

# QA Testing Methodologies: Cheat Sheet

10+ must-know key words to help advance your skillset



## In this e-guide

Words to go: QA testing methodologies p. 2

Further reading p. 7

## In this e-guide:

Do bottom-up and top-down testing throw you for a loop? In the dark about the difference between black box and functional testing?

Print out this handy cheat sheet of 10+ testing terms.

## In this e-guide

Words to go: QA testing methodologies p. 2

Further reading p. 7

## Words to go: QA testing methodologies

**Ryan Black**, Assistant Site Editor, SearchSoftwareQuality

A seemingly endless number of methods share the umbrella of software testing. In fact, sometimes, IT professionals struggle with the vernacular of these QA testing methodologies, because terms like *component testing* and *unit testing* are fairly interchangeable.

Put simply, testing entails a large number of steps and, therefore, a large number of methods. Likewise, applications include a tremendous number of components to examine before release.

Additionally, events in the software testing space alter -- [or shift](#) -- the purpose or nature of each practice, as well as testers' roles. Automation, for example, changed who should perform certain tests, and [continuous practices](#) changed when and how testing factors into the application development process.

But some fundamental practices remain, and they serve as a great place to build your knowledge of QA testing methodologies. Learn how each kind of test fits into the bigger QA picture in a software's journey to eventual release.

## In this e-guide

Words to go: QA testing methodologies p. 2

Further reading p. 7

## Black box vs white box testing

**White box testing:** Any testing process described as a white, clear or glass box examines a piece of software's internal structure. In most cases, QA professionals design white box tests around an application's [source code](#). These tests put the app's architecture, not what the software does, under review.

**Black box testing:** These methods, also referred to as *behavioral testing*, focus entirely on the application's functionality. Black box processes neither look at the inner workings of software nor how a function is facilitated.

## QA testing methodologies and techniques

**Functional testing:** A kind of black box testing, this practice confirms whether or not an application works according to its design. Feature testing, a new form of functional testing, determines whether a [particular feature](#) works as it was intended when that aspect of the software integrates with the completed application.

**Nonfunctional testing:** These tests examine software's systems, as well as external behaviors tangential to an application's core purpose. A number of practices fall within this branch of testing, including:

## In this e-guide

Words to go: QA testing methodologies p. 2

Further reading p. 7

- [Performance testing](#), which measures various attributes about a piece of software while it runs and can find bottlenecks within a system;
- [Load testing](#), which purposefully subjects the application to an amount of work close to its limits;
- [Stress testing](#), which examines how well software operates under progressively worse conditions and whether it remains effective despite that worsening environment; and
- Other QA testing methodologies that assess the [usability](#), maintainability, portability and [reliability](#) of software.

**[Regression testing](#):** This term refers to tests that ensure changes don't break existing aspects of an application. By design, every new build or candidate for release has differences from past iterations, and even small tweaks can unintentionally disable functionality. Regression testing reduces that risk, and testers often use automation as a part of this method to make sure older features work correctly by rerunning previously scripted test scenarios.

## Test levels

Some debate surrounds what exactly is a testing level. However, [many experts](#) recognize four stages, with each subsequent process increasingly widening the scope and the degree of interconnectedness.

---

## In this e-guide

---

■ Words to go: QA testing methodologies p. 2

---

■ Further reading p. 7

**Unit testing:** Developers often perform this test, also known as *component testing*, because it happens so early in the code creation process. Unit testing narrows in on application modules to ensure each individual unit of code performs according to its design and to catch problems early on.

**Integration testing:** Once a module passes unit testing, it is time to evaluate larger components. Integration testing helps testers find the defects in how individual modules interact with one another. To accomplish this task, testers can perform integration testing in two possible ways:

- **Bottom-up:** Testers focus on lower-level components first and then move on to higher-level ones. However, keep in mind that, if your application's higher-level modules perform critical functions, this approach tests those components last. So, bugs within those modules might take longer to uncover and resolve.
- **Top-down:** As the name implies, this type of integration test examines higher-level modules at the beginning. The test uses **stubs** to simulate the behavior of lower-level modules, while modules important to crucial application functions receive tests first.

## In this e-guide

Words to go: QA testing methodologies p. 2

Further reading p. 7

**System testing:** After developers integrate the application, system testing takes all the components and tests them as a whole system, even though they were previously tested individually. At this stage, an application is almost ready for production, and this process checks if the code meets every requirement as a cohesive software before launch. This test level entails a number of nonfunctional tests, such as performance and load testing, to examine customer-facing aspects of the application. System testing is also considered a form of black box testing.

**Acceptance testing:** This is the final check on whether software is ready for deployment. QA testers must use [detailed acceptance criteria](#) because this level is the last step before end users experience the application. Since this test level is so close to product launch, it should include more than the [alpha tests](#) that are performed by internal engineers to validate if the software operates as expected; this phase entails white box testing. Perform [beta testing](#), also known as [user acceptance testing](#), as part of this process to enable an application's intended users to try out the software themselves and provide feedback. This end-user response could result in developer updates that reset the whole test process over again.

 **Next Article**

## In this e-guide

---

Words to go: QA testing  
methodologies p. 2

---

Further reading p. 7

## Further reading

---

### How can I course-correct my testing career path?

Don't get stuck in QA monotony. Show your technical knowhow and expand your horizons to boost your career -- even if it means throwing out the map.

[Keep reading >>>](#)

---

### 10 skills of highly effective software testers

While test automation has gained traction in recent years, a QA professional is still a better fit than robotic tools for many cases -- but only if the tester has these 10 skills.

[Learn more >>>](#)

Images: Fotalia

© 2019 TechTarget. No part of this publication may be transmitted or reproduced in any form or by any means without written permission from the publisher