1. **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.

#### ****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.,

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

Dynamic testing involves the execution of code. It validates the output with the expected outcome.

#### ****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.

Various white-box testing techniques are:

1. Statement Coverage
2. Decision Coverage
3. Condition Coverage
4. Multiple Condition Coverage

#### ****5.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.

#### ****6.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.

#### ****7.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.

#### ****8.What is Test Suite?****

Test Suite is a collection of test cases. The test cases which are intended to test an application.

#### ****9.What is Test Scenario?****

Test Scenario gives the idea of what we have to test. Test Scenario is like a high-level test case.

#### ****10.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.

#### ****11. What is Test Bed?****

An environment configured for testing. Test bed consists of hardware, software, network configuration, an application under test, other related software.

**12. 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.

#### ****13. 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.

#### ****14. 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.

**15. 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)

#### ****16. What is a Test Report?****

A test report is a document that provides an overview of testing objectives, activities, and results. It is necessary to summarize testing results and compare them against expectations. It helps us determine if the product is ready for release or not. Additionally, it allows us to see the current status of the project and assess the quality of the product

**17. 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

**18. Write some common mistakes that lead to major issues.**

Some of the most common mistakes include:

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

#### ****19. 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

#### ****20. 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.

#### ****21. 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.

#### ****22. 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.

#### ****23. 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.

#### ****24.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.

#### ****25. In manual testing what are stubs and drivers?****

In software testing, Stubs and drivers are used in manual testing to test the functionality of a system without having to use the actual system.

A stub is a small piece of code that is called by the Module under Test. A driver is a small piece of code that calls the Module to be tested.

#### ****26. 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.

#### ****27. 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. |

#### ****28. What is End-To-End Testing?****

In simple words, end-to-end testing is the process of testing software from start to end.

#### ****29. 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.

#### ****30. 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.

#### ****31. 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 |

#### ****32. 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.

**32.On what basis is the acceptance test 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.

#### ****33. 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.

#### ****34. 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.

#### ****35. 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.

#### ****36. 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.

#### ****37. 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.

#### ****38. 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. |

#### ****39. 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.

**40. 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).

#### ****41. 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.

#### ****42. What is GUI Testing?****

Graphical User Interface Testing is to test the interface between the application and the end user.

#### ****43. 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.

#### ****44. What is Globalization Testing/Internationalization Testing (I18N Testing)****?

Globalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.

#### ****45. 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.

#### ****46. What is Installation Testing?****

It is to check whether the application is successfully installed and it is working as expected after installation.

#### ****47. What is Formal Testing?****

It is a process where the testers test the application by having pre-planned procedures and proper documentation.

#### ****48. What is Risk Based Testing?****

Identify the modules or functionalities which are most likely cause failures and then testing those functionalities.

#### ****49. What is Compatibility Testing?****

It is to deploy and check whether the application is working as expected in a different combination of environmental components.

#### ****50. 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.

#### ****51. What is Monkey Testing?****

Perform abnormal action on the application deliberately in order to verify the stability of the application.

#### ****52. 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.

#### ****53. What is Security Testing?****

Security testing is a process to determine whether the system protects data and maintains functionality as intended.

#### ****54. 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.

#### ****55. What is Endurance Testing?****

Endurance testing is a non-functional testing type. It is also known as Soak Testing. Refer Soak testing.

#### ****56. 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.

#### ****57. 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.

#### ****58. What is Volume Testing?****

It is toverify that the system/application can handle a large amount of data

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

It is to verify the behavior of the system once the load increases more than its design expectations.

#### ****60. 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.

#### ****61. 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.

#### ****62. 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.

#### ****63. 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.

#### ****64. What is Cross-Browser Testing(browser compatibility)?****

Cross Browser Testing is a type of non-functional test which helps us ensure that our website or web application works well in various browsers.

Web applications rely on browsers like Google Chrome, Mozilla Firefox, Internet Explorer and Safari to function. Even though they all support web standards to some extent, there are still slight differences between them. This can pose a problem for developers who have to test their software on multiple browsers and take note of any inconsistencies.

Different browsers display websites differently due to styling, and it’s not possible to have every browser installed on one machine. Each browser is designed by a different company. Every browser has its own individualized features to make it stand out from the rest. When testing a website, we have to confirm that our site looks the same on all browsers.

Browser automation with automation tools like Selenium, Katalon, Cypress, etc., is significantly quicker than testing it manually because these tools can test many scenarios in a very short amount of time.

#### ****65. What is meant by browser automation?****

Browser automation is the process of testing a web application’s functionality in a browser automatically, using a script. The script launches the browser, navigates to the site, and then interacts with it like an normal end user would. This involves clicking buttons or links.

#### ****66. What is Interface Testing?****

Interface testing is performed to evaluate whether two intended modules pass data and communicate correctly to one another.

#### ****67. What is Reliability Testing?****

Perform testing on the application continuously for long period of time in order to verify the stability of the application

#### ****68. 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.

#### ****69. What is A/B Testing / What is Split Testing?****

A/B testing is the process of comparing two versions of a webpage or other marketing asset to see which one performs better. The comparison doesn’t stop with web pages – it can be app interfaces, landing pages, email marketing, etc.

**70. What are the principles of Software Testing?**

1. **Testing shows presence of defects:** Although testing can discover software defects, it cannot ensure that the product is free of errors. Testing may reduce the quantity of glitches, but it will never be able to eliminate them all.
2. **Exhaustive testing is impossible:**Since it’s impossible to test the software exhaustively, we can only run a few select tests and we assume that the software will always produce correct output. More comprehensive testing would be too costly and time-consuming.
3. **Early testing:**It’s crucial to test software early on to find defects. In the early stages of SDLC, it’s easier and less expensive to identify defects. Software testing should begin in the first phase of software development, during requirement analysis.
4. **Defect clustering:**According to the Pareto Principle, 80% of software defects arise from 20% of modules. In other words, most project defects are found in only a few sections of code.
5. **Pesticide Paradox:**If you want to find any new bugs, you can’t just keep running the same test cases repeatedly. You need to add or update your existing test cases.
6. **Testing is context depending:**Depending on the software development context, the testing approach will differ. Based upon its type, different software requires variations in testing.
7. **Absence of error fallacy:**Not only does the software need to be bug-free 99% of the time, but it must meet customer requirements if it is ever going to be used.

#### ****71. What is Exhaustive Testing?****

Testing all the functionalities using all valid and invalid inputs and preconditions is known as Exhaustive testing.

#### ****72. 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.

#### ****73. 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.

#### ****74. 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.

#### ****75. 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.,

#### ****76. 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.

#### ****77. What is a Bug?****

If testers find any mismatch in the application/system in testing phase then they call it as Bug.

#### ****78. 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.

#### ****79. 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.

#### ****80. 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.

#### ****81. 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.

#### ****82. 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.

#### ****83. 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.

#### ****84. 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.

**85. 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/)

**86. 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.

#### ****87. 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.

#### ****88. 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.

#### ****89. 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.

#### ****90. 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.)

#### ****91. 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.

#### ****92. 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.

#### ****93. 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).

#### ****94. What’s a bugfix?****

A bugfix is a build aimed at resolving a bug which is detected by the testers in the testing cycle.

#### ****95. 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.

#### ****96. 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.

#### ****97. 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.

#### ****98. 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.

#### ****99. What is an entry criteria?****

The prerequisites that must be achieved before commencing the testing process.

#### ****100. What is an exit criteria?****

The conditions that must be met before testing should be concluded.

**101. 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.

**102. 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/)

#### ****103. 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).

#### ****104. 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.

**105. 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.

**106. 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.

#### ****107. 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.

**108. 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.

#### ****109. 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.