SOFTWARE TESTING & QA

manual testing interview questions with answer

Published: September 3, 2026 | Index ID: #-

Quick Takeaways

- Understand the foundational difference between SDLC and STLC to answer process-oriented questions.
- Master the nuance between Severity and Priority; this is a high-frequency interview topic.
- Be prepared for scenario-based questions that test your logical approach to ambiguous requirements.
- Modern manual testing involves understanding APIs, databases, and user experience, not just clicking buttons.
- Emphasis on E-E-A-T: Demonstrating real-world experience through detailed bug reports and test plans.

The role of a manual tester has evolved significantly in the last decade. While automation often dominates the conversation, manual testing remains the bedrock of quality assurance. It provides the human perspective, intuition, and exploratory edge that scripts simply cannot replicate. Whether you are a fresh graduate aiming for your first role or a seasoned professional looking to transition into a senior QA position, mastering the core concepts of manual testing is non-negotiable. This guide provides over 3,000 words of expert-level knowledge, structured to help you ace your next interview with confidence.

Section 1: Foundational Manual Testing Concepts

In any interview, the first few questions will likely gauge your understanding of the basics. Don't underestimate these; your ability to explain complex terms in simple, accurate language demonstrates seniority and clarity of thought.

1. What is Manual Testing and why is it still relevant?

Manual testing is the process of manually verifying that software performs as expected by following test cases without the aid of automation tools. It is crucial because automation cannot judge user experience (UX), visual aesthetics, or the "feel" of an application. Manual testers find bugs that automated scripts might miss because humans can adapt to unexpected behaviors on the fly.

2. Distinguish between SDLC and STLC.

Software Development Life Cycle (SDLC) encompasses the entire process of creating software, from requirement gathering to deployment and maintenance. Software Testing Life Cycle (STLC) is a subset of SDLC that focuses specifically on the testing phases, ensuring that quality standards are met at every step.

3. Explain the Software Testing Life Cycle (STLC) phases.

STLC consists of the following phases:

- Requirement Analysis: Understanding what needs to be tested.
- Test Planning: Defining the strategy, scope, and resources.
- Test Case Development: Creating detailed steps and expected results.
- Environment Setup: Preparing the hardware and software for testing.
- Test Execution: Running the tests and documenting results.
- Test Cycle Closure: Final reporting and analysis of the testing process.

Section 2: Comparative Analysis - Essential Tables

Interviews often involve comparing two similar concepts. Use these tables to visualize the differences clearly.

Table 1: Manual Testing vs. Automation Testing

Feature	Manual Testing	Automation Testing
Human Observation	High; catches UX/UI issues.	None; follows programmed logic.
Cost	Low for small, one-off projects.	High initial investment in tools/scripts.
Reliability	Prone to human error over time.	Highly reliable for repetitive tasks.
Execution Speed	Slow; limited by human speed.	Very fast; can run 24/7.
Exploratory Testing	Possible and encouraged.	Not possible (script-dependent).

Table 2: Verification vs. Validation

Aspect	Verification	Validation
Question	Are we building the product right?	Are we building the right product?
Focus	Process, documentation, architecture.	Final product, user requirements.
Methods	Reviews, walkthroughs, inspections.	Black box testing, white box testing.
Timing	Done before validation.	Done after verification.

Section 3: Test Design Techniques & Methodologies

To be an effective tester, you must know how to design test cases that provide maximum coverage with minimum effort. Interviewers love asking about these techniques.

4. What is Boundary Value Analysis (BVA)?

BVA is a black-box testing technique based on testing at the boundaries between partitions. For example, if a text field accepts values from 1 to 100, the boundary values to test would be 0, 1, 2, 99, 100, and 101. Most errors occur at the edges of input ranges.

5. Explain Equivalence Class Partitioning (ECP).

ECP involves dividing input data into classes where all elements are expected to behave similarly. Instead of testing every single value (e.g., 1-100), you pick one representative value from each valid and invalid partition. This significantly reduces the number of test cases while maintaining coverage.

6. What is the difference between Smoke and Sanity Testing?

Smoke Testing is performed on initial builds to ensure the "critical" functionalities work. If smoke testing fails, the build is rejected. **Sanity Testing** is a subset of regression testing performed on stable builds to verify that specific bug fixes or changes work as intended.

Section 4: Advanced Scenario-Based Questions

Mid-to-senior level interviews often involve scenarios. You aren't just asked "what" something is, but "how" you would handle it.

Scenario: You find a critical bug two hours before a major release. What do you do?

Answering this requires showing leadership and process awareness:

1. Immediate Documentation: Log the bug with all necessary details, logs, and screenshots.
2. Impact Analysis: Assess how the bug affects the end-user and the system stability.
3. Communication: Inform the Project Manager, Developers, and stakeholders immediately.
4. Risk Assessment: Participate in a triage meeting to decide: Should we delay the release, fix and retest, or release with a known issue?
5. Decision Support: Provide the data necessary for the team to make an informed choice.

Scenario: How would you test a login page without requirements?

This tests your exploratory testing skills and common sense. You should categorize your testing into:

- UI/UX: Alignment, branding, fonts, responsiveness.

- Functionality: Valid credentials, invalid credentials, empty fields, password masking.
- Security: SQL injection, brute force attempts, session management.
- Edge Cases: Extremely long usernames, copy-pasting, special characters.

Section 5: Defect Management and Reporting

The primary output of a manual tester is a defect report. If you can't communicate a bug, you haven't effectively tested the product.

7. What are the components of a good Bug Report?

A professional bug report should include:

- Defect ID: Unique identifier.
- Summary/Title: Brief, descriptive headline.
- Description: Context of the bug.
- Steps to Reproduce: Detailed, numbered steps.
- Expected Result: What should have happened.
- Actual Result: What actually happened.
- Severity & Priority: Impact and urgency.
- Environment: OS, Browser, Build version.
- Attachments: Screenshots, video recordings, or log files.

8. Severity vs. Priority: Explain the Difference.

Severity describes the technical impact of a bug on the application's functionality (e.g., System Crash = High Severity). **Priority**describes how quickly the business needs the bug fixed (e.g., A typo in the company logo on the homepage = High Priority, but Low Severity).

Section 6: Testing Types in Depth

Black Box, White Box, and Grey Box Testing

- Black Box: Testing without internal knowledge of code. Focus is on input/output.
- White Box: Testing with knowledge of the internal code structure, logic, and paths.
- Grey Box: A hybrid approach. Testing with partial knowledge of the internal structure, often used for integration testing.

Regression Testing vs. Retesting

Retesting is specifically checking if a previously found bug is fixed. **Regression Testing** is checking the entire application (or relevant parts) to ensure that recent changes haven't broken existing functionality.

Section 7: Soft Skills and Strategy

A great tester is also a great communicator. In an interview, show that you can collaborate with developers rather than being an "enemy" of the code.

9. How do you handle a situation where a developer says "It's not a bug, it's a feature"?

Respond with data and empathy. Show that you would refer back to the requirement documents, check industry standards for UX, and if the discrepancy remains, discuss it with the Product Owner for final clarification.

10. What is Exploratory Testing?

Exploratory testing is simultaneous learning, test design, and execution. Unlike scripted testing, the tester explores

the application like an end-user to find bugs that weren't anticipated during the formal test case writing phase.

Section 8: Real-World Blueprint for a Testing Strategy

When asked how you approach a new project, follow this blueprint:

1. Analyze Requirements: Read the PRD (Product Requirement Document) and identify ambiguities.
2. Scope Definition: Decide what is in-scope and out-of-scope for the release.
3. Test Plan Creation: Document the tools, resources, and timelines.
4. Test Case Design: Use BVA and ECP to create robust test cases.
5. Execution & Logging: Run tests, log defects, and track them through their lifecycle.
6. Reporting: Provide a Summary Report indicating the quality status of the build.

Frequently Asked Questions (FAQ)

Conclusion: Preparing for Success

Manual testing is much more than finding bugs; it is about advocating for the user and ensuring that the software provides real value. When preparing for your interview, remember that technical knowledge is only half the battle. Your attitude, attention to detail, and ability to handle pressure are just as important. Use the questions and structures provided in this guide to build a solid foundation. Practice your delivery, refine your bug reporting skills, and enter your next interview as a QA expert ready to make an impact.

Tags: #manual testing #qa interview prep #software testing #test cases #stlc #sdlc #defect management