# Manual Testing Glossary - 200 Q&A Simplified

## Manual Testing

Testing done by humans without automation tools.

## Software Testing

Checking software to make sure it meets requirements and works correctly.

## Verification

Ensures product is built the right way.

## Validation

Ensures the right product is built.

## Defect

A problem in the software that makes it behave incorrectly.

## Bug Report

Document with defect details, steps, and expected results.

## Test Case

Step-by-step instructions to check a feature.

## Test Scenario

High-level idea of what to test.

## Test Plan

Document with testing scope, approach, and goals.

## Test Strategy

High-level testing principles for a project.

## Exploratory Testing

Testing without scripts, just exploring app.

## Ad-hoc Testing

Unstructured testing to quickly find issues.

## Smoke Testing

Basic test to check build stability.

## Sanity Testing

Quick test to check small fixes didn’t break app.

## Regression Testing

Re-check old features after changes.

## Functional Testing

Check features against requirements.

## Non-Functional Testing

Check performance, usability, security.

## Usability Testing

Check if app is user-friendly.

## SDLC

Software Development Life Cycle, stages from requirement to release.

## STLC

Software Testing Life Cycle, steps from planning to closure.

## RTM (Requirement Traceability Matrix)

Maps requirements to test cases.

## Test Environment

Setup of hardware, software, and data for testing.

## Boundary Value Analysis

Test values at input edges (min, max).

## Equivalence Partitioning

Divide inputs into valid and invalid groups.

## Positive Testing

Test with valid inputs.

## Negative Testing

Test with invalid inputs.

## Acceptance Testing

Check if app meets business needs.

## Alpha Testing

Internal team testing before release.

## Beta Testing

Real users test before final release.

## Compatibility Testing

Check app on different devices, browsers, OS.

## Performance Testing

Measure app speed under load.

## Load Testing

Check app under expected load.

## Stress Testing

Push app beyond limits.

## Recovery Testing

Check app recovery after crash.

## Installation Testing

Check install and uninstall process.

## Defect Severity

How badly defect affects system.

## Defect Priority

How quickly defect should be fixed.

## Defect Life Cycle

Stages: New → Open → Fixed → Retest → Closed.

## Error vs Defect vs Failure

Error = mistake, Defect = found in test, Failure = defect in production.

## Root Cause Analysis

Finding main reason for a defect.

## QA vs QC

QA = process focus, QC = product focus.

## Test Coverage

How much requirements/code tested.

## Defect Leakage

Defects missed in testing but found later.

## Defect Density

Number of defects per size of module.

## Localization Testing

Check app for specific language/region.

## Globalization Testing

Check app works worldwide.

## Static Testing

Review docs/code without running app.

## Dynamic Testing

Running app to check behavior.

## Black-box Testing

Test without knowing internal code.

## White-box Testing

Test with full knowledge of code.

## Grey-box Testing

Test with partial code knowledge.

## Unit Testing

Check smallest code parts (methods).

## Integration Testing

Check modules working together.

## System Testing

Check full app.

## End-to-End Testing

Check complete flow (like shopping cart).

## Monkey Testing

Random inputs to break app.

## Pair Testing

Two testers test together.

## Session-based Testing

Exploration done in fixed time slots.

## Checklist Testing

Follow a list of checks.

## Decision Table Testing

Use table of conditions and actions.

## State Transition Testing

Check changes in app states.

## Error Guessing

Use experience to guess defects.

## Test Harness

Setup of tools to run tests.

## Defect Clustering

Most defects found in few modules.

## Pesticide Paradox

Same tests lose power over time.

## Shift-left Testing

Testing starts earlier in cycle.

## Shift-right Testing

Testing done in production.

## Continuous Testing

Run tests in CI/CD pipeline.

## Bug Triage

Meeting to prioritize and assign defects.

## Defect Prevention

Steps to stop defects before happening.

## Test Summary Report

Document summarizing test results.

## Test Closure Report

Final report at end of testing.

## Walk-through

Informal review of docs/code.

## Inspection

Formal detailed review.

## Peer Review

Review by team members.

## Risk-based Testing

Focus on high-risk areas first.

## Acceptance Criteria

Conditions that must be met to accept feature.

## Exploratory Testing Advantage

Finds bugs quickly without much planning.

## Showstopper Bug

Defect that blocks release/testing.

## Testing Debt

Unfinished/skipped tests to do later.

## Heuristic

Experience-based guess for testing.

## Test Optimization

Reduce cases while keeping coverage.

## Test Lead Role

Plans and manages testing team.

## Test Manager Role

Oversees full testing strategy.

## Critical Bug Before Release

Report quickly, assess impact before release.

## Test Without Requirements

Explore, talk to stakeholders, compare similar systems.

## Tight Deadlines

Test most critical features first.

## Login Page Testing

Check valid/invalid inputs, UI, security.

## Payment Gateway Testing

Check success, fail, network issues.

## Accessibility Testing

Check for screen readers, keyboard navigation.

## Mobile App Testing

Check on real devices and emulators.

## Cloud Testing

Check scalability, security, global use.

## API Manual Testing

Use Postman to send/receive requests manually.

## Data Migration Testing

Compare old vs new data after move.

## Regression vs Retesting

Regression = old features. Retesting = fixed defect.

## Severity vs Priority Example

Severity = impact, Priority = urgency.

## Boundary Value Example

If 18–60 valid, test 17,18,60,61.

## Entry Criteria

Conditions needed before starting testing.

## Exit Criteria

Conditions to stop testing.

## Exploratory vs Ad-hoc

Exploratory = structured. Ad-hoc = random.

## Compliance Testing

Check app meets laws/regulations.

## Future of Manual Testing

More focus on usability and domain expertise.