Share on your Social Media

Top 14 Software Testing Interview Questions and Answers

Published On: March 16, 2022

Software Testing Interview Questions and Answers

Software Testing is a part and parcel of the software development process. Without Software Testing no software will see the light of the day. Which is why, being employed in the Software Testing sector can have numerous benefits as a career including a long-lasting career. So, these **Software Testing Interview Questions and Answers** will give you exposure to some of the most frequently asked Software Testing Interview Questions like, Software Testing Life Cycle, difference between Shift Left Testing and Shift Right Testing, Bug Life Cycle etc. So, explore down below to be well prepared for your Software Testing interview.

Featured Articles

Want to know more about becoming an expert in IT?

Click Here to Get Started

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER IBM

Quick Enquiry

Related Course at SLA

- Software Testing Training in Chennai
- Software Testing Online Training

Related Posts

Top 12 Business Intelligence and Data Analytics Interview

Download Software Testing Interview Questions PDF

Software Testing Interview Questions and Answers

1. What is Software Testing?

Software testing involves evaluating the quality, functionality, and performance of a software product before its launch. Testers perform this process either through manual interaction with the software or by running automated test scripts to detect bugs and errors, ensuring that the software operates as intended. Moreover, software testing aims to confirm the adherence to business logic and pinpoint any discrepancies in requirements that need immediate addressing.

2. Describe the Software Testing Life Cycle (STLC).

The Software Testing Life Cycle (STLC) provides a approach to testing software, which has the following key phases:

- **Requirement Analysis:** Testers assess requirements to understand software functionality and identify testable aspects and potential risks.
- **Test Planning:** Creation of a comprehensive test plan, detailing approach, goals, scope, resources, schedule, and test cases.
- **Test Case Development:** Formulation of test cases based on requirements and design documents, specifying steps, outcomes, and necessary data.
- **Test Environment Setup:** Configuration of hardware, software, and network infrastructure to mirror the production environment for testing.

Questions and Answers

Published On: June 19, 2024

Business Intelligence and Data Analytics Interview Questions and Answers Our Business Intelligence and Data Analytics...

Top 20 Azure DevOps Interview Questions and Answers

Published On: June 19, 2024

Azure DevOps Interview Questions and Answers One of the most in-demand skills in the IT...

Top 20 AWS DevOps Interview Questions and Answers

Published On: June 19, 2024

AWS DevOps Interview Questions and Answers The integration of AWS and DevOps is doing magic...

Top 14 MEAN Stack Interview Questions and Answers

Published On: June 19, 2024

- **Test Execution:** Execution of test cases according to the plan, either manually or through automated scripts, with documentation of results and detected defects.
- **Defect Tracking and Management:** Logging and assessment of defects in terms of priority and severity, with subsequent resolution by the development team.
- **Test Reporting:** Documentation of test outcomes, including executed test cases, pass/fail status, and open defects, distributed to stakeholders for insight into software quality.
- **Test Closure:** Conclusion of testing activities, involving analysis of the process, documentation of lessons learned, and preparation of closure reports.

3. Differentiate Shift Left Testing and Shift Right Testing in Software Testing.

Aspect	**Shift Left Testing**	**Shift Right Testing**
Initiation of Testing	Begins at an early stage of SDLC, often during development or even before coding.	Extends testing beyond development and testing phases, including the production environment.
Orientation	Prioritizes averting defects by detecting them early in development.	Emphasizes real-world feedback for continuous improvement post-deployment.
	Frequently	Utilizes

MEAN Stack Interview Questions and Answers Since MEAN Stack combines several other applications as part...

Automation	integrates extensive automation for quick issue identification and resolution.	continuous monitoring and feedback loops to refine subsequent releases.
Collaboration	Emphasizes collaboration among developers, testers, and stakeholders from project inception.	Underlines user-centric feedback acquisition to comprehend software performance.
Testing Approach	Prominently focuses on testing against requirements to ensure alignment with customer expectations.	Frequently involves exploratory testing to uncover unforeseen issues or usability concerns.
Enhancement Approach	Concentrates on preemptive defect mitigation and early quality integration.	Aids in mitigating risks associated with deploying new features or updates through swift issue identification.

Software Testing Syllabus PDF

4. What is the Bug Life Cycle?

The Bug Life Cycle, also referred to as the Defect Life Cycle, outlines the sequential stages a bug or defect undergoes from its detection to its resolution.

Below is a typical Bug Life Cycle:

- **New/Reported:** Identification of the bug occurs, reported by a tester, user, or automated testing tool. At this juncture, the bug receives a unique identifier and is logged into the bug tracking system.
- **Assigned:** The bug is delegated to a developer or team responsible for rectifying it. The assigned developer commences analyzing the bug to comprehend its nature and scope.
- **Open:** Confirmation of the bug's existence by the developer initiates work towards its resolution. The bug retains the "Open" status until resolved.
- **In Progress:** The developer actively engages in rectifying the bug. Activities may involve coding, debugging, and testing to pinpoint the root cause and implement a solution.
- **Fixed:** The developer deems the bug fixed and implements the requisite changes in the codebase. Subsequently, the bug is labeled as "Fixed" and is prepared for verification.
- **Pending Retest:** The bug awaits retesting to ascertain whether the fix successfully resolves the issue without introducing new complications.
- **Retest:** Testers conduct a verification process to ensure the effectiveness of the fix. If the bug is confirmed as resolved, it proceeds to the subsequent stage. Otherwise, it is reopened and returned to the developer for further action.
- **Closed:** Upon successful verification and validation of the bug fix, it is marked as "Closed." Consequently, the bug is considered resolved and inactive.

5. Explain the importance of software

testing in the process of software development.

Software testing is integral to the software development process for the following reasons:

- **Quality Assurance:** Testing ensures software meets requirements and functions correctly, enhancing user satisfaction by identifying and resolving defects.
- **Risk Reduction:** It identifies risks and vulnerabilities early, preventing costly errors post-deployment.
- **Cost Efficiency:** Early defect detection reduces rework costs, saving resources.
- **Customer Satisfaction:** High-quality software meets user needs, improving satisfaction and loyalty.
- **Compliance:** Testing ensures adherence to regulatory standards, minimizing legal and financial risks.

6. What are the different types of software testing?

Software testing encompasses various types, including:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing
- Regression Testing
- Performance Testing
- Security Testing
- Usability Testing.

7. What is the difference between verification and validation?

Verification ensures adherence to specifications during software development, while validation ensures that the software meets user requirements and expectations.

8. What is the difference between black-box testing and white-box testing?

Black-box testing examines software functionality without internal structural knowledge, whereas white-box testing assesses based on understanding the internal code and structure.

9. What is a test case?

A test case constitutes a predefined set of conditions or variables used by testers to assess whether a system fulfills requirements or functions correctly.

10. What is exploratory testing?

Exploratory testing involves manual exploration of software without predefined test cases, enabling testers to uncover defects or issues based on their experience and intuition.

Software Tester Salary

11. What is a traceability matrix in Software Testing?

In software testing, a traceability matrix acts like a blueprint, illustrating the relationships between requirements and their corresponding test cases. Imagine it as a grid that tracks which tests ensure each feature (requirement) functions as planned.

Here's how it benefits the testing process:

- **Thorough Testing:** By mapping requirements to test cases, you can guarantee all functionalities are tested, leaving no stone unturned.
- **Swifter Debugging:** When a bug arises, the matrix helps pinpoint the exact requirement it affects, making the debugging process much faster.
- **Clear Communication:** The matrix serves as a

central source of information, providing a transparent view of testing progress for both testers and stakeholders.

There are two main ways traceability works in this context:

- **Forward Tracing:** This ensures each requirement has one or more test cases designed to validate it.
- **Backward Tracing:** This confirms that every test case links back to a specific requirement, preventing the creation of unnecessary tests.

12. What’s the difference between static testing and dynamic testing in software development?

Here’s a breakdown of static vs. dynamic testing:

Static Testing:

- This approach analyzes software components like code, design documents, and requirements without running the program itself. It’s like examining blueprints for a house to identify potential issues before construction begins.
- Static testing helps prevent defects by catching errors early on.
- Examples include code reviews, where developers inspect each other’s code for mistakes, and walkthroughs, where a team discusses the design and logic of the software.

Dynamic Testing:

- This method involves executing the actual software code to see how it behaves under various conditions. It’s like testing the built house to ensure everything functions as planned.
- Dynamic testing helps find and fix defects in the running application.
- Examples include unit testing (testing individual modules of code), integration testing

(testing how different modules work together), and system testing (testing the entire software system with real-world data).

13. What is the purpose of test cases and test scenarios in software testing?

Test scenarios and test cases are both crucial elements within the software testing domain, however, they address distinct aspects of the testing process.

Test Scenarios:

- These provide a comprehensive overview of the functionalities that require testing.
- They are formulated at a high level, outlining the functionality from the perspective of the end-user.
- For instance, a test scenario might state: “The login feature should grant users access to the system upon successful authentication with valid credentials.”

Test Cases:

- In contrast, test cases delve into the intricacies of how to evaluate a specific functionality.
- They are considerably more detailed, providing step-by-step instructions for executing the test along with the anticipated outcomes.
- An example test case for the login scenario could involve specific steps such as entering a valid username and password, verifying a successful login occurs, and ensuring error messages are displayed for invalid credentials.

14. What is Test Driven Development (TDD)?

TDD breaks the conventional development mold by prioritizing testing from the get-go.

- Traditionally, code is written first, followed by testing to unearth bugs.

- In TDD, the process starts with crafting an automated test case that initially fails. This test case outlines the desired behavior for a particular feature.
- Developers then write just enough code to make the test pass.
- Once the test succeeds, they refactor the code to enhance readability and maintainability without impacting functionality.

Conclusion

This **Software Testing Interview Questions and Answers** will give you the much needed knowledge required from you to impress anyone in your Software Testing Interview. We have curated this blog with some of the most popular and most frequently asked Software Testing Questions in the interview. So, learn from this and get placed in your job as Software Tester.

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

- Python
- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS
- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking
- VBA Macros
- Mobile App Development
- DevOps

- SLA’s Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Social Media Links

f X ◎ in ▶

Review Sources

- Google
- Trustpilot
- Glassdoor
- Mouthshut
- Sulekha
- Justdial
- Ambitionbox
- Indeed
- Software Suggest
- Sitejabber