tracking and tracing requirement. Some commonly applied to investigate procedures of the documentation associated with software testing are:

1. Test Plan
2. Test Scenario
3. Test Case
4. Traceability Matrix

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

The different phases of the software testing life cycle are:

|  |  |
| --- | --- |
| **Phases** | **Explanation** |
| **Requirement Analysis** | QA team first learn the requirement in terms of what they will be testing & calculate the testable requirements. |
| **Test Planning** | In this phase, the test strategy is made. The objective & the scope of the project is defined. |
| **Test Case Development** | In this phase, detailed test cases are defined and evolved. For testing the data is prepared by the team. |
| **Test Environment Setup** | It is arranging software and hardware as per the requirements of the testing teams to execute test cases. |
| **Test Execution** | It is the process of executing the code and checking the expected results with the actual results. |
| **Test Cycle Closure** | It gives a summary of all the tests conducted during the software development, includes the testing team member meeting & assessing cycle completion criteria based on test coverage, quality, cost, time, critical business objectives, and software. |

### ****11. Explain the difference between alpha testing and beta testing.****

* **Alpha Testing** – Alpha Testing is a type of user acceptance testing in which software testing is performed to identify bugs and errors before releasing the product in the market or to the users.
* **Beta Testing** – Beta testing is quite opposite of the alpha testing. In this testing, the product is tested by the real users in the real environment and according to the needs the changes are made in it. It is also a type of acceptance testing.

### ****12. What are the different levels of manual testing?****

  There are Four levels of [manual testing](https://www.mygreatlearning.com/manual-testing/free-courses/?gl_blog_id=36488) are:

* **Unit testing** – Unit testing is referred to the testing of the smallest piece or the individual units of code that can be logically isolated in a system. It is mainly focused on the functional accuracy of the software product.
* **Integration Testing** – In Integration testing, the individual units or components are combined together as a group. This testing is used to test whether the combined units or components are working as they intend to when integrated The main aim here is to expose the faults in the interface between the modules.
* **System Testing** – In system testing all the components of the software are validated as a whole in order to make sure that the overall product meets the requirements specified. It evaluates all the end to end system specifications for accurate results. There are many types of system testing, including usability testing, regression testing, and functional testing.
* **User Acceptance Testing** – It is the last level of testing, acceptance testing, or UAT (user acceptance testing), which is performed after the software is completely tested. It is used to determines whether or not the software is ready to be released.

### ****13. What is a testbed in manual testing?****

An environment configuration for testing is known as a testbed. It is creating an environment suitable for testing an application, it includes the hardware as well as any software needed to run the program and to be tested. The testbed consists of specific hardware, operating system, software, network configuration, the product under test, other system software.

### ****14. Explain the procedure for manual testing?****

The manual testing process is consists of the following steps:

* Planning a test and Controlling a test
* Analysis of the requirements.
* Test cases Implementation and Execution
* Evaluating exit criteria and Reporting the bugs
* Checking completion of the test

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

There are many cases when manual testing is best suited over automation testing, like:

* **Short-time projects:**Automation testing requires high investment and planning, this testing is aimed at time saving and saving resources. But still takes time and resources to design and maintain them and the expenses are also high. For example, if you are creating a small promotional website, it can be much more efficient and easy to depend on manual testing.
* **Ad-hoc Testing:** In ad-hoc testing, there is not any specific approach. Ad-hoc testing is totally performed without planning and documentation where the understanding and insight of the tester is the only important factor. Which can be achieved by using manual testing.
* **Exploratory Test:** In this type of testing approach it mostly depends upon the tester’s knowledge, experience, analytical, logical skills, creativity, and intuition. So human involvement is a must in exploratory testing. So manual testing is very suitable here.
* **Usability Testing:** While performing usability testing, it is measured by the tester how user-friendly, efficient, or convenient the software or product is for the end-users. In usability testing Human observation plays the most important part, so manual testing sounds seems very appropriate.

### ****16. What is the test case?****

Atest case is a document that has all the requirements on it like a set of conditions or actions, the inputs, procedures that are performed on the software application in order to check whether it fulfils all the given requirements or not.

Test cases are a concept that includes a specific idea that is to be tested, without specifying the exact steps to be taken or data to be used. For example, in a test case, you document something like ‘**Test if a discount can be applied on actual price**’.

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

API testing is a type of software testing which involves testing application programming interfaces (APIs) to determine if they meet expectations for functionality, reliability, performance, and security. In simple words, API testing is planned to find out bugs, inconsistencies or deviations from the expected behaviour of an API. Commonly, applications have three separate layers:

* Presentation Layer or user interface
* Business Layer or application user interface for business logical processing
* Database Layer for modelling and manipulating data

API testing is performed at the most critical and important layer of software architecture, the Business Layer.

### ****18. What is the difference between verification and validation in testing?****

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| It is a technique, in which documents, design, code and program is checked in order to ensure the requirements are fulfilled or not. Here testing is done without executing the code. The verification process mainly includes activities like reviews and inspection. | It is a dynamic mechanism of testing and validating where testing is done by executing the code and to check the software product actually meets the customer needs or not. Validation includes activities like functional and non-functional testing techniques. |

### ****19. What is the difference between a bug and a defect?****

When the software or applications are not working as per the requirements then these faults are called a bug, it occurs during testing time. A defect is an irregularity between expected results and actual results, it is detected by the developer after the product release in the market.

### ****20. What are the advantages of manual testing?****

Advantages of manual testing are:

* This testing is cheaper in cost when compared to automated testing.
* Manual testing is easy to learn and it needs less time.
* The product analysis from the customer end is possible only with manual testing
* GUI testing can be done more precisely with the help of manual testing as visual accessibility and precedences are difficult to automate
* It is highly suitable for small applications, short-term projects where the test scripts are not going to be repeated or reused more times
* It is very suitable when the project is especially at the early stages of its development
* Highly reliable, since it is done manually where automated tests can contain errors and missed bugs.

### ****21. Explain the defect life cycle****

A**defect life cycle**, also known as Bug Life Cycle is a journey in which a defect goes through various phases during its whole lifetime. The cycle starts when a defect is found and ends when a defect is solved, after ensuring that it will not occur again

### ****22. What is regression testing? When to apply it?****

“Regression Testing is basically retesting a previously tested program to ensure that it still performs the same after making changes in it and no defects have been occurred or uncovered in unchanged areas of the software, this process is known as regression testing.”

When some small changes are made in some part of the system, a regression test is used to ensure that the system’s functionality does not break somewhere in the system. The regression test makes sure that the changes do not change the performance and working of the system. When the following events occur it is recommended to perform regression testing:

* When new functionalities are added to the system
* In case there are some changes in code or any other part of the system
* When the is a defect fix in the system.
* When the performance is not up to the mark.
* In case of environmental changes
* When there is a particular segment fix

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

**Bug** – A bug is a fault in the software which is occurred during the testing time. They come up due to some coding error and leads a program to malfunction. They can also cause a functional issue in the product. These are fatal errors that could break the functionality and can result in a system crash, or cause performance failure.

**Defect** – A defect is a difference between the expected results and actual results, it is detected by the developer when the product is live. The defect is an error found when the application goes into production. Basically, it refers to several problems or issues of the software products, with its external behaviour, or with its internal programs.

**Error** – An error basically refers to a mistake, misunderstanding, or misconception, on the part of a software developer. The developers are divided into various categories includes software engineers, programmers, analysts, and testers. For example, a design notation is misunderstood by the developer, or a variable name is incorrectly written by the programmer – leads to an error. An error normally occurs in software, and due to which there is a change in the functionality of the system.

### ****24. What are the drawbacks of manual testing?****

Drawbacks of manual testing are:

* The main drawback is it quite risky and suspectable because of the human factor, humans make mistakes. So it is not reliable.
* Some of the test types like load testing and performance testing are not possible to test manually
* Regression tests can be really time-consuming if they are done manually
* Manual testing has very limited scope when compared to automation testing
* Manual testing is not useful in very large organizations and in time-bounded projects. Due to more time consumption.

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

|  |  |
| --- | --- |
| **System Testing** | **Integration Testing** |
| System Testing tests the software application as a whole to ensure if the system is complementing the user requirements | Integration testing tests the interface between modules of the software application |
| Involves both functional and non-functional testing like sanity, usability, performance, stress and load | Only functional testing is performed to check whether the two modules when combined give the right outcome |
| It is high-level testing accomplish after the integration testing | It is low-level testing performed after unit testing |

### ****26. What is the test harness?****

A test harness is the collection of software and test information that is put together to test a program unit by running it under varying environments like stress, load, data-driven, and monitoring its behaviour, response and outputs. Test Harness is divided into two main parts:

1. A Test Execution Engine
2. Test script repository

### ****27. What is test closure?****

Test Closure is a document that gives a summary of all the tests that are conducted during the software development life cycle and it also gives a detailed analysis of the removed bugs and errors found. This document contains a report of test cases executed, total no. of experiments executed, total no. of imperfections discovered, add total no. of imperfections settled, total no. of bugs not settled, total no of bugs rejected and so on.

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

According to the pesticide paradox, if the same tests are carried out over and over again, then an outcome of this, the same test cases will no longer be able to find new bugs. Developers will be extra careful in those parts where testers found more defects and might not look into other areas.

 Methods to prevent pesticide paradox are following:

* To continually write a whole new and different set of test cases to exercise different parts of the software.
* To regularly review the existing test cases and add a new test case to them.

Using these methods, it’s possible that we can find more defects in the segment where defect numbers dropped.

### ****29. Define what is a critical bug****

A critical bug is a bug that has got the tendency to impacts a major functionality of the given application. This means a large area of functionality or major system component is completely broken and there is no method to overcome this problem to proceed forward. The Application cannot be given to the end-user unless the critical bug is fixed.

### ****30. What is the difference between Positive and Negative Testing?****

|  |  |
| --- | --- |
| **Positive Testing** | **Negative Testing** |
| **Positive testing** ensures that your application working as expected. If any error occurred during positive testing, then the test fails | **Negative testing** determines that your application can handle the invalid input or unwanted user behavior |
| In this testing, only a valid set of values are checked by the tester. | In this testing, Testers apply all their creativity and validations to the application against unwanted data |

### ****31.What is Defect Cascading in Software Testing?****

A defect that is induced by another defect is known as defect cascade. In this case, one problem in the application triggers the occurrence of another defect. Hide to further phases without being recognised when a problem is present in any step but not identified. As a result, the number of defects will rise.

Let’s have a look at a visual representation of this.

You’re working on a web page’s login module.

Phase 1 You’re developing the Register User Module for Login in phase one, and while the Mobile number is required, you can leave it blank due to a glitch and it will go undiscovered.

Phase 2 entails creating a login form with a username and password, with the password being an OTP given to the user’s registered mobile number.

Now that the Register module has a defect in which the mobile number can be left blank, this could result in a login failure or even a system error or crash if the null mobile number is not handled properly.

This is a case of defect cascading.

### ****32. What is the term ‘quality’ mean when testing?****

To find out what is meant by ‘quality’ in Quality Assurance (QA), first let’s understand what is Quality Assurance (QA). Quality assurance testing is a quality assurance (QA) or quality testing technique that ensures a company’s products or services are of the highest quality.

Organizations must also ensure that their processes for delivering the intended results meet certain quality benchmarks, as QA attempts to offer consistent results through a set of defined procedures.

In a nutshell, quality assurance (QA) encompasses all efforts centred on developing standards and procedures for ensuring that software meets a set of requirements before it is published to the public.

The important thing to remember is that QA does not entail product testing. Instead, it concentrates on the techniques in order to achieve the best results. In the end, QA activities are process-oriented. It’s all about establishing quality standards and putting in place the necessary checks and balances to ensure that the final product fulfils those criteria.

Organizations must first develop a measurable set of quality measures, as well as a procedure for verifying that those metrics are reflected in software. This means that businesses must obtain a thorough understanding of what the end-user considers to be a “excellent experience.”

These metrics must be properly defined so that the software testing team can obtain data and identify what needs to be changed. The software’s internal quality (code) must be honed to perfection before the exterior quality (the end-user experience) can be maximised.

It’s all about establishing quality standards and putting in place the necessary checks and balances to ensure that the final product fulfils those criteria.

Quality software is generally free of bugs, delivered on time and on budget, fulfils specifications and/or expectations, and is maintainable. But, then again, the term “quality” is a subjective one. It depends on who the ‘client’ is and how influential they are in the greater scheme of things. Each type of ‘client,’ for example, will have their own definition of ‘quality’ — the accounting department may describe quality in terms of revenues, while an end-user may define quality as user-friendly and bug-free.

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

Black-Box Testing, also known as specification-based testing, examines a software/functionality application’s without knowing anything about the item’s internal structure or design. The goal of this testing is to ensure that the system as a whole function properly and meets user expectations. The following are some examples of black-box testing techniques:

* Analysis of Boundary Values
* Partitioning by equivalence
* Transitional State Testing
* Decision Table Based Technique
* Graph-Based Testing
* Error Detection Method

1. **Analysis of Boundary Values**

It is a type of black-box testing that is frequently utilised and also serves as the foundation for equivalency testing. Boundary value analysis puts software to the test by using extreme test data values in test cases. BVA is used to discover defects or errors caused by data restrictions in the input.

Boundary value analysis puts software to the test by using extreme test data values in test cases. BVA is used to discover defects or errors caused by data restrictions in the input.

1. **Partitioning by equivalence**

This test case design technique splits the input into comparable classes and checks the input and output. It must be tested at least once to ensure that the data is adequately tested. It is the most thorough type of testing, and it also decreases input redundancy.

In the example above, taking inputs for a test case data will have three classes, one of which will be tested.

1. **Transitional State Testing**

This testing approach makes advantage of the system’s inputs, outputs, and state during the testing process. It compares the programme to the test data’s sequence of transitions or occurrences.

It tests for behavioural changes of a system in a certain state or another state while keeping the same inputs, depending on the type of software being tested.

A login screen, for example, will allow you to enter your username and password three times before rejecting you. The user will be directed to the login page for each invalid password. The user will be taken to an error page after the third try. This state transition approach takes into account the system’s numerous states and inputs in order to pass just the proper testing sequence.

1. **Decision Table Based Technique**

This technique produces test cases from various contexts. It evaluates multiple test scenarios in a decision table structure, where each condition is evaluated and satisfied, in order to pass the test and produce accurate results. When there are various input combinations and possibilities, it is preferable.

When placing an order, for example, food delivery software checks various payment methods as input and makes a decision based on the table.  
Case 1: If the end-user has a card, the system will skip checking for cash or coupons and go straight to placing the order.

Case 2: If the customer has a coupon, no credit card or cash will be checked, and action will be taken.

Case 3: if the end-user has cash, action will be taken.

Case 4: No action will be performed if the end-user is unable to supply anything.

1. **Graph-Based Testing:**

Graph-based testing is a type of testing that uses graph

It’s similar to a decision-based test case design technique in that it considers the relationship between input instances and linkages.

1. **Error Detection Method:**

This approach of developing test cases includes taking accurate predictions about the output and input in order to remedy any system flaws. It depends on the talents and judgement of the tester.

This method compares and validates the findings using two different versions of the same software.

### ****34. What is white box testing, and what are the various techniques?****

White-Box Testing, also known as structure-based testing, necessitates a deep understanding of the code. The goal of this testing is to improve security, input/output flow across the programme, and design and usability. The following are some examples of white-box testing methods:

Statement Coverage: This approach dictates that every single statement in the code be evaluated at least once throughout the software engineering testing process.

Branch Coverage: This method explores every possible path that a software programme may take (if-else and other conditional loops).

In addition to the above-mentioned coverage types, there are also Condition Coverage, Multiple Condition Coverage, Path Coverage, and Function Coverage. Each approach has its own set of benefits and is designed to test (cover) every part of software code. With Statement and Branch coverage, you can often obtain 80-90 percent code coverage, which is sufficient.

* Statement Coverage
* Decision Coverage
* Condition Coverage
* Multiple Condition Coverage
* Finite State Machine Coverage
* Path Coverage
* Control flow testing
* Data flow testing

### ****35. What is Experience-based testing techniques?****

Discovery, research, and learning are all part of experience-based testing. The tester is continually studying and analysing the product and applying his abilities, traits, and experience to build test techniques and test cases in order to carry out the necessary testing. Various strategies for testing based on experience include:

* Exploratory Testing
* Error Guessing
* Random Testing

1. **Random Testing**: This is a black box testing technique in which all test inputs are produced at random (sometimes with the use of a tool) and then used for testing.
2. **Exploratory Testing**: This is a hands-on approach in which testers conduct as many tests as possible with as little planning as possible.
3. **Error Guessing**: This is a Software Testing technique for estimating the likelihood of an error occurring in the code. The major goal of this technique is to make educated guesses about potential defects in regions where formal testing isn’t feasible.

### ****36. What is a top-down and bottom-up approach in testing?****

Top-down – In top-down Testing is done from the top to the bottom. To put it another way, high-level modules are evaluated first, followed by low-level modules. Finally, the low-level modules are merged into a high-level state to ensure that the framework performs as planned.

Advantages:

– Helpful if severe faults appear near the end of the programme.

– Once the I/O functions have been added, it is much easier to express test scenarios.

– The Early Skeletal Program promotes morale by allowing demonstrations.

Drawbacks:

– Stub modules must be created

– Stub Modules are frequently more difficult than they appear.

– It can be difficult to describe test scenarios in stubs before the I/O routines are added.

– Creating test conditions may be impossible or extremely complex.

– Examining test results is more challenging.

– Encourages the idea that design and testing can be combined.

– Induces one to postpone the completion of particular module testing.

Bottom-up — In Bottom-up Testing takes place at all levels, from the ground up. First, the lowest-level modules are examined, followed by high-level state modules. Finally, the high-level state modules are coordinated at a low level to ensure that the framework fills in as expected.

Advantages:

– Useful if severe faults appear near the end of the programme.

– Creating test conditions is less difficult.

– It is simpler to observe test outcomes.

Drawbacks:

– Driver Modules need to be created.

– The programme does not exist as a whole until the final module is added.

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

|  |  |  |
| --- | --- | --- |
| **Features** | **Smoke Testing** | **Sanity Testing** |
| **System Builds** | Initial builds of software products are subjected to testing. | Builds that have passed smoke tests and regression tests are subjected to testing. |
| **Motive of Testing** | To test the new build’s stability before putting it through more rigorous testing | To assess the rationale and originality of software builds’ functionalities |
| **Subset of?** | Is a type of acceptability testing that is a subset of acceptance testing? | Is a type of regression testing that is a subset of regression testing? |
| **Documentation** | Work includes documentation and programming. | Doesn’t put a lot of emphasis on documentation. |
| **Test Coverage** | To integrate all of the essential capabilities without diving too deep, we took a shallow and broad approach. | An strategy that is both narrow and deep, comprising extensive testing of functionalities and features. |
| **Performed By?** | Developers or testers are in charge of the execution. | Testers are in charge of the execution. |

**Key points about smoke testing and sanity testing:**

Smoke testing is carried out at the start of the project to check for the most basic functions. Sanity testing, on the other hand, thoroughly checks software builds.

Smoke testing requires a written set of tests or automated tests for documentation, whereas sanity testing does not.

The smoke testing method is shallow and broad, which means it tests every build but does not go to extremes. Sanity testing, on the other hand, employs a focused and deep method in which a single build is extensively tested.

The smoke testing method is shallow and broad, which means it tests every build but does not go to extremes. Sanity testing, on the other hand, employs a focused and deep method in which a single build is extensively tested.

The fundamental goal of smoke testing is to swiftly cover every aspect of the software. Sanity testing, on the other hand, focuses on the functionality of each software module.

The developer is in charge of the smoke test, whereas the tester is in charge of the sanity test.

In a document verification procedure, smoke testing is similar to counting the number of documents. Sanity testing, on the other hand, entails a thorough examination of a single document.

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

|  |  |
| --- | --- |
| **Static testing** | **Dynamic testing** |
| Static Testing is a white box testing technique that entails browsing records in order to identify flaws in the very early stages of the SDLC. | Dynamic testing is performed at the end of the software development lifecycle and involves the execution of code. It verifies and approves the output to ensure that the desired outcomes are achieved. |
| Static Testing is used during the verification process. | During the validation step, dynamic testing begins. |
| Before deploying the code, static testing is carried out. | After the code has been deployed, dynamic testing is carried out. |
| In this form of testing, code error detection and programme execution are not a problem. | For dynamic testing, code execution is required. |

### ****39. How will you determine when to stop testing?****

It can be tough to know when to stop testing. Many modern software applications are so complicated and run in such an interconnected environment that thorough testing is impossible. The following are some frequent criteria to consider when considering when to end testing:

* Deadlines are important (release deadlines, testing deadlines, etc.)
* Completed test cases with a particular percentage of them passing
* When the test budget runs out
* When the coverage of code, functionality, or requirements reaches a certain point, it is said to be complete.
* When the bug rate falls below a specific threshold,
* When the beta or alpha testing phase is over

To determine when to end testing, use the following scientific methods:

**1) Decision based on the number of test cases that pass or fail:**

a) Before the test execution cycle, prepare a predetermined number of test cases.

b) All test cases must be completed. During each testing cycle.

c) When all test cases are passed, the testing procedure is terminated.

d) Alternatively, if the proportion of failure in the previous testing cycle is exceedingly low, testing can be terminated.

**2) Metric-driven decision**:

a) Mean Time Between Failure (MTBF): this is calculated by calculating the average operational time before the system fails.

b) Coverage metrics: by keeping track of how many instructions were executed during tests.

c) Defect density: measuring the number of defects per 1000 lines of code or the number of open bugs, as well as their severity levels.

### ****40. What if the software is so buggy it can’t really be tested at all?****

Frequently, testers come across a bug that cannot be fixed. 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 a concentration on critical bugs. Because this type of issue 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.

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

It’s possible that a requirement stack for a product isn’t available. It could take a lot of work to figure out if an app has a lot of unexpected functionality, and it could suggest deeper issues with the software development process. If a feature isn’t required for the application’s goal, it should be eliminated. Create a test plan based on the assumptions you’ve made about the product if all else fails. However, make sure that all of your assumptions are adequately documented in the test plan.

### ****42. What if an organization is growing so fast that fixed testing processes are impossible? What to do in such situations?****

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:

• People who are good at what they do should be hired.

• Quality issues should be ‘ruthlessly prioritised’ by management, with a constant focus on the client.

• Everyone in the company should understand what ‘quality’ means to the customer.

### ****43. How do you know the code has met specifications?****

Code that works, is bug-free, and is understandable and manageable is considered “good code.” Most organisations 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 restrictions 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.

### ****44. What are the cases when you’ll consider choosing automated testing over manual testing?****

When the following scenarios apply, automated testing should be preferred over manual testing:

* When testing must be run on a regular basis
* Repetitive steps are included in the tests.
* When you have a limited amount of time to complete the testing phase, tests must be conducted in a regular runtime environment.
* When a large amount of code needs to be tested repeatedly
* Every execution necessitates a report.

### ****45. What is ‘configuration management’?****

Every well-functioning company has a “master plan” that outlines how it will operate and complete tasks. It’s the same with software development and testing. SCM is a collection of processes, policies, and tools for organising, controlling, coordinating, and tracking:

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

### ****46. Is it true that we can do system testing at any stage?****

In system testing, all of the software’s components are tested as a whole to guarantee that the final product fits the required requirements. As a result, no. Only until all of the units are in place and performing properly can the system testing begin. Prior to the UAT, system testing is frequently performed (User Acceptance Testing).

### ****47. What are some best practices that you should follow when writing test cases?****

* The following are some guidelines to follow when writing test cases:
* Prioritize which test cases to build based on your application’s risk considerations and project timeframes.
* Keep the 80/20 rule in mind. To get the best coverage, you should have 20% of your tests cover 80% of your application.

Instead of trying to test all of your instances at once, improvise as you go.

* Make a list of all of your test cases and categorise them according to business scenarios and functionality.
* Ascertain that test cases are modular and that test case steps are as detailed as possible.
* Write test cases in a way that others can easily understand and alter them if necessary.
* Always keep end-user requirements in mind because, at the end of the day, the product is developed for the client.
* To maintain a solid release cycle, actively use a test management solution.
* Keep an eye on your test cases on a frequent basis. Remove irrelevant and duplicate test cases and write unique test cases.

### ****48. Why is it that the boundary value analysis provides good test cases?****

The fact that a greater number of errors occur at the boundaries rather than in the middle of the input domain for a test is one of the reasons why boundary value analysis produces effective test cases.

Test cases for the boundary value analysis technique are designed to include values at the edges. It’s called ‘Positive testing’ if the input is within the boundary value. Negative testing happens when the input value is outside of the boundary value. Maximum, minimum, inside or outside edge, normal values, or error values are all included.

Assume you’re looking for a number input box that takes numbers ranging from ’01 to 10′.

We may divide test scenarios into three categories using boundary value analysis:

Test scenarios with test data that is identical to the input boundaries: 1 and ten (in this case)

Values just below the input domains’ extreme edges: 0 and 9 are the two most common numbers. Test data with values right above the input domains’ extreme edges: 11 and 2

As a result, the boundary values would be 0, 1, 2 and 9, 10, 11 respectively.

### ****49. Why is it impossible to test a program thoroughly or in other terms 100% bug-free?****

It is impossible to create a software product that is bug-free 100 percent of the time. You can simply minimise a computer program’s or system’s error, flaw, failure, or fault that causes it to generate an inaccurate or unexpected outcome.

Here are the two main reasons why testing a programme completely is impossible.

Software specifications are usually subjective, leading to a range of interpretations.

To test a software programme, it may be necessary to use too many inputs, outputs, and path combinations.

### ****50. Can automation testing replace manual testing?****

Manual testing is not replaced by automation testing. You can’t automate everything, no matter how good your automated tests are. Manual tests are useful in software development because they can be used in situations where automation isn’t possible.

Both automated and manual testing have advantages and disadvantages. Manual testing allows us to have a better understanding of the problem and explore different test angles with greater freedom.

Automated testing saves time in the long run by conducting a high number of surface-level tests in a short period of time.

Now, let’s have a quick discussion about manual testing in general.

The following are some of the advantages of manual testing:

* For products with a short life cycle, this is the best option.
* It helps you save time, money, and resources.
* Ascertain that the product is free of errors.
* Ad hoc testing, exploratory testing, and usability testing are all possible with this tool.
* To make simple adjustments, there’s no need to alter the complete code.
* Obtain precise feedback on the user interface
* Ability to better handle complex use case conditions
* GUI testing can be done with precision.
* Exceptionally dependable
* Improve user friendliness
* New testers will find it simple to pick up

The following are some of the disadvantages of manual testing:

* Not recommended for time-sensitive projects or large enterprises.
* Human errors and blunders are more likely to occur.
* Less efficient because there is no option to document the testing procedure.
* Less Dependable
* Testing for regression takes a long time.
* Doesn’t cover everything there is to know about testing.
* Manual load testing and performance testing are possible.
* In the long term, the process is more costly.

Process of manual testing

* Planning and control
* Analysis and Design
* Implementation and Execution
* Evaluating exit criteria and Reporting
* Test closure activities

# WisdomJobs: 40Q

1. **Question 1. What Is Baseline Testing?**

**Answer :**

Baseline testing is the process of running a set of tests to capture performance information. Baseline testing use the information collected to made the changes in the application to improve performance and capabilities of the application. Baseline compares present performance of application with its own previous performance.

1. **Question 2. What Is Benchmark Testing?**

**Answer :**

Benchmarking testing is the process of comparing application performance with respect to industry standard which is given by some other organization. Benchmark informs us where our application stands with respect to others. Benchmark compares our application performance with other company’s application’s performance.

[Testing Tools Interview Questions](https://www.wisdomjobs.com/e-university/testing-tools-interview-questions.html)

1. **Question 3. What Is Verification And Validation?**

**Answer :**

Verification: process of evaluating work-products of a development phase to determine whether they meet the specified requirements for that phase.

Validation: process of evaluating software during or at the end of the development process to determine whether it specified requirements.

Difference between Verification and Validation:  
•Verification is Static Testing where as Validations is Dynamic Testing.  
•Verification takes place before validation.  
•Verification evaluates plans, document, requirements and specification, where as Validation evaluates product.  
•Verification inputs are checklist, issues list, walkthroughs and inspection ,where as in Validation testing of actual product.  
•Verification output is set of document, plans, specification and requirement documents where as in Validation actual product is output.

1. **Question 4. Explain Branch Coverage And Decision Coverage.**

**Answer :**

•Branch Coverage is testing performed in order to ensure that every branch of the software is executed atleast. To perform the Branch coverage testing we take the help of the Control Flow Graph.   
•Decision coverage testing ensures that every decision taking statement is executed atleast once.  
•Both decision and branch coverage testing is done to ensure the tester that no branch and decision taking statement, will not lead to failure of the software.  
•To Calculate Branch Coverage:

Branch Coverage = Tested Decision Outcomes / Total Decision Outcomes.

[Testing Tools Tutorial](https://www.wisdomjobs.com/e-university/testing-tools-tutorial-239.html" \o "Testing Tools Tutorial)

1. **Question 5. What Is Difference Between Retesting And Regression Testing?**

**Answer :**

•Retesting is done to verify defect fix previous in now working correctly where as regression is perform to check if the defect fix have not impacted other functionality that was working fine before doing changes in the code.  
•Retesting is specific and is performed on the bug which is fixed where as in regression is not be always specific to any defect fix it is performed when any bug is fixed.  
•Retesting concern with executing those test cases that are failed earlier where as regression concern with executing test cases that was passed in earlier builds.  
•Retesting has higher priority over regression.

[QTP Interview Questions](https://www.wisdomjobs.com/e-university/qtp-interview-questions.html" \o "QTP Interview Questions)

1. **Question 6. What Is Mutation Testing & When Can It Be Done?**

**Answer :**

Mutation testing is a performed to find out the defect in the program. It is performed to find put bugs in specific module or component of the application. Mutation testing is based on two assumptions:

Competent programmer hypothesis: according this hypothesis we suppose that program write the correct code of the program.

Coupling effect: according to this effect collection of different set of test data can also find large and complex bugs.

1. **Question 7. Explain Bug Leakage And Bug Release.**

**Answer :**

**Bug Leakage:** When customer or end user discovered a bug which can be detected by the testing team. Or when a bug is detected which can be detected in pervious build then this is called as Bug Leakage.

Bug release: is when a build is handed to testing team with knowing that defect is present in the release. The priority and severity of bug is low. It is done when customer want the application on the time. Customer can tolerate the bug in the released then the delay in getting the application and the cost involved in removing that bug. These bugs are mentioned in the Release Notes handed to client for the future improvement chances.

[QTP Tutorial](https://www.wisdomjobs.com/e-university/qtp-tutorial-260.html" \o "QTP Tutorial) [Database Testing Interview Questions](https://www.wisdomjobs.com/e-university/database-testing-interview-questions.html" \o "Database Testing Interview Questions)

1. **Question 8. What Is Alpha And Beta Testing?**

**Answer :**

Alpha testing: is performed by the IN-House developers. After alpha testing the software is handed over to software QA team, for additional testing in an environment that is similar to the client environment.

Beta testing: beta testing becomes active. It is performed by end user. So that they can make sure that the product is bug free or working as per the requirement. IN-house developers and software QA team perform alpha testing. The public, a few select prospective customers or the general public performs beta testing.

1. **Question 9. What Is Monkey Testing?**

**Answer :**

Monkey testing is a type of Black Box Testing used mostly at the Unit Level. In this tester enter the data in any format and check the software is not crashing. In this testing we use Smart monkey and Dumb monkey.

Smart monkeys are used for load and stress testing, they will help in finding the bugs. They are very expensive to develop. Dumb monkey, are important for basic testing. They help in finding those bugs which are having high severity. Dumb monkey are less expensive as compare to Smart monkeys.

Example: In phone number filed Symbols are entered.

[Automation Testing Interview Questions](https://www.wisdomjobs.com/e-university/automation-testing-interview-questions.html" \o "Automation Testing Interview Questions)

1. **Question 10. What Is Test Driver And Test Stub?**

**Answer :**

•The Stub is called from the software component to be tested. It is used in top down approach.  
•The driver calls a component to be tested. It is used in bottom up approach.  
•Both test stub and test driver are dummy software components.

We need test stub and test driver because of following reason:  
•Suppose we want to test the interface between modules A and B and we have developed only module A. So we cannot test module A but if a dummy module is prepare, using that we can test module A.  
•Now module B cannot send or receive data from module A directly so, in these cases we have to transfer data from one module to another module by some external features. This external feature used is called Driver.

[Database Testing Tutorial](https://www.wisdomjobs.com/e-university/database-testing-tutorial-966.html" \o "Database Testing Tutorial)

1. **Question 11. What Is Random Testing?**

**Answer :**

When tester performs testing of application by using random input from the input domain of the system, this is Random Testing.

Random testing involve following procedures:  
•Selection of input domain.  
•Randomly selecting any input from input domain.  
•Using these test input testing of application is performed.  
•The results are compared to the system specification. The test is a failure if any input leads to incorrect results, otherwise it is a success.

[Software testing Interview Questions](https://www.wisdomjobs.com/e-university/software-testing-interview-questions.html" \o "Software testing Interview Questions)

1. **Question 12. What Is Agile Testing?**

**Answer :**

Agile Testing means to quickly validation of the client requirements and make the application of good quality user interface. When the build is released to the testing team, testing of the application is started to find the bugs. As a Tester, we need to focus on the customer or end user requirements. We put the efforts to deliver the quality product in spite of short time frame which will further help in reducing the cost of development and test feedbacks will be implemented in the code which will avoid the defects coming from the end user.

[Testing Tools Interview Questions](https://www.wisdomjobs.com/e-university/testing-tools-practice-tests-239-327206" \o "Testing Tools Interview Questions)

1. **Question 13. What Is The Purpose Of Test Strategy?**

**Answer :**

We need Test Strategy for the following reasons:  
1. To have a signed, sealed, and delivered document, where the document contains details about the testing methodology, test plan, and test cases.  
2. Test strategy document tells us how the software product will be tested.  
3. Test strategy document helps to review the test plan with the project team members.  
4. It describes the roles, responsibilities and the resources required for the test and schedule.  
5. When we create a test strategy document, we have to put into writing any testing issues requiring resolution.

The test strategy is decided first, before lower level decisions are made on the test plan, test design, and other testing issues.

[Software testing Tutorial](https://www.wisdomjobs.com/e-university/software-testing-tutorial-1303.html" \o "Software testing Tutorial)

1. **Question 14. Explain Bug Life Cycle.**

**Answer :**

•The bug is assigned to development project manager who will analyze the bug .He will check whether it is a valid defect. If not valid bug is rejected then status is REJECTED.  
•If not, next the defect is checked whether it is in scope. When bug is not part of the current release .Such defects are POSTPONED  
•Now, Tester checks whether a similar defect was raised earlier. If yes defect is assigned a status DUPLICATE  
•When bug is assigned to developer. During this stage bug is assigned a status IN-PROGRESS  
•Once code is fixed. Defect is assigned a status FIXED  
•Next the tester will re-test the code. In case the test case passes the defect is CLOSED  
•If the test case fails again the bug is RE-OPENED and assigned to the developer. That’s all to Bug Life Cycle.

1. **Question 15. What Is Error Guessing And Error Seeding?**

**Answer :**

Error Guessing is a test case design technique where the tester has to guess what faults might occur and to design the tests to represent them.

Error Seeding is the process of adding known faults intentionally in a program for the reason of monitoring the rate of detection & removal and also to estimate the number of faults remaining in the program.

[Test Cases Interview Questions](https://www.wisdomjobs.com/e-university/test-cases-interview-questions.html" \o "Test Cases Interview Questions)

1. **Question 16. Explain Compatibility Testing With An Example.**

**Answer :**

Compatibility testing is to evaluate the application compatibility with the computing environment like Operating System, Database, Browser compatibility, backwards compatibility, computing capacity of the Hardware Platform and compatibility of the Peripherals. Example, If Compatibility testing is done on a Game application, before installing a game on a computer, its compatibility is checked with the computer specification that whether it is compatible with the computer having that much of specification or not.

1. **Question 17. What Is Test Harness?**

**Answer :**

A test harness is a collection of software and test data required to test the application by running it in different testing condition like stress, load, data- driven, and monitoring its behavior and outputs. Test Harness contains two main parts:  
•Test execution engine  
•Test script repository

Automation testing is the use of a tool to control the execution of tests and compare the actual results with the expected results. It also involves the setting up of test pre-conditions.

[Web testing Interview Questions](https://www.wisdomjobs.com/e-university/web-testing-interview-questions.html" \o "Web testing Interview Questions)

1. **Question 18. Explain Statement Coverage.**

**Answer :**

Statement Coverage is a metric used in White Box Testing. Statement coverage is used to ensure that all the statement in the program code is executed at least once. The advantages of Statement Coverage are:  
•Verifies that written code is correct.  
•Measures the quality of code written.  
•Determine the control flow of the program.  
•To Calculate Statement Coverage:  
•Statement Coverage = Statements Tested / Total No. of Statements.

[QTP Interview Questions](https://www.wisdomjobs.com/e-university/qtp-interview-questions.html" \o "QTP Interview Questions)

1. **Question 19. What Are The Types Of Testing?**

**Answer :**

There are two types of testing:  
•Static testing: Static testing is a technique used in the earlier phase of the development life cycle. The code error detection and execution of program is not concern in this type of testing. Also known as non-execution technique. The Verification of the product is performed in this testing technique like Code Reviews, Inspections, Walkthroughs are mostly done in this stage of testing.  
•Dynamic testing: Dynamic Testing is concern with the execution of the software. This technique is used to test the dynamic behavior of the code. Most of the bugs are identified using this technique. These are the Validation activities. It uses different methodologies to perform testing like Unit Tests, Integration Tests, System Tests and Acceptance Testing, etc.

1. **Question 20. Explain User Acceptance Testing.**

**Answer :**

User Acceptance Testing (UAT) is performed by the end users on the applications before accepting the application.

Alpha testing: is performed by the IN-House developers. After alpha testing the software is handed for the Beta testing phase, for additional testing in an environment that is similar to the client environment.

Beta testing: is performed by the end user. So that they can make sure that the product is bug free or working as per the requirement. IN-house developers and software QA team perform alpha testing. The public, a few select prospective customers or the general public performs beta testing.

Gamma Testing: Gamma Testing is done when the software is ready for release with specified requirements. This testing is done directly by skipping all the in-house testing activities.

[Rational Functional Tester Interview Questions](https://www.wisdomjobs.com/e-university/rational-functional-tester-interview-questions.html" \o "Rational Functional Tester Interview Questions)

1. **Question 21. What Should Be Done After A Bug Is Found?**

**Answer :**

After finding the bug the first step is bug to be locked in bug report. Then this bug needs to be communicated and assigned to developers that can fix it. After the bug is fixes by the developer, fixes should be re-tested, and determinations made regarding requirements for regression testing to check that fixes didn't create problems elsewhere.

1. **Question 22. What If The Software Is So Buggy It Can't Really Be Tested At All?**

**Answer :**

In this situation is for the testers to go through the process of reporting of bugs with the focus being on critical bugs. Since this type of problem can severely affect schedules, and indicates deeper problems in the software development process project managers should be notified, and provided with some documentation.

1. **Question 23. What Are The Types Of Maintenance?**

**Answer :**

There are four types of maintenance. They are:  
•Corrective Maintenance  
•Adaptive Maintenance  
•Perfective Maintenance  
•Preventive Maintenance

[Test Estimation Interview Questions](https://www.wisdomjobs.com/e-university/test-estimation-interview-questions.html" \o "Test Estimation Interview Questions)

1. **Question 24. What Are The Advantages Of Waterfall Model?**

**Answer :**

The advantages of the waterfall model are:  
•Simple to implement and required fewer amounts of resources.  
•After every phase output is generate.  
•Help in methods of analysis, design, coding, testing and maintenance.  
•Preferred in projects where quality is more important than schedule and cost.  
•Systematic and sequential model.  
•Proper documentation of the project.

[Database Testing Interview Questions](https://www.wisdomjobs.com/e-university/database-testing-interview-questions.html" \o "Database Testing Interview Questions)

1. **Question 25. What Is Rapid Application Development Model (rad)?**

**Answer :**

The RAD model Rapid Application development (RAD) is incremental software development process models that focus on the development of the project in very short time. It is enhanced version of Waterfall model. It is proposed when requirements and solutions can be made independently system or software components, which is developed by different teams. After these smaller system components are developed, they are integrated to produce the large software system solution.

1. **Question 26. What Are The Advantages Of Black Box Testing?**

**Answer :**

The advantages of this type of testing include:  
•Developer and tester are independent of each other.  
•The tester does not need knowledge of any programming languages.  
•The test is done from the point-of-view of the user.  
•Test cases can be designed when specifications are complete.  
•Testing helps to identify issues related to functional specifications.

[Mobile Application Testing Interview Questions](https://www.wisdomjobs.com/e-university/mobile-application-testing-interview-questions.html" \o "Mobile Application Testing Interview Questions)

1. **Question 27. What Is Software Review?**

**Answer :**

A software review can be defined as a filter for the software engineering process. The purpose of any review is to discover errors in the analysis, design, and coding, testing and implementation phases of the software development cycle. The other purpose of a review is to see whether procedures are applied uniformly and in a manageable manner. It is used to check the process followed to develop the software is right.

[Automation Testing Interview Questions](https://www.wisdomjobs.com/e-university/automation-testing-interview-questions.html" \o "Automation Testing Interview Questions)

1. **Question 28. What Is Reverse Engineering?**

**Answer :**

By analyzing a final product the process of recreating a design is known as reverse engineering. Reverse engineering is the process followed in order to find difficult, unknown, and hidden information about a software system. It is important when software products lack proper documentation, and are highly unstructured, or their structure has degraded through a series of maintenance efforts. Maintenance activities cannot be performed without a complete understanding of the software system.

1. **Question 29. What Is Data Flow Diagram?**

**Answer :**

The Data Flow Diagram gives us information of the flow of data within the application.  
•The DFD can be used to analyze the design of the application.  
•It is a graphical representation of the structure of the data.  
•A developer draws context level DFD first showing interaction between the different components of the application.  
•DFD help in developing the software by clarifying the requirements and major functionalities.  
•DFDs show the flow of data through a system.  
•It is an important modeling tool that allows us to picture a system as a network of functional processes.

[Test Manager Interview Questions](https://www.wisdomjobs.com/e-university/test-manager-interview-questions.html" \o "Test Manager Interview Questions)

1. **Question 30. What Is Exploratory Testing?**

**Answer :**

Exploratory testing: means testing an application without a test plan and test script. In exploring testing test explore the application on the basis on his knowledge. The tester has no knowledge about the application previously. He explores the application like an end user and try to use it. While using the application his main motive is to find the bugs which are in the application.

1. **Question 31. What Is Compatibility Testing?**

**Answer :**

Compatibility testing is a type of testing used to find out the compatibility between the application and platform on which application works, web browsers, hardware, operating systems etc. Good software must be compatible with different hardware, web browser and database.

1. **Question 32. What Is Srs And Brs Document?**

**Answer :**

Software Requirements Specification (SRS) is documented form of the requirement of the customer. It consists of all requirement of the customer regarding that software to be developed. The SRS document work as agreement between the company and the customer consisting of all functional and non functional requirements.

Business Requirement Specification (BRS) are the requirements as described by the business people. The business tells “what” they want for the application to do. In simple word BRS contain the functional requirement of the application.

[Test Director Interview Questions](https://www.wisdomjobs.com/e-university/test-director-interview-questions.html" \o "Test Director Interview Questions)

1. **Question 33. Can You Explain V Model In Manual Testing?**

**Answer :**

**V model:** it is enhanced version of waterfall model where each level of the development lifecycle is verified before moving to next level. In this testing starts at the very beginning. By testing we mean verification by means of reviews and inspections, static testing. Each level of the development life - cycle has a corresponding test plan. A test plan is developed to prepare for the testing of the products of that phase. Be developing the test plans, we can also define the expected results for testing of the products for that level as well as defining the entry and exit criteria for each level.

[Software testing Interview Questions](https://www.wisdomjobs.com/e-university/software-testing-interview-questions.html" \o "Software testing Interview Questions)

1. **Question 34. What Is Concurrency Testing?**

**Answer :**

Concurrency Testing is used to know the effects of using the software by different users at the same time. In this type of testing we have multiple users performing the exact same requests at the same time. It helps in identifying and measuring the problems in Response time, levels of locking and deadlocking in the application. For this we use Load runner to create VUGen (Virtual User Generator) is used to add the number of concurrent users and perform operation on the application on the same time.

1. **Question 35. What Is An Inspection In Software Testing?**

**Answer :**

An inspection is more formalized than a walkt hrough. Inspection technique involves 3 to 8 team member consisting of a moderator, reader, and a recorder to take notes. The subject of the inspection is typically a document such as a requirements or a test plan, and the purpose is to find problems and see what is missing, most problems will be found during this preparation. The result of the inspection meeting should be a written report. It is one of the most cost effective methods of ensuring quality.

1. **Question 36. A Form Has Four Mandatory Fields To Be Entered Before You Submit. How Many Numbers Of Test Cases Are Required To Verify This? And What Are They?**

**Answer :**

Five test cases are required to test:  
1. Enter the data in all the mandatory fields and submit, should not display error message.  
2. Enter data in any two mandatory fields and summit, should issue an error message.  
3. Do not enter in any of the fields should issue an error message.  
4. If the fields accept only number, enter numbers in the fields and submit, should not issue an error message, try to enter only in two fields should issue an error message, and enter alphabets in two fields and number in other two fields it should issue an error message.  
5. If the fields do not accept special characters, then enter the characters and submit it.

[Test Cases Interview Questions](https://www.wisdomjobs.com/e-university/test-cases-interview-questions.html" \o "Test Cases Interview Questions)

1. **Question 37. What Is Cyclomatic Complexity?**

**Answer :**

Cyclomatic complexity is used to measure the complexity of the software using the control flow graph of the software. It is a graphical representation, consisting of following:

NODE: statement of the program is taken as node of the graph.

Edges: the flow of command is denoted by edges. Edges are used to connect two node , this show flow of control from one node to other node in the program.

Using this node and edges we calculate the complexity of the program. This determines the minimum number of inputs you need to test always to execute the program.

1. **Question 38. What Is The Key Difference Between Preventative And Reactive Approaches To Testing?**

**Answer :**

Preventative tests are designed early; reactive tests are designed after the software has been produced.

1. **Question 39. What Is The Purpose Of Exit Criteria?**

**Answer :**

The purpose of exit criteria is to define when a test level is completed.

1. **Question 40. When Is Used Decision Table Testing?**

**Answer :**

Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.

# EngineeringInterviewQs: 130Q