UnCodemy

Software Testing Cheat Sheet PDF

A software testing cheat sheet condenses the essentials: the four test levels, functional and non-functional types, black box and white box techniques, test case and defect fields, severity versus priority, SDLC versus STLC, and entry and exit criteria. Use it for revision, but learn each idea with examples first.

Key takeaways

- The four classic levels are unit, integration, system and acceptance testing.
- Functional testing checks what the system does; non-functional checks how well it does it.
- Black box techniques include equivalence partitioning and boundary value analysis.
- Severity is technical impact; priority is how soon to fix.
- Verification asks if we built it right; validation asks if we built the right thing.

What is software testing quick reference?

A software testing cheat sheet is a compact reference that lists common testing terms, levels, types, techniques and document fields so that learners and testers can recall them quickly.

Which levels of testing should I remember?

Level	Focus	Usually done by
Unit	One function or class	Developers
Integration	Interaction between modules	Developers and testers
System	The complete application	Testers
Acceptance	Fit for business use	Users, product owners
Remember the order from small to large. Each level finds different kinds of defects.		

Which types of testing are most common?

Functional types include smoke, sanity, regression, retesting and exploratory testing. Non-functional types include performance, load, stress, security, usability, compatibility and

accessibility. Maintenance testing happens after changes in a live system. Know a one-line definition and an example for each.

How do black box and white box differ?

Black box testing uses requirements and behaviour without looking at code; examples are equivalence partitioning, boundary value analysis, decision tables and state transition testing. White box testing uses the structure of the code, for example statement and branch coverage. Grey box mixes both views, such as testing with knowledge of the database design.

What fields does a test case have?

A test case usually has an ID, title, preconditions, test data, steps, expected result, actual result, status and notes. Good test cases are clear, repeatable and independent. A test scenario is a higher-level statement, such as "verify login", from which several test cases come.

What fields does a defect report have?

Include ID, title, environment, build version, steps to reproduce, expected result, actual result, severity, priority, attachments and status. A clear title and reliable steps matter most, because developers need to reproduce the issue quickly.

What is severity versus priority?

Term	Meaning	Example
Severity	Impact on the system	Crash on payment: high
Priority	Urgency to fix	Typo on home page before launch: high
A high severity defect can have low priority if it affects a rarely used path, and a low severity defect can have high priority if it hurts the brand.		

How do SDLC and STLC relate?

The software development life cycle covers requirement analysis, design, coding, testing, deployment and maintenance. The software testing life cycle covers requirement review, test planning, test case design, environment setup, execution, defect reporting and closure. Testing activities run in parallel with development, not only at the end.

What are entry and exit criteria?

Entry criteria are conditions to start a phase, such as a stable build and approved test cases. Exit criteria are conditions to stop, such as all critical tests executed and no open high-severity defects. Teams define them in the test plan.

What are the seven testing principles?

Testing shows the presence of defects, not their absence; exhaustive testing is impossible; test early; defects cluster; the pesticide paradox means repeated tests lose power; testing is context dependent; and absence of errors is a fallacy if the system does not meet needs.

How do I use this sheet well?

Write each table from memory, check against the sheet and fix gaps. Then apply each idea to a real application, such as writing five test cases for a login page.

How do I turn terms into examples?

For each term on the sheet write one example from a familiar app. Smoke testing could be opening a shopping app and checking that it launches, shows products and lets you log in. Regression testing could be re-running checkout after a coupon change. Boundary value analysis could be testing a quantity field with 0, 1, 99 and 100 if the limit is 99. Examples make terms stick and help in interviews where you must explain ideas rather than only name them.

What should I add to the sheet over time?

Add the tools you use, such as your bug tracker, test management tool and automation framework, along with shortcuts and conventions followed by your team. Add three mistakes you made and how you fixed them. Add a list of questions to ask when you receive a new feature, such as who uses it, what could go wrong and how will we know it works. A living cheat sheet becomes a personal guide to your own practice.

What is a quick last-day revision routine?

Read the sheet once, close it and write the levels, types and defect fields from memory, then check. Spend remaining time on the two topics you missed most often and explain each aloud with an example.

Frequently asked questions

What should a software testing cheat sheet include?

It should include test levels, types, techniques, test case and defect fields, severity versus priority, SDLC and STLC and the testing principles.

What is the difference between verification and validation?

Verification checks whether we built the product right against specifications, while validation checks whether we built the right product for user needs.

What is the difference between severity and priority?

Severity is the technical impact of a defect, while priority is how soon it should be fixed according to business needs.

What are the levels of testing?

The usual levels are unit, integration, system and acceptance testing, moving from small parts to the whole product and user readiness.

Is a cheat sheet enough for an interview?

No. It helps recall, but interviewers ask for examples and reasoning, so practise explaining each term with a real scenario.

Sources and further reading

- ISTQB glossary and resources
- ISTQB glossary

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