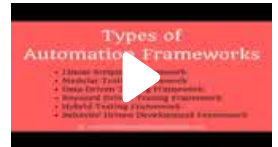


FEATURED VIDEOS

Powered by **[primis]**

Top 10 Times Celebs Mocked Horrible Interview Questions

NOW
PLAYING

- [Why You Choose Software Testing As A Career](#)
- [General Interview Questions](#)
- [SQL Interview Questions for Software QAs](#)
- [Agile Interview Questions](#)
- [Selenium WebDriver Interview Questions](#)
- [Explain Test Automation Framework](#)
- [Test Automation Framework Interview Questions](#)
- [TestNG Interview Questions](#)

Manual Testing QUIZ

www.SoftwareTestingMaterial.com**Take The Quiz**

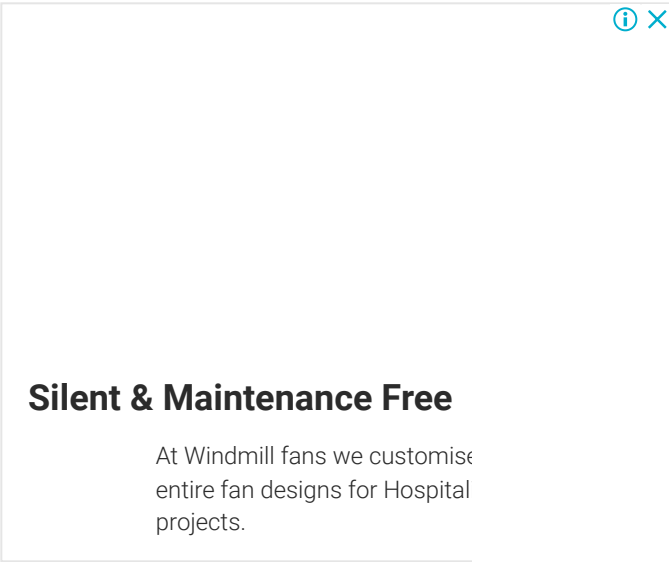
Software Testing Interview Questions – 1-25:

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.](#)

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 Engineer	SDET	Developer
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.

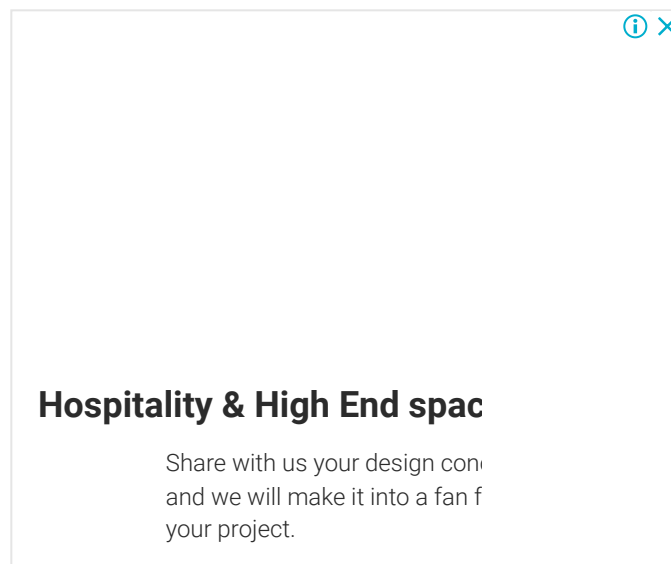
4. How many test cases you can execute in a day?

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](#)

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 developi

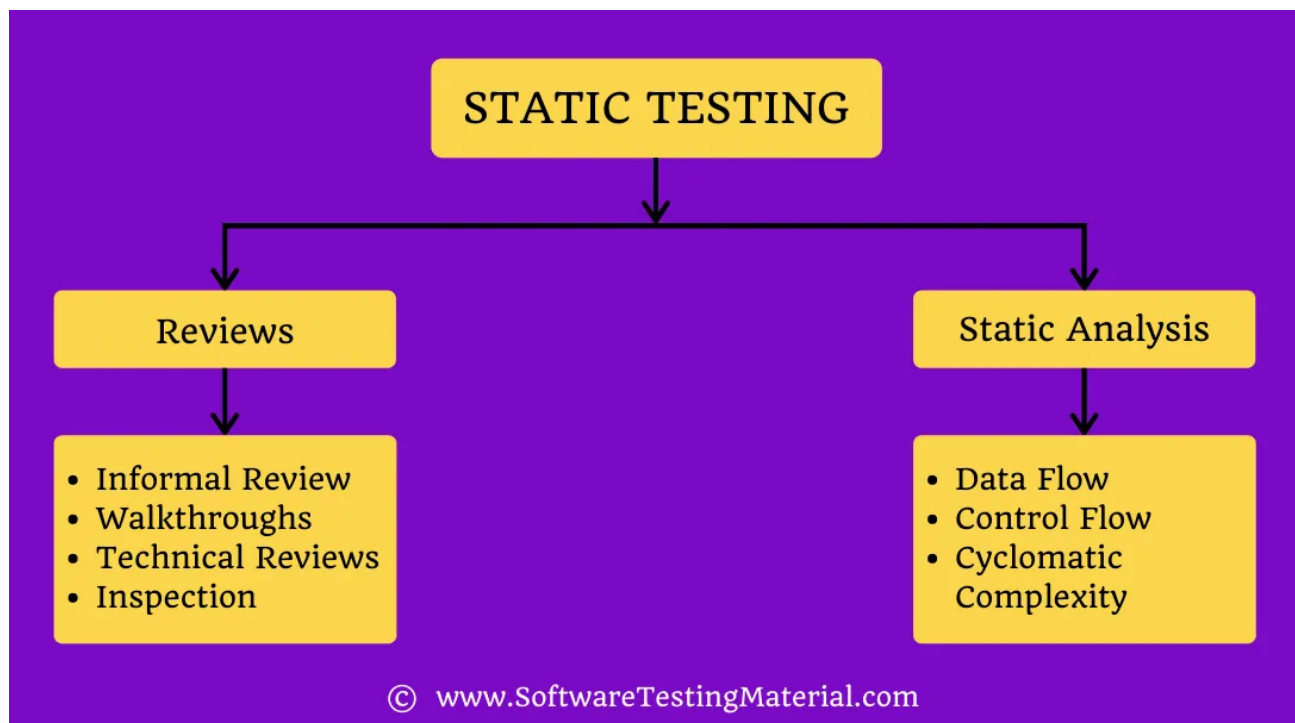
the product accordingly or not. Activities involved here are Inspections, Reviews, Walk-throughs. [Click here for more details.](#)

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.](#)

Don't miss: [Software QA Interview Questions](#)

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](#)

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 Dynamic Testing](#) and [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.](#)

Various white-box testing techniques are:

- i. Statement Coverage
- ii. Decision Coverage
- iii. Condition Coverage
- iv. Multiple Condition Coverage

18. What is Black Box Testing?

Black Box Testing is a [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.](#)

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.](#)

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.](#)

Learn [Difference Between Test Plan vs Test Strategy](#)

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.](#)

Learn [Difference Between Test Case vs Test Scenario](#)

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](#).

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.](#)

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](#) 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](#) in detail.

[Click here for more details.](#)

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](#)

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 reciprocal 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: <ul style="list-style-type: none"> • Top bottom integration testing • Bottom top integration testing • Big bang integration testing • Sandwich integration testing 	Different types of system testing are: <ul style="list-style-type: none"> • 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

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](#) guide for more information. Also, refer [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](#) 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

Functional Testing**Non-functional Testing**

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](#). 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](#) 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](#). 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](#)

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](#).

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](#) and also don't miss these popular [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](#).

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](#).

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](#)

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 to verify 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](#)

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.](#)

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.

Outsource Testing

Crowdsourced testing

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.

Outsource Testing

High paid in comparison to Crowd sourced testing but lesser than Inhouse Testing team.

Crowdsourced testing

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.](#)

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.](#)

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.](#)

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.](#)

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.](#)

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.](#)

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.](#)

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.](#)

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 environment 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.](#)

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.](#)

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.](#)

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.](#)

121. What is an entry criteria?

The prerequisites that must be achieved before commencing the testing process. [Click here for more details.](#)

122. What is an exit criteria?

The conditions that must be met before testing should be concluded. [Click here for more details.](#)

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.](#)

124. What are the different available models of SDLC?

1. [Waterfall](#)
2. [Spiral](#)
3. [V Model](#)
4. Prototype
5. [Agile](#)

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 \(SDLC\)](#) & [Software Testing Life Cycle \(STLC\)](#).

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.](#)

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 at least 1 test case. Requirement Traceability Matrix AKA Traceability Matrix or Cross Reference Matrix. [Click here for more details.](#)

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.](#)

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](#)

132. What is API Testing?

API testing is a type of [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 the main focus will be on a Business logic layer of the [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](#)

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 Question

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](#)
- [General Interview Questions](#)
- [Selenium Interview Questions](#)
- [Explain Test Automation Framework](#)
- [Test Automation Framework Interview Questions](#)
- [TestNG Interview Questions](#)
- [SQL Interview Questions](#)
- [Manual Testing Interview Questions And Answers](#)
- [Agile Interview Questions](#)

If you have any more manual interview questions, feel free to ask via comments. If you find this post useful, do share it with your friends on Social Networking.

Like This Post?

Subscribe and get free access to subscriber-only guides, templates, and checklists.

You are subscribing to email updates. Unsubscribe anytime. Your data is safe.