



Top 26 Software Testing and Quality Assurance Interview Questions and Answers

Introduction

You open an app and it crashes. You click a button and nothing happens. These are not small issues — they are signs that testing was not done right.

Software Testing and Quality Assurance is what keeps that from happening. It is not just finding bugs — it is making sure the product actually works the way it was meant to.

QA interviews go deeper than most people expect. Testing types, defect life cycles, automation tools, and methodologies all come up.

This guide covers the most important **Software Testing and Quality Assurance Interview Questions and Answers** to help you prepare properly.

List of Software Testing and Quality Assurance Interview Questions for Freshers

1. What is quality assurance?
2. What distinguishes software testing from quality assurance?
3. What is a quality assurance process's lifecycle?
4. Give an overview of QA's role in software development.
5. Describe the terms "build" and "release." Distinguish between them.
6. How Much Do You Know About Bug Releases and Leakages?
7. What are the differences between stress and load testing?
8. What is ad-hoc testing?
9. What is an effect-cause diagram?
10. What does "monkey testing" mean?
11. What Does "Gorilla Testing" Mean to You?
12. What is a traceability matrix?
13. Explain thread testing.

Software Testing and Quality Assurance

Interview Questions for Freshers

1. What is quality assurance?

Quality assurance aims to guarantee that the software that is produced satisfies all of the criteria and specifications found in the SRS document.

2. What distinguishes software testing from quality assurance?

QA creates plans to prevent possible bugs when software is being developed. Its primary focus is management-related subjects, including checklists, project analysis, and development methods and procedures.

Software testing is a technique for examining a system's operation and searching for any possible bugs. A variety of testing methods are used to find flaws in the product and assess whether they have been fixed.

3. What is a quality assurance process's lifecycle?

The QA lifecycle is PDCA-based:

Plan: During the planning stage of the quality assurance process, the company outlines the steps required to produce software of the highest standard.

Do: Do is the phase where processes are created and evaluated.

Check: The purpose of this stage is to keep an eye on the activities and assess if they meet the demands of the users.

Act: The Act is a step toward implementing the required processes.

4. Give an overview of QA's role in software development.

The responsibilities and roles of QA:

- Defining the goals of the test and the plan of action for reaching them.
- Creating a test plan under the project's requirements and schedule.
- Documenting test failures and carrying out tests in the proper ways (manually or with test execution tools).
- Examining the defects helps identify the underlying reason.
- Fixing errors to prevent them from lowering the final product's quality.
- Notifying developers of software defects with a bug tracking system (such as QA Touch, Mantis, or Bugzilla). Early testing reduces the cost and time required for bug repairs by removing flaws at an early stage.

Check out: [Software Testing and Quality Assurance Salary](#)

5. Describe the terms "build" and "release." Distinguish between them.

Build: The development team sends a "build" to the test team.

If any tests fail or a "build" doesn't meet the requirements, the test team has the right to reject it.

A single release is comprised of many builds.

Release: A product is officially released to the public when it is done so officially.

After a build has been accepted and tested by the test team, it is “released” to clients.

Take your Knowledge
Test Report

Check your Score



6. How Much Do You Know About Bug Releases and Leakages?

Bug leakage: When a flaw is overlooked in earlier iterations or builds of the program, it results in a bug leak.

A flaw that occurs during testing and is subsequently found by the tester or end-user is known as bug leakage.

Bug Release: When a particular software version is made available with multiple known faults or defects, it's called a bug release.

These bugs are usually mentioned in the release notes. Many times, these bugs are of low severity and low priority.

The decision is made, when the organization can afford it, to leave the bug in the released program instead of investing time and resources in fixing it in that particular version.

7. What are the differences between stress and load testing?

What distinguishes each is its purpose:

You can find out how a system reacts to an anticipated load by doing load testing.

You can understand the maximum loads at which a system can operate by conducting stress tests.

To put it another way, stress tests demonstrate how a system will react in various scenarios or to high demand, like a DDoS attack or the Slashdot effect. You might be prepared for unanticipated events in this way.

8. What is ad-hoc testing?

One causal approach to software testing is ad hoc testing. It doesn't follow recognized practices like requirement documentation, test cases, and test plans.

The following are characteristics of ad-hoc testing:

- It is completed on an application following the conclusion of formal testing.
- Its main objective is to cause the program to malfunction without following a preset protocol.
- Adhoc testers ought to be extremely informed about the product they are evaluating.

However, load testing makes sure you meet user expectations, like your service level agreement (**SLA**) commitments. Hence, ensuring a positive user experience is the goal, rather than breaking the program. You may confidently roll out new code due to it.

Check out: [Software Testing & QA Architect Course Syllabus PDF](#)

9. What is an effect-cause diagram?

Using a set of criteria, the cause-and-effect graph is a black box testing technique that finds the fewest test cases necessary to fully evaluate the product's scope. It also draws attention to the relationship between a certain outcome and the factors affecting it.

The shortened test execution time and cost are the true advantages of cause-and-effect graph testing.

10. What does “monkey testing” mean?

In software development, monkey testing is used to evaluate an application's efficacy. In “monkey testing,” a tester enters data into the program at random without intending to cause a crash. It is suitable for load testing since it attempts to break the application by providing arbitrary inputs.

Conventional approaches may not necessarily reveal all problems at regular intervals. If random inputs are given, the likelihood of finding these issues increases. This strategy is used throughout the system to identify bugs that are typically overlooked through more conventional techniques.

11. What Does “Gorilla Testing” Mean to You?

While gorilla testing laboriously tests every line of code until random inputs cause it to break, application resilience is carefully examined manually.

Gorilla testing and monkey testing differ primarily in that the former tests individual modules, while the latter assesses the system as a whole. The

modules of each product get random inputs, both legitimate and invalid, until a module crashes.

To test the application's resilience, this step is completed for every module in the final stages of the software development cycle. Because of its rigor, it is frequently referred to as "torture testing" or "fault tolerance testing."

12. What is a traceability matrix?

In software development projects, a special type of document called a traceability matrix is used to monitor, identify, and validate the progress of a given capability or component. helps link and follow the requirements for business, applications, security, and other areas to their fulfillment, testing, or implementation. It provides information on the status of the project's needs in terms of their level of completion and evaluates and contrasts different system components.

A traceability matrix is often a worksheet-style document with a table. To compare the two sets of values, an identifier for one group is positioned in the top row, and an identifier for the other set is positioned in the left column. If something is similar or connected, a mark is made.

13. Explain thread testing.

Software testing that looks at a task's essential functional characteristics is called thread testing (thread). It's one of the gradual methods that's frequently used when system integration testing first starts.

List of Software Testing and Quality Assurance Interview Questions for Freshers

14. Describe testware.
15. What is a quality audit?
16. What is the difference between regression and retesting?
17. What is the Defect Leakage Ratio?
18. What does the “test-driven development rule” entail?
19. Which Five Dimensions of Risk Are There?
20. Explain the various types of documentation for software quality assurance.
21. How Well-Informed Are You About Regression Testing? Which test cases should be used in this procedure?
22. What distinguishes exploratory and monkey testing from ad hoc testing?
23. What Distinguishes Functional Testing from Non-Functional Testing?
24. How Much Do You Know About Drivers and Stubs? Distinguish Between Them
25. How Do You Determine When to Give Up on a Test?
26. Regarding Bug/Defect Triage in the Context of Quality Assurance, What Do You Know?

Software Testing and Quality Assurance

Interview Questions for Experienced

14. Describe testware.

The objects produced throughout the testing process that are required to organize, plan, and execute tests are referred to as testware.

Testware includes files, databases, environments, scripts, inputs, expected results, setup and teardown procedures, documentation, and other software or utilities required for testing.

A configuration management system that is appropriately used to manage and store testing instruments is known as testware.

There are two ways that testware differs from standard software:

- Testers create it with a specific goal in mind.
- It has many users and different quality standards.

15. What is a quality audit?

The fundamental objective of conducting an audit of a software testing phase is to achieve the required standard. In software testing, an audit compares a software product to predetermined criteria. To make sure the resulting product conforms with the specifications, a quality audit establishes whether the procedure being used and implemented in the testing process is defined.

Check out: [Software testing and QA architect Salary](#)

16. What is the difference between regression and retesting?

Retesting: Retesting is a method used to confirm specific test cases that are found to contain errors during final execution.

- When these flaws are discovered during software testing, testers usually report them to developers so they can be fixed.
- After fixing the issues, the developers provide the files back to the testers for approval. The process we call this continuous process is retesting.

Regression testing: Regression testing verifies whether an update to the code has badly impacted the capabilities and functionalities of the program as it is.

Whereas retesting is limited to failed test cases, regression testing is carried out on passed test cases.

- Regression testing searches for unexpected side effects, whereas retesting confirms that the original defect has been corrected.
- Verifying defects is part of retesting, as opposed to regression testing.
- Whereas regression testing is referred to as generic testing, retesting is a planned test.
- Regression testing can be done automatically, but retesting cannot.

17. What is the Defect Leakage Ratio?

A statistic called defect leakage is used to assess how well quality assurance testing is carried out or how many issues are missed.

Calculation of the defect leakage ratio formula:

Defect Leakage = (No. of Defects Found in UAT / No. of Defects Found in QA Testing)

18. What does the “test-driven development rule” entail?

- Until the production code is needed to pass a failing unit test, the QAs are not allowed to write any.
- You may only write as much of a unit test as is required for it to fail; compilation errors are considered failures.
- The QAs may write only the production code required to pass the one failed unit test.

Take your Knowledge
Test Report

Check your Score



19. Which Five Dimensions of Risk Are There?

The following are the five aspects of risk:

Schedule: impossible deadlines, such as creating a substantial piece of software in a single day.

Client: ambiguous requirements, requirements that are subject to change, and imprecise requirements descriptions.

Human Resources: Insufficient personnel with the necessary knowledge for the project.

System Assets: Negative effects will result from not obtaining all necessary resources, including software and hardware tools or licenses.

Quality: Many issues, including a shortage of resources, a rigid delivery schedule, and frequent adjustments, will affect the quality of the product.

20. Explain the various types of documentation for software quality assurance.

Test Policy: This high-level document describes the organization's core testing goals, techniques, and guiding ideas.

Testing Strategy: The project's test levels and types are described in the testing strategy.

Test Plan: A test plan is a comprehensive planning document that contains details on the goal, approach, available tools, timetable, and additional testing activities. It is frequently presented as an interactive PDF or digital flipbook.

Requirements Traceability Matrix: This document's requirements and test cases are connected.

Test Scenario: A software system's test scenario is any event or component that one or more test cases could validate.

Test Case: It consists of a set of input values, results, and anticipated postconditions for the execution.

Test Data: pre-run data that is available before a test is conducted.

Defect Report: A documented account of every software system error that stops the system from functioning as planned is called a defect report.

Test Summary Report: The testing procedures and test outcomes are described in this high-level report.

21. How Well-Informed Are You About Regression Testing?

Which test cases should be used in this procedure?

Testing to make sure a software upgrade won't affect how the product works as it does now is called regression testing.

The following test scenarios can be used in practical regression tests:

- Users can see more features if they are visible.
- scenarios that look at the fundamental characteristics of the product
- Case studies of functionalities that have changed recently and significantly
- Each Integration Test Case
- Every Detailed Test Case
- Boundary value test examples
- Several instances of failed test cases

Check out: [Software Testing and Quality Assurance Project Ideas](#)

22. What distinguishes exploratory and monkey testing from ad hoc testing?

Both ad hoc and monkey testing employ a casual approach. Adhoc testing necessitates that testers have a deep understanding of the program, whereas monkey testing does not.

A summary of the differences between ad-hoc and exploratory testing is as follows:

- Software is tested ad hoc—that is, without consulting specifications or requirements documentation. Investigating the application and comprehending the program are necessary for exploratory testing.
- Ad hoc testing does not require documentation. It is necessary to document exploratory testing.
- The main objective of ad-hoc testing is to improve the testing procedure. The main objective of exploratory testing is to learn more about the application.
- Exploratory testing is a formal technique, whereas ad-hoc testing is a non-formal approach.

23. What Distinguishes Functional Testing from Non-Functional Testing?

Functional Testing:

- Functional testing is used to validate each feature and function of software.
- Client needs are given priority during functional testing.
- Validating program activities is the goal of functional testing.
- A functional test can involve verifying the login process.
- Functionality defines what a product performs, whereas non-functionality explains how a product operates.
- Before nonfunctional testing, functional testing is carried out.

Non-Functional Testing:

- Nonfunctional testing validates nonfunctional characteristics like performance, usability, and reliability.
- The client's expectations serve as the basis for non-functional testing.
- Manually performing non-functional testing is challenging.
- Non-functional testing verifies the functionality of the software.
- An example of non-functional testing might verify that the dashboard loads in two seconds or less.
- Functional testing is carried out first, followed by non-functional testing.

24. How Much Do You Know About Drivers and Stubs?

Distinguish Between Them

In software testing, the phrases "stub" and "drivers" refer to copies of the modules that take the place of newly added or missing modules.

Drivers are designed to enhance the testing process and are mostly utilized in individual bottom-up integration testing. Top-down integration testing is the primary application for stubs.

The essential distinction between them is:

1. Top-down integration testing uses stubs, whereas bottom-up integration testing uses drivers.
2. Stubs are a coded replica of the called function. A portion of code acts as a driver by imitating the actions of a calling function.
3. Stubs promote the creation of incomplete and absent modules. Test cases are passed to other programs by drivers, who also invoke test modules.

4. When high-level modules are tested and lower-level modules are being developed, stubs are taken into consideration. Higher-level modules have not yet been developed, although drivers are taken into account during the testing of lower-level modules.

25. How Do You Determine When to Give Up on a Test?

As project managers or leads, we occasionally need to postpone testing to finish the product sooner. In certain cases, we have to assess whether the testers have tested the goods sufficiently.

Several factors need to be taken into account in real-time projects when determining when to end testing:

- If the deadlines for testing or release are fulfilled,
- By inputting the test case pass rate that was established,
- If the real-time project's risk is less than what is allowed,
- If every significant issue and obstacle has been fixed,
- If the contribution satisfies the criteria.

26. Regarding Bug/Defect Triage in the Context of Quality Assurance, What Do You Know?

Defect triage, often known as bug triage, is a regularly used technique in software testing. It is essential to explain the significance and gravity of the errors. The impact a bug has on the application under test determines how serious it is.

The order in which a defect needs to be fixed or addressed is called priority. In essence, defect triage is an approach to rebalance the process, which is usually troublesome for the test team because of insufficient

resources. In defect triage, defects are often ranked only according to their risk, likelihood of recurrence, and severity.

Conclusion

You have now gone through these **Software Testing and Quality Assurance Interview Questions and Answers**. Use them to revise before your interview. Read each question. Try to answer it in your own words. This helps you speak with more clarity.

Practice is important. Test simple applications. Write test cases. Check how bugs are reported and fixed. This builds real understanding of Software Testing and Quality Assurance.

If you want structured learning, join a [Software Testing and Quality Assurance Course](#) at Softlogic Systems. The course focuses on hands on practice and helps you prepare for testing roles.