

QA Engineer English Essentials

Must-know terms and phrases for QA engineers — test types, bug reports, automation, test plans.

<https://coderslingo.com/resources/cheatsheets/qa-engineer-essentials/>

Test types

unit test — Tests a single function or class in isolation.

integration test — Tests that several components work correctly together.

end-to-end (e2e) — Tests a full user flow through the whole system, like a real user.

smoke test — A quick check that the most critical paths work after a build or deploy.

sanity test — A narrow check that a specific fix or feature behaves as expected.

regression test — Re-running tests to confirm a change didn't break existing behaviour.

load test — Checking how the system behaves under expected traffic.

stress test — Pushing the system past its limits to find the breaking point.

performance test — Measuring speed, latency and resource use under defined conditions.

exploratory testing — Unscripted, hands-on testing to discover unexpected issues.

manual testing — A human checking behaviour by hand, without automation.

acceptance test — Verifying the feature meets the agreed business requirements.

accessibility test — Checking the product is usable with assistive technology.

security test — Probing for vulnerabilities and unsafe behaviour.

Bug lifecycle

defect / bug — A flaw causing the software to behave incorrectly.

reproduce — To make a bug happen again by following specific steps.

steps to reproduce — The exact sequence that triggers the bug, written for others to follow.

severity — How bad the impact is if the bug occurs.

priority — How urgently it should be fixed relative to other work.

blocker — A bug serious enough to stop a release or further testing.

flaky test — A test that passes and fails unpredictably without code changes.

triage — Reviewing new bugs to assign severity, priority and owner.

regression — A bug that breaks something that used to work.

expected vs actual — What should happen versus what actually happened — the heart of a bug report.

workaround — A temporary way to avoid the bug until it's properly fixed.

cannot reproduce — A status meaning the bug couldn't be triggered as described.

duplicate — A bug already reported elsewhere.

verify / close — Confirming a fix works, then closing the bug.

Test automation

test case — A defined input and expected result that checks one behaviour.

test suite — A grouped collection of related test cases.

assertion — A statement that a result must be true, or the test fails.

selector / locator — How an automated test finds an element on the page.

fixture — Reusable setup data or state prepared before a test runs.

mock — A fake stand-in for a real dependency during a test.

stub — A canned response returned in place of a real call.

test runner — The tool that executes tests and reports results.

coverage — How much of the code is exercised by the tests.

CI integration — Running the test suite automatically on every change.

headless — Running a browser test with no visible UI window.

test data — Inputs prepared specifically for testing scenarios.

teardown — Cleanup that runs after a test to reset state.

Test design

acceptance criteria — The conditions a feature must meet to be considered done.

happy path — The expected flow when everything goes right.

edge case — An unusual or extreme input that may not be handled.

boundary value — A value right at the limit of what's allowed (0, max length).

equivalence class — A group of inputs expected to behave the same, tested with one example.

negative test — Checking the system rejects invalid input gracefully.

test plan — A document describing scope, approach and what will be tested.

test scenario — A high-level description of what to test, before detailed cases.

precondition — The state that must exist before a test can run.

traceability — Linking tests back to the requirements they verify.

Key phrases used at work

"I can reproduce this consistently on the latest build — steps are in the ticket."

"Marking this a blocker: it stops the checkout flow on production data."

"Expected: the form shows a validation error. Actual: it submits with an empty email."

"This is a regression — it worked in 2.3 but broke after the last release."

"The test is flaky; it fails about one run in five, likely a timing issue."

"I'd set severity high but priority medium — it's nasty but rare."

"Could you add the browser, OS and exact steps so I can reproduce it?"

"I can't reproduce this — can you share a screen recording or a HAR file?"

"The happy path is covered; I'm now writing negative and boundary cases."

"These acceptance criteria are ambiguous — what's the expected behaviour when the cart is empty?"

"Closing as a duplicate of TICKET-481; let's track the fix there."

"There's a workaround, so I'd hold the release rather than block it."

"Coverage on this module is low — let's add tests before we refactor."

"The assertion is too loose; it passes even when the value is wrong."

"I've mocked the payment service so the e2e test doesn't hit the real API."

"Smoke tests are green on staging — clearing this for release."

"Verified the fix on staging; the issue no longer reproduces. Closing the bug."

"Let's add this scenario to the regression suite so it can't silently break again."

"Yesterday: automated the login suite. Today: triaging the new bug backlog. No blockers."

"Quick question before I sign off: is this edge case in scope for this release?"

