Software Testing Q&A with Scenario-Based Answers

1. What is manual testing, and why is it important?

Manual testing is the process of testing software applications without using automation tools. Testers manually execute test cases to identify defects.

Scenario: Imagine you're testing a login page. You enter the correct username and password to see if it logs in. Then, you test with the wrong password to check error messages. You're doing it all manually - that's manual testing.

Importance:

- Helps find real user issues
- Needed when UI/UX testing is required
- Useful in early stages before automation is set up

2. Can you explain the Software Testing Life Cycle (STLC)?

STLC is a step-by-step process that testers follow to ensure quality in a structured way.

Stages with Scenario:

1. Requirement Analysis: Understand what the app should do. e.g., The app should allow users to make payments.
2. Test Planning: Decide resources, tools, timelines. e.g., Use Postman for API testing, team of 3 testers.
3. Test Case Design: Write test cases. e.g., Check if payment succeeds with valid card.
4. Test Environment Setup: Set up test servers/databases.
5. Test Execution: Run test cases and log defects.
6. Test Closure: Final report, lessons learned.

3. What is the difference between verification and validation?

Verification: Are we building the product right? (Process-oriented)

Validation: Are we building the right product? (User-oriented)

Scenario:

- Verification: Checking if login code follows the design.
- Validation: Testing if users can actually log in and use the system.

4. What is a test case? What are key components of a good test case?

A test case is a step-by-step guide to test a specific functionality.

Key Components:

- Test Case ID
- Description
- Preconditions
- Test Steps
- Expected Result
- Actual Result
- Status (Pass/Fail)
- Remarks

Scenario:

Test Case for Login:

- Step: Enter valid username & password
- Expected: Dashboard opens
- Actual: Dashboard opens -> Pass

5. What is the difference between a test plan and a test strategy?

Test Plan: Project-specific detailed document (who, what, when, how)

Test Strategy: Organization-level high-level approach for testing

Scenario:

If your project is an E-commerce app, the test plan tells how login, cart, and payment will be tested. The test strategy explains overall practices like manual vs. automation, tools used, etc.

6. What is the purpose of a Requirement Traceability Matrix (RTM)?

RTM maps requirements to test cases to ensure every feature is tested.

Scenario:

Requirement: "Users can reset password"

RTM shows: Test Case TC-105 is written and passed for it. -> Confirms all requirements are covered in testing.

7. What are positive and negative testing?

Positive Testing: Test with valid data (expected use)

Negative Testing: Test with invalid data (unexpected use)

Scenario (Login):

- Positive: Valid username & password -> should log in
- Negative: Empty password field -> should show error

8. What is the difference between severity and priority in bug reporting?

Severity: How bad the bug is (technical impact)

Priority: How soon it should be fixed (business impact)

Scenario:

- High Severity, Low Priority: App crashes in rare condition.
- Low Severity, High Priority: Company logo is wrong on homepage.

9. What are functional vs. non-functional testing?

Functional Testing: Tests what the system does (features)

Non-Functional Testing: Tests how the system behaves (performance, usability, etc.)

Scenario:

- Functional: "User can add items to cart"
- Non-Functional: "Cart loads in <2 seconds under 1000 users"

10. What is exploratory testing, and when would you use it?

Exploratory testing means testing without predefined test cases, based on intuition and experience.

When to use:

- When requirements are incomplete
- During initial builds
- Short deadlines

Scenario:

You're given a new shopping app with no test cases. You explore the app freely-try adding items, changing settings, logging out-to find bugs.

11. What are some examples of boundary value testing?

This testing focuses on edge cases.

Scenario:

If a field accepts 1-100:

Test inputs:

- Below lower: 0
- At lower: 1
- Just above lower: 2
- Below upper: 99
- At upper: 100
- Just above upper: 101

12. How do you decide what to test when requirements are unclear?

Approach:

- Communicate with stakeholders
- Refer to similar projects
- Perform exploratory testing

- Use error guessing technique

Scenario:

The client gives you a payment page without full requirements. You test card validation, payment success/failure, confirmation messages, error handling - based on common scenarios.

13. What is a bug life cycle?

Bug life cycle is the journey of a bug from detection to closure.

States:

- New
- Assigned
- Open
- Fixed
- Retested
- Closed / Reopened
- Deferred / Rejected

Scenario:

You find an issue in the payment page -> mark as "New"

Developer fixes it -> mark "Fixed"

You test it again -> "Retested" and "Closed"

14. What are common fields in a bug report?

Fields:

- Bug ID
- Title
- Description
- Steps to Reproduce
- Expected Result
- Actual Result

- Severity & Priority

- Attachments (screenshots/logs)

- Status

- Environment

Scenario:

Bug: "Cart not updating total price"

Steps: Add 2 items -> Expected: Rs.500 -> Actual: Rs.0

Attach screenshot, mark severity = High

15. Can you explain smoke and sanity testing?

Smoke Testing: Basic test to check app stability before deeper testing

Sanity Testing: Quick test to verify specific bug fixes or functionality after changes

Scenario:

After a new build is deployed:

- Smoke Test: Can we log in? Navigate? Load dashboard?

- Sanity Test: Was the cart issue fixed? Check only that feature.