

# Software testing

## Unit tests

(component tests)

*White-box tests to cover edge cases, code branches.*

**Covers:** Code paths, individual components (functions).

**How?** Test parameters, pre/post conditions, state changes, output. Mock dependencies.

**Written by:** Developer.

**Ideally written** Before function code; **Last chance:** Before commit (while building).

**Key tech:** xUnit, mock objects (PHPUnit, RSpec, QUnit, Jasmine, Unit.js, Hamcrest).

## Integration tests

(integration and testing, I&T)

*Grey-box tests to cover interactions between objects/interfaces.*

**Covers:** Success/failure states, component interactions.

**How?** Test subsystem inputs/outputs. Mock dependencies, shared resources, IPC.

**Written by:** Developer.

**Ideally written** Before object code; **Last chance:** Before merging feature-branch (while building).

**Key tech:** xUnit, mock objects.

## Acceptance tests

(conformance tests, behaviour tests, functional tests, UAT, validation testing)

*Black-box tests to cover user paths through the system.*

**Covers:** Functions correctly, can complete tasks, meets client's requirements.

**How?** Look at user story/ticket, write steps to complete.

**Written by:** PO/Client/Tester/Developer.

**Ideally written** Before UI; **Last chance:** Before merging feature branch.

**Key tech:** Gherkin, SimpleTest, manual tests (Quail, Behat, Selenium; a11y/browser tests).

## Refinement tests

(UX tests, usability tests, business-value tests, ROI tests, end-user tests, field tests)

*Black-box tests to determine if the proposed UI is useful, usable, aesthetic, identifiable, inspirational and valuable.*

**Covers:** End-user experience.

**How?** Present design to users, gather data on how they experience it.

**Written by:** Designer/UX expert.

**Ideally written** After 1st design proposal; **Last chance:** Before finalizing design, after release.

**Key tech:** Visual diff tools, A/B testing frameworks, observation (Quail, Acquia Lift, PhantomJS, Selenium).

## System testing

(end-user tests, field tests, regression tests)

*Black-box tests to cover the system as a whole and catch regressions.*

**Covers:** The system as a whole.

**How?** Regression, performance, load/stress/volume, compatibility, recovery tests.

**Written by:** Tester/Developer/PO.

**Ideally written ; Last chance:** After integrating code, before demo.

**Key tech:** Continuous integration tools, manual tests (New Relic).

Coverage scope, Run-time, Effort, Cost

Number of tests