

# QA Terminologies

## Test Execution, Environment & Automation

Test Execution	Running test cases and capturing results
• Test Report	Summary of coverage, defects, risks & quality insights
• Test Environment	Tools, data, infrastructure & configurations used for testing
• Test Automation	Scripts/tools that execute tests automatically
• CI/CD Pipeline	Automated code integration, build, test & deployment flow
• Test Data Management	Processes to create, manage and maintain data used in testing
• Mocking/Stubbing	Faking services/components for isolated testing
Code Coverage	% of code executed during automated tests
• Static Testing	Code/review validation without executing program
• Dynamic Testing	Testing by executing the application
• Shifting Left	Performing testing earlier in the development cycle
• Shifting Right	Testing in production-like or live environments

## Bug / Defect Terminologies

Defect Life Cycle	Stages a defect goes through from discovery to closure
• Severity	Impact level of a defect on functionality
• Priority	Urgency with which a defect must be fixed
• Root Cause Analysis	Identifying the underlying reason for a defect
• Bug Triage	Review and prioritization of defects
• Defect Density	Defects relative to module/code size
• Defect Leakage	Defects that escape to later stages (e.g., QA → Production)
Defect Rejection	When reported defect is considered invalid/not a bug
• Defect Duplication	Two or more logged defects represent the same issue
• Retesting	Verifying that a fixed defect is resolved
• Verification	Confirmation that defect fix meets requirements

## Miscellaneous

Exploratory Testing	Simultaneous learning, test design & execution
• Test Oracle	Source that determines whether output is correct
• Flaky Test	Test that fails intermittently without consistent reasons
• Build Verification Test (BVT)	Another term for smoke tests in some companies
• Benchmark Testing	Comparison with industry or internal performance benchmarks
• Reliability Testing	Ensures system consistency over time
• Observability	Ability to understand internal system state through logs/metrics
SIT/UAT	System Integration Testing / User Acceptance Testing

## Tools & Apps

Test Management / ALM	Jira, Azure DevOps (ADO), Micro Focus ALM/Quality Center, TestRail, Zephyr, qTest
• Defect Tracking	Jira, Bugzilla, Mantis, Redmine, YouTrack
• Version Control	Git, GitHub, GitLab, Bitbucket
• CI/CD & Build Pipelines	Jenkins, GitLab CI, GitHub Actions, Azure DevOps Pipelines, Bamboo, TeamCity, CircleCI
• UI Test Automation	Selenium, Playwright, Cypress, WebdriverIO, TestCafe, Robot Framework
• API Testing / Automation	Postman, Rest Assured, Karate, SoapUI, K6 API, Swagger/Insomnia
• Performance / Load Testing	JMeter, LoadRunner, Gatling, K6 Performance, Locust
Mobile App Testing	Appium, Espresso, XCUITest, BrowserStack, Sauce Labs
Security Testing	Burp Suite, OWASP ZAP, Nessus, Acunetix
• Test Data / Mocking / Virtualization	MockServer, WireMock, Hoverfly, Parasoft Virtualize
• Observability / Debugging	Kibana, Grafana, Splunk, Datadog, New Relic, CloudWatch
• Code Quality & Static Analysis	SonarQube, Checkmarx, Veracode, ESLint/PMD/FindBugs

## Types of Testing

Functional Testing	Ensures features behave according to requirements
• Non-Functional Testing	Validates performance, security, usability, reliability
• Integration Testing	Verifies interaction between modules/APIs/services
• System Testing	End-to-end validation of the full application
• Smoke Testing	Basic build stability check before deeper testing
• Sanity Testing	Quick validation after fixes/changes
• Regression Testing	Re-testing to ensure no side effects of new changes
Performance Testing	Measures speed, load, scalability, and responsiveness
• Security Testing	Detects vulnerabilities and data risks
• Usability Testing	Evaluates UX for clarity, ease of use & accessibility
• Compatibility Testing	Checks app behavior across browsers, OS, devices
• Accessibility Testing	Ensures application usage for differently-abled users
Entry/Exit Criteria	When testing can begin and when it can be closed

## Test Design & Planning

Test Scenario	High-level idea of what needs to be validated
• Test Case	Step-by-step instructions + expected outcome to verify behavior
• Test Suite	Group/collection of related test cases
• Test Data	Input values used during test execution
• Test Plan	Approach, scope, timelines, and resources for testing
• Test Coverage	% of requirements/features tested
• Traceability Matrix	Mapping between requirements and test cases to ensure coverage

## FYI Common QA Abbreviations

QA	Quality Assurance	Ensures processes to deliver quality software
• QC	Quality Control	Validates product accuracy via testing activities
• SQA	Software Quality Assurance	Broader set of engineering and process quality checks
• SDLC	Software Development Life Cycle	Phases from requirement → deployment
• STLC	Software Testing Life Cycle	Phases from test planning → closure
• SRS	Software Requirements Specification	Requirements document
• FRD/BR D	Functional/Business Requirements Document	Describes system functionality/business rules
BVA	Boundary Value Analysis	Test technique focusing on boundary limits
ECP/EQP	Equivalence Partitioning	Input grouping to reduce test cases
• RCA	Root Cause Analysis	Finding why a defect occurred
• RFE	Request For Enhancement	Suggested improvement, not a defect
• UAT	User Acceptance Testing	Final testing by end users/business
SIT	System Integration Testing	Testing combined components/systems
• BVT	Build Verification Test (Smoke Test)	Initial build stability check
• CI/CD	Continuous Integration / Continuous Delivery/Deployment	Pipeline for automated build/test/release
• TDD	Test-Driven Development	Tests written before implementation
• BDD	Behavior-Driven Development	Test style using business-readable specs (Given/When/Then)
• POM	Page Object Model	Selenium/automation framework design pattern
• API	Application Programming Interface	Interface for app-to-app communication
JSON/XML	JavaScript Object Notation / Extensible Markup Language	Common data formats for APIs
KPI	Key Performance Indicator	Metric used to evaluate quality goals
• DRE	Defect Removal Efficiency	% of defects detected before release
• NFR	Non-Functional Requirement	Performance, security, usability, etc.
• MTTR	Mean Time To Resolve/Repair	Average time to fix a defect/incident
MTBF	Mean Time Between Failures	Product/system reliability metric