**What is Test Coverage?**

***Simply put, coverage is “What are we testing and How much are we testing?”***

Test coverage helps monitor the quality of testing, and assists testers to create tests that cover areas that are missing or not validated.

Most teams base their coverage calculations on functional requirements alone. It is also fair because first and foremost an application should do what it is supposed to do. If not, its speed or security or ease of use – none of it matters.

However, if dedicated and independent [non-functional testing](https://www.softwaretestinghelp.com/what-is-non-functional-testing/) teams are working on performance, security, usability testing, etc., then they will have to track their requirements all the way to execution through test coverage analytics.

### Test Coverage and Code Coverage

Test coverage is often confused with Code Coverage. Even though the underlying principles are the same, they are two different things.

[Code Coverage](https://www.softwaretestinghelp.com/code-coverage-tutorial/) really talks about unit testing practices that have to target all areas of the code at least once and is done by developers.

Test Coverage, on the other hand, is [testing every requirement](https://www.softwaretestinghelp.com/how-to-test-software-requirements-specification-srs/) at least once and is obviously a QA team activity.

What really qualifies to be a covered requirement depends on the interpretation of each team.

**For example**, Some teams call a requirement covered if there is at least one test case against it. Sometimes, it is covered if at least one team member is assigned to it. Or, if all the test cases associated with it are executed.

* If there are 10 requirements and 100 tests created – when these 100 tests target all of the 10 requirements and don’t leave out any – we call this adequate testing coverage at the design level.
* When only 80 of the created tests are executed and target only 6 of the requirements. We say that 4 requirements are not covered even though 80% of testing is done. This is coverage statistics at an execution level.
* When only 90 tests relating to 8 requirements have assigned testers and the rest of them are not, we say the test assignment coverage is 80% (8 out of 10 requirements).

It is also important as to when to calculate coverage.

If you do this too early in the process, you will see a lot of gaps because things are still incomplete. So it is generally advised to **wait until the Last Build** i.e. Final Regression Build. This will give a correct coverage of the Tests performed for the given Requirements.

What are Testing Techniques : Types, Advantages & Disadvantages

Testing Techniques is the method applied to evaluate a system or a component with a purpose to find if it satisfies the given requirements. Testing of a system helps to identify gaps, errors, or any kind of missing requirements differing from the actual requirements. Testing techniques are the best practices used by the [testing](https://www.elprocus.com/automation-testing-test-process-and-its-types/) team to assess the developed software in regards to given requirements. These techniques ensure the overall quality of the product or software including performance, [security](https://www.elprocus.com/gsm-based-home-security-system-working-with-applications/), customer experience, and so on. This article gives the reader a basic understanding of testing techniques, types of testing techniques, applications, and advantages and disadvantages.

## What are Testing Techniques?

A book written by Kaner Bach Pettichordon on Testing Techniques describes that testing is a five-fold system for any testing that the user wants to do. They are

* **Testers** – Users who perform the testing
* **Coverage** – What components gets covered
* **Potential problems** – Reason for testing, is to find errors?
* **Activities** – The way you test or how you test
* **Evaluation** – Compare the results to know if the test is successful or unsuccessful

All types of testing involve the above five dimensions. The testing techniques enable the user to focus on one or more dimensions to achieve the result.

### Types of Testing Techniques

Based on the requirements of the software, a suitable testing technique is employed. Each testing technique offers various features and benefits to serve the purpose better.

Though there are several types of testing techniques available, we shall focus on Black box testing and White box testing.

#### Black Box Testing

Black box testing is a type of software testing, which checks for the functionality of a software or an application without knowing the design, internal components, or structure of an application to be tested. It is also referred to as Specifications-based testing.

The black box testing method is mainly used to find missing functions, performance errors, initialization errors, and errors while accessing the external database.

The testing techniques of black-box testing include

**Equivalence Partitioning** **–** In equivalence partitioning, the input data of an application to be tested into equal partitions. This technique ensures to cover each partition at least once.

**Boundary Value Analysis** **–** In boundary value analysis is a technique used in which the testing of an application is done using the boundary values.

**Cause-effect Graph** **–** In this type of testing technique, causes are the inputs of a program and effects as the outputs of the program. Here, a graphical representation is used to show the relationship between the input and output and the factors that impact the outcome

**Error Guessing** **–** The error guessing testing method utilizes the skills and experience of the tester to detect errors when tools fail to do.

**All-pairs Testing** – In this approach, the software is tested using a combinatorial method to test all the possible discrete combinations of the parameters involved.

#### White Box Testing

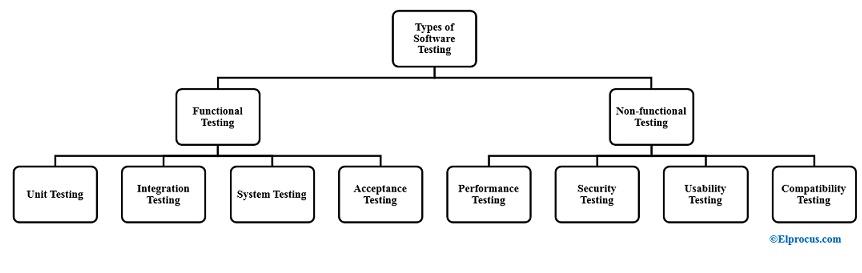
White box testing is a method of software testing that tests internal programming structures of an application. This type of testing technique is known as clear box testing, open box testing, structural testing, and transparent box testing. Its operation is opposite to black-box testing and is used at unit, integration, and system levels of the testing process.

The testing techniques of white-box testing include:

* **Statement Coverage** **–** In this technique, all programming statements are applied with a minimal number of tests.
* **Branch Coverage** **–** In this type of technique, all branches shall be tested by running them a sequence of tests.
* **Path Coverage** **–** All paths including statements and branches are tested using this technique.

## Types of Software Testing Techniques

Software testing is a method used to ensure that a software application is tested for software bugs and also checks if the developed software meets the specific requirements to produce a quality product. Software testing techniques are classified as shown in the figure below.

types-of-software-testing

### Functional Testing

Functional testing is an approach that is used to verify every functionality of the software and each function complies with the given requirement. Functional testing is divided into four types:

#### Unit Testing

In unit testing, each component or individual units of the software shall be tested. The aim of the unit testing is to check internal data structures, logic, boundary conditions for input and output data as per the design.

#### Integration Testing

In integration testing, individual units are integrated and tested to understand if the integrated components work efficiently.

#### System Testing

The purpose of system testing is to verify that all the system elements are tested and its overall function and performance comply with the specific requirements. In this approach, the system’s hardware and software components are integrated and tested as a whole.

#### Acceptance Testing

Is the developed software ready for delivery? This type of testing helps to identify if the application is ready for delivery and meets the business requirements. Alpha testing and Beta testing are the two types of acceptance testing.

### Non-functional Testing

The non-functional attributes of software such as performance, usability, security, reliability, and quality are tested in non-functional types of testing. The quality and performance of the software can be enhanced with non-functional testing. Various types of non-functional testing include:

#### Performance testing

Performance testing is performed to ensure that the software applications should be able to handle the workload well with increased performance. There are four kinds of performance testing, which include Load testing, Stress testing, Endurance testing, Spike testing.

#### Security testing

Security experts use the security level of testing to ensure that the system and application are protected from all types of loopholes. This testing provides security to the application and protects the loss of information.

#### Usability testing

The usability testing checks for usability and user-friendliness of the software. This test is performed to determine if the software is seamless to use by any user.

#### Compatibility testing

In this level of testing, compatibility of the software is tested for different [operating systems](https://www.elprocus.com/different-types-of-computer-operating-systems/), internet browsers, and so on. Such as an Android app is checked if it is compatible with different versions of Android OS.

### Advantages and Disadvantages of Testing Techniques

Software testing is an eminent tool and has a significant role in today’s business. Some of the foremost advantages are

* Highly efficient
* Quality
* Satisfies customer
* Good product, good revenue
* User experience
* Business optimization

Some of the disadvantages are:

* Appropriate communication and coordination with the tester
* Competition among similar service providers
* Lack of experienced professionals
* Finding the right service provider