UnCodemy

Software Testing Syllabus PDF

A complete software testing syllabus moves from testing fundamentals and test design to defect management, SQL and API testing, then automation with Selenium or Playwright, CI/CD and projects. This page lists the topics module by module with the skill each one builds, so you can compare it with any course outline.

Key takeaways

- A good syllabus starts with fundamentals and test design before any tool.
- Manual testing, SQL and API testing are core for most roles.
- Automation is taught after the basics, using one tool in depth.
- Projects and interview preparation should be part of the plan.
- Check that the tools and versions listed match what employers use.

What is software testing syllabus?

A software testing syllabus is the ordered list of topics a learner studies to become a tester, covering manual testing, test design, tools, automation and the way testers work in Agile teams.

What does the syllabus cover?

The syllabus below is a typical structure for a complete testing programme. Duration depends on the course, so it is not stated here. Topic names are standard, and a particular institute may add or reorder modules.

What is Module 1: testing fundamentals?

What testing is, errors, defects and failures, the seven principles, verification and validation, test levels (unit, integration, system, acceptance), test types (functional, non-functional, regression, smoke, sanity) and the roles of tester and developer.

Module 2: SDLC, STLC and Agile

Software development models (waterfall, V-model, Agile), the test life cycle, Scrum roles and ceremonies, sprint testing and the definition of done.

Why is Module 3, test design, central?

Requirement analysis, test scenarios, test cases, equivalence partitioning, boundary value analysis, decision tables, state transition testing, error guessing and exploratory testing. This

module is the heart of manual testing.

Module 4: defect management

Defect life cycle, severity versus priority, writing a good bug report, defect triage, test execution reports and metrics, plus tools such as Jira.

Module 5: databases and SQL for testers

SELECT queries, filtering, joins and grouping to validate data after a test, checking inserts and updates and basic data integrity checks.

Module 6: API testing

HTTP basics, REST, status codes, JSON, testing with Postman, writing assertions, environments and collections, and an introduction to API automation.

What does Module 7, web automation, teach?

Locators and XPath or CSS, waits, page objects, test data, reporting and running tests on several browsers. Learn one tool well, such as Selenium with Java or Playwright with TypeScript, and understand the other at an overview level.

Module 8: framework and CI/CD

Framework design, assertions and test runners, version control with Git, running tests in a pipeline, and reading reports.

Module 9: non-functional and mobile testing

Performance testing concepts, security basics, accessibility checks, compatibility and an introduction to mobile app testing.

Module 10: AI in testing (optional)

Using AI tools to generate test ideas and data, their limits and how to review their output. This module is optional and changes quickly.

Module 11: projects and career preparation

Two end-to-end testing projects with test plan, test cases, defect logs and a summary report, a resume and interview practice.

How should I compare syllabi?

Look for depth in test design and defect handling, hands-on tool practice, real projects, current tool versions and a clear plan for practice. Be careful of syllabi with a very long tool list and little

practice time.

How long does each module take?

Duration depends on the course format, the learner's background and practice time, so no fixed number applies to every programme. As a rough guide, fundamentals and test design take longer than tool modules because they need repeated practice with examples. Ask any institute for the hours of hands-on practice per module, not only the total course length.

What skills should I be able to show after the syllabus?

You should be able to read a requirement and derive test scenarios, write clear test cases, log a defect that a developer can reproduce, query a database to check results, test an API in Postman, write a small automated test and explain your work in an Agile team. Keep proof of each, such as a test case sheet and a small repository.

How should I revise the syllabus?

Make a one-page map of the modules, mark your confidence from one to five and spend the next week on the lowest scores. Re-test yourself with a short exercise for each module.

What tools appear across the modules?

Common tools are Jira for defects, Postman for APIs, a database client for SQL checks, Selenium or Playwright for web automation, Git for version control and a CI tool to run tests. Tool lists change, so choose a syllabus that teaches the ideas behind each tool and not only the buttons.

What projects should the syllabus include?

At least one web application project covering requirement analysis, test scenarios, test cases, execution, defect reports and a summary report, and one API or automation project with a small framework. Ask to see sample project work from a course before you decide.

How do certifications fit in?

Certifications such as ISTQB Foundation can add structure to learning and help some resumes. They are optional and complement practical skills; check the official body for the current syllabus, exam rules and fees.

What prerequisites do I need?

No programming background is required for the manual modules. Basic computer skills and clear written English help, since test cases and bug reports are written documents. A little logic and curiosity about how software fails are the most useful traits for a tester.

Frequently asked questions

What topics are in a software testing syllabus?

Fundamentals, SDLC and Agile, test design, defect management, SQL, API testing, web automation, CI/CD, non-functional testing and projects.

Should manual testing come before automation?

Yes. Test design and defect handling are the base, and automation is easier to learn after them.

Which automation tool should I learn?

Learn one deeply, such as Selenium or Playwright, based on what your target employers use.

Is SQL needed for testers?

Yes. SQL helps you verify that data was saved correctly after a test.

Does this syllabus show course duration?

No. Duration varies by course, so compare it with the actual outline of the programme you consider.

Sources and further reading

- ISTQB
- Selenium documentation
- Playwright documentation

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550