

TECHNICAL INTERVIEW QUESTIONS FOR SOFTWARE TESTER'S

1. What is Software testing?

Software testing involves evaluating and verifying a software product's functionality. Basically, it checks whether the software product matches anticipated requirements and makes sure it is defect-free. It can be said that testing enhances the quality of the product by preventing bugs, reducing development costs, and reducing performance issues.

2. What are the principles of software testing?

Software testing is governed by seven principles:

- **Absence of errors fallacy:** Even if the software is 99% bug-free, it is unusable if it does not conform to the user's requirements. Software needs to be bug-free 99% of the time, and it must also meet all customer requirements.
- **Testing shows the presence of errors:** Testing can verify the presence of defects in software, but it cannot guarantee that the software is defect-free. Testing can minimize the number of defects, but it can't remove them all.
- **Exhaustive testing is not possible:** The software cannot be tested exhaustively, which means all possible test cases cannot be covered. Testing can only be done with a select few

test cases, and it's assumed that the software will produce the right output in all cases. Taking the software through every test case will cost more, take more effort, etc., which makes it impractical.

- **Defect clustering:** The majority of defects are typically found in a small number of modules in a project. According to the Pareto Principle, 80% of software defects arise from 20% of modules.
- **Pesticide Paradox:** It is impossible to find new bugs by re-running the same test cases over and over again. Thus, updating or adding new test cases is necessary in order to find new bugs.
- **Early testing:** Early testing is crucial to finding the defect in the software. In the early stages of SDLC, defects will be detected more easily and at a lower cost. Software testing should start at the initial phase of software development, which is the requirement analysis phase.
- **Testing is context-dependent:** The testing approach varies depending on the software development context. Software needs to be tested differently depending on its type. For instance, an ed-tech site is tested differently than an Android app.

3. What is a bug in software testing?

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

4. Is Automation testing in agile methodology useful?

It is extremely beneficial to use automation testing when using the agile model in software testing. It helps in achieving maximum test coverage in a lesser time of the sprint.

5.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.

6.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.

7.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

8. 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

9. 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.

10. What is a test case?

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

11. What is a test scenario?

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

12. What is a test plan?

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

13.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

14.What is white box testing and its various techniques?

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

- Statement Coverage
- Decision Coverage

15.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.

16. 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.

17.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.

18. 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.

19.List the four different test levels

The four levels are:

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

- System testing
- Acceptance testing

20. Mention a few advantages of Automated testing.

The following are some major advantages of [automated testing](#) -

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

21. What is Regression Testing?

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

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

The phases involved in the Software Testing Life Cycle are:

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

23.What is the difference between system testing and integration testing?

System testing is a type of software testing that evaluates a complete and fully integrated software product. It verifies that the software meets the requirements specified in the design and the system-level technical specifications. System testing also identifies any weaknesses, errors, or bugs.

Integration testing is software testing that verifies the interactions between two or more system components. It is

performed after unit testing and before system testing. It checks how components interact with each other and how they fit together. Integration testing is necessary to ensure that the components of the system work together as expected.

24.What is the difference between smoke testing and sanity testing?

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

25.How will you determine when to stop testing?

When testing, it is vital to determine when to stop to prevent wasting resources. When deciding when to stop testing, then you should consider the following criteria:

- Desired levels of quality
- Adherence to timelines and budget

- Number of defects found
- Number of test cases that have been completed
- Risk factors associated with the project

26.What are the cases when you'll consider choosing automated testing over manual testing?

The following steps are the considering cases:

- When the test requires repetitive steps:

Automated testing is ideal for running tests requiring multiple iterations or repeating the same actions repeatedly.

- When the test requires a large amount of data:

Automated testing can quickly insert large amounts of data into the tested system.

- When the test requires multiple environments:

Automated testing can easily be configured to test systems in various domains, such as multiple operating systems, browsers, and devices.

- When the test requires precise timing:

Automated tests can be programmed to run precisely, ensuring that each test step is performed at the exact time it needs to be.

- When the test requires multiple users:

Automated testing can simulate multiple users accessing the system simultaneously, allowing for more realistic testing.

27.Is it true that we can do system testing at any stage?

No, system testing is typically carried out at the end of the development process, after integration and user acceptance testing.

28.Why is it impossible to test a program thoroughly or 100% bug-free?

It is impossible to test a program thoroughly or 100% bug-free because it is impossible to anticipate and test every possible combination of inputs, environments, and states a program might encounter.

29.Can automation testing replace manual testing?

No, automation testing cannot fully replace manual testing. Automation testing is designed to supplement manual testing, not replace it. Automation testing can automate repetitive,

tedious test cases and make the testing process more efficient. However, it cannot replace manual testing completely, as some tests can only be performed manually.

30.What is cross-browser testing?

In cross-browser testing, a sort of automated browser testing, the tester determines whether or not the online application will function properly on different browsers, including Chrome, Firefox, Internet Explorer, Safari, and others.

31.What are the different components of Selenium?

Selenium is a framework and a collection of tools which works in conjunction to provide various forms of automation testing. Selenium WebDriver, Selenium Grid, and Selenium IDE are its main parts.

32.What is UI testing?

UI testing aims to ensure that the software provides a consistent user experience and that none of the screen's graphical or visual elements is damaged.

33.What is CAPTCHA?

CAPTCHA is a sort of security that stands for Completely Automated Public Turing Test to Tell Computers and Humans Apart. The CAPTCHA's principal objective is to protect you against spam or denial-of-service attacks by bots/scripts by requiring you to complete a simple test that computers find difficult to understand.

34.What is automation testing?

Automation testing is a software testing strategy in which a tester programmatically runs the tests using a tool or a framework instead of manually going through the test cases and executing them one by one.

35.What's the difference between *driver.close()* and *driver.quit()* commands?

The *driver.close()* command is used to close the current tab/window of the browser which is controlled by the Selenium WebDriver.

The *driver.quit()* command closes all browser windows and ends the Selenium WebDriver session.

36. Explain the differences between *driver.findElement()* and *driver.findElements()* in Selenium WebDriver

driver.findElement () returns the first web element matching the provided locator strategy, while *driver.findElements ()* returns a list of all elements that match the locator.

The former is useful when you expect a single element, like clicking a button, and the latter is used when dealing with multiple elements, like a list of search results.

37.What is TestNG in Selenium?

TestNG is a popular Java framework that permits the user to write automation tests for testing applications and packages. NG stands for Next Generation, and it's commonly used in Selenium to boost the efficiency and organization of test cases.

Candidates should include how the tester can use features like grouping, parallel testing, and reporting with TestNG.

38.What is Selenium Grid and how does it work?

Answer:

Selenium Grid is a component of the Selenium framework that allows you to distribute and run your tests across multiple machines, browsers, and operating systems simultaneously.

It's especially useful for running automated tests in parallel and providing better coverage across different configurations.

39.What is Postman?

Postman is a free, HTTP Client based software application primarily used to perform API testing. It supports testing of HTTP requests by utilizing GUI (Graphical User Interface) which can be executed and the responses can be validated. It also helps in collaborating among the team members for the development of API by providing a platform to design, develop, test, and document APIs.

40. What is a collection in Postman?

A collection in Postman helps to group similar requests. It helps in systematically arranging the requests into folders.

41.What are the different types of API requests supported in Postman?

Postman supports the following type of requests:

- GET
- POST
- PUT
- PATCH
- DELETE
- COPY
- HEAD
- OPTIONS

- LINK
- UNLINK
- PURGE
- LOCK
- UNLOCK
- PROPFIND
- VIEW

42.What is Jira?

JIRA is a software tool that was developed by the software company, Atlassian. It is used mainly for bug and issue tracking and project management. It is the perfect solution for organizing tasks and [managing agile teams](#).

43.Why is JIRA used? /What are the benefits of using JIRA?

JIRA is used as a project tracking tool with benefits like:

- Ability to track [project progress](#) from time to time
- Use cases include project management, bug tracking, feature implementation
- Easily customizable and extensible
- Platform-independent
- Upfront and fair licensing policy

44.What are the issue types in Scrum project?

In JIRA, an individual unit of work is referred to as issue. The types include:

- Story – single feature that needs to be implemented
- Epic – a big user story
- Bug – problem that needs to be fixed
- Task – generic task that is not a bug or story

45.What is Git

- Git is the most popular, open-source, widely used, and an example of distributed version control system (DVCS) used for handling the development of small and large projects in a more efficient and neat manner.

46.What is a version control system (VCS)?

A VCS keeps track of the contributions of the developers working as a team on the projects. They maintain the history of code changes done and with project evolution, it gives an upper hand to the developers to introduce new code, fixes bugs, and run tests with confidence that their previously working copy could be restored at any moment in case things go wrong.

47.Name the common techniques for performance testing?

There are five testing techniques that are common in the present scenario. They are

1. Spike Testing
2. Load Testing
3. Volume Testing
4. Endurance testing
5. Stress testing

48.What do you mean by the term Jmeter?

It is basically a Java-based tool that is meant to test the performance of apps. Its open-source nature makes it simply the best for this task. A number of testing related to the performance of the apps can simply be handled with this tool. It really doesn't matter whether it's web services or web-based apps. Moreover, it is capable to handle databases as well as FTP servers.

49. Explain different data types in Java.

There are 2 types of data types in Java as mentioned below:

1. Primitive Data Type
2. Non-Primitive Data Type or Object Data type

Primitive Data Type: Primitive data are single values with no special capabilities. There are 8 primitive data types:

- **boolean:** stores value true or false

- **byte**: stores an 8-bit signed two's complement integer
- **char**: stores a single 16-bit Unicode character
- **short**: stores a 16-bit signed two's complement integer
- **int**: stores a 32-bit signed two's complement integer
- **long**: stores a 64-bit two's complement integer
- **float**: stores a single-precision 32-bit IEEE 754 floating-point
- **double**: stores a double-precision 64-bit IEEE 754 floating-point

Non-Primitive Data Type: Reference Data types will contain a memory address of the variable's values because it is not able to directly store the values in the memory. Types of Non-Primitive are mentioned below:

- Strings
- Array
- Class
- Object
- Interface

50. How many types of operators are available in Java?

All types of operators in Java are mentioned below:

1. [Arithmetic Operators](#)
2. [Unary Operators](#)
3. [Assignment Operator](#)

4. [Relational Operators](#)
5. [Logical Operators](#)
6. [Ternary Operator](#)
7. [Bitwise Operators](#)
8. [Shift Operators](#)
9. [instance of operator](#)

