# Test Types

classification based on specific test objectives

## Functional Testing

Evaluating functional quality characteristics

(completeness, correctness, appropriateness)

Sources of functional requirements:

- business requirements specifications;
- epics;
- user stories;
- use cases;
- functional specifications.

Considers the behavior of the software, so black-box testing techniques may be used to derive test conditions and test cases for the functionality of the component or system.

Measured through functional coverage using traceability between tests and functional requirements.

## Non-Functional Testing

Evaluating non-functional quality characteristics

(reliability, performance efficiency, security, compatibility, usability)

It is the testing of “how well” the system behaves.

Black-box testing techniques may be used to derive test conditions and test cases for non-functional testing.

For instance, boundary value analysis can be used to define the stress conditions for performance test.

Measured through non-functional coverage using traceability between, for instance, test cases and operational environments where the product is supposed to be used.

## White-Box Testing

Evaluating structure or architecture of component or system

(correctness, completeness, compliance with specifications)

Derives test based on the system’s internal structure or implementation.

Internal structure may include code, architecture, work flows, and/or data flows within the system.

Measured through structural coverage and code coverage.

At component testing level, code coverage may be based on the percentage of component code that has beed tested and may be measured in terms of different aspects of code (coverage items).

At component integration testing level, white-box testing may be based on the architecture of the system, such as interfaces between components. Structural coverage may be measured in terms of the percentage of interfaces covered by tests.

## Change-Related Testing

Evaluating the effects of changes

(confirming that defects are fixed, looking for side-effects of changes)

There are two types of change-related testing:

**Confirmation testing** is executed to confirm that the changes have corrected a defect. The software may be tested with all test cases that failed due to the defect and with new tests to cover changes needed to fix the defect.

The purpose is to confirm if the original defect has been successfully fixed.

**Regression testing** is executed to confirm that new functionality or changes to the environment (for instance, to OS or database management system) was implemented correctly, and have not caused any unforseen adverse consequences (regressions).

**Impact Analisys for Maintenance** (*in case of modification or migration*) – *additional activity* identifying the areas in the system that will be affected by the change.

## Test Levels

Should be performed at all test levels, though the focus is different at each level.

Can and often should be performed at all test levels, and done as early as possible.

Can be performed not only at component testing and integration testing levels, but also at system testing level.

It can test paths within a component, paths between units during integration, and between subsystems during a system testing.

Can be performed at all test levels, especially in iterative and incremental lifecycles (Agile).