



Types and types of testing





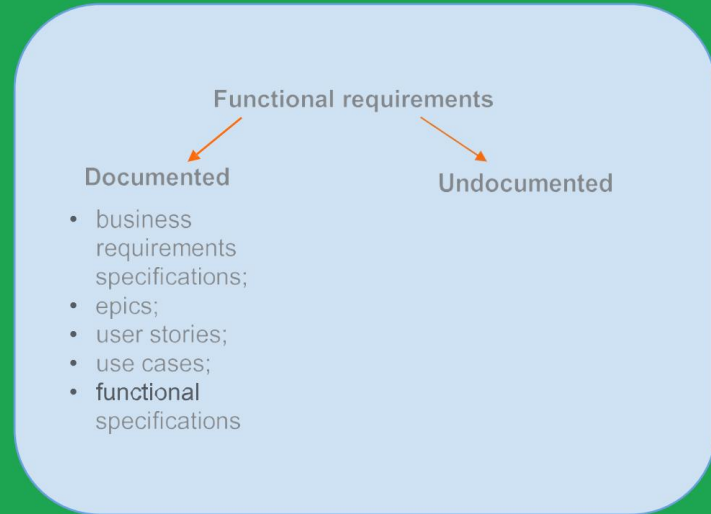
Functional testing

System functional testing includes tests that evaluate **the functions performed by the system. (= "does it work")**

Functional testing deals with specific behavior and is often called black box testing.

Functional testing should be performed at all levels of testing, although the emphasis at each level varies.

The development and execution of functional tests may require special skills or knowledge, such as knowing the specific business problem the software solves or the specific role it plays.



Non-functional Testing – non-functional testing

Non-functional tests evaluate **the characteristics of systems** and software. They answer the question "**how does the system behave**".

Non-functional testing, like functional testing, can and often should be done at all levels of testing and as early as possible.

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

Designing and executing non-functional tests may require special skills or knowledge, such as knowledge of vulnerabilities specific to a given project or technology (e.g. security vulnerabilities associated with specific programming languages) or a specific group of users (e.g. user profiles of facility management systems). health).



Security testing - security testing

is a type of tests that check how well the system is protected against unauthorized access to the system or data and various attacks.

- Security testing is based on six rules: •

- **confidentiality** - limited access to the resource for certain categories of users; • **integrity**

- control over consistency and precision data

- **authorization and authentication** - the system can only be changed in a certain way and by a certain group of users; • **non-repudiation** - no

- possibility of denying one's participation in all or part of the data exchange by one of the entities participating in this exchange; • **Availability** - Resources must be available to the user, internal authorized object, or device.



Interoperability Testing - interoperability testing

it is a type of testing that tests a program's ability to interact with one or more components or systems and includes compatibility testing and integration testing.



Performance(Efficiency) testing – performance testing



Types of performance tests

- **Load Testing** - focuses on the system's ability to cope with increasing levels of expected realistic loads from transaction requests created by a number of concurrent users or processes. You can measure and analyze the average response time of users in various common usage scenarios (operational profiles).
- **Stress testing** - is one type of performance testing. They check how the system/module behaves during the expected number of users, with limited or lack of resources such as: processor, memory or disk space.
- **Scalability Testing** - focuses on the system's ability to meet future performance requirements that may exceed current requirements. The purpose of the tests is to determine the ability of the system to develop (e.g. with more users, more data stored) without exceeding the current value of performance requirements or shortages.
- **Volume testing** - a type of performance testing performed while processing various (usually large) digital data.

Usability testing – usability tests

Usability testing - this is one of the UX research techniques that allows you to verify how users react to a given product, whether they can use it, what problems they encounter, what they think about a given solution.

Usability testing deals with software defects that affect the user's ability to perform tasks through the user interface. Usability issues can cause confusion, errors, delays, or complete failure for the user to complete a task.

Ease of use features: •

Understandable •

Learnable

• Practicality •

Attractiveness •

Protection against user errors • Availability



Maintainability testing – maintainability testing

More time in the software lifecycle is spent on support than on development.

To enable more efficient maintenance, **maintenance testing** is performed to measure the ease with which code can be analyzed, changed, and tested.

Typical maintainability testing objectives may include:

- Minimizing the cost of owning or operating the software; - Minimization of downtime required for software maintenance.

The maintainability of a system can be measured in terms of the effort required to diagnose problems identified in the system (analysability), make changes to the code (variability), and verify the changed system (testability).

Stability is related to the reaction of the system to changes. Systems with low stability show a large number of problems every time a change is made.

Portability testing – portability testing

Portability testing generally refers to how easily the software can be transferred to its intended environment (e.g. from an existing environment.). Example: from one operating system to another

Accessibility testing – accessibility testing

a testing process to determine the suitability of software for use by people with disabilities.



White-box Testing - white-box testing

White-box testing (or structure-based testing) is based on the internal structure of the system.

The internal structure may include:

- Code;
- Architecture;
- Workflows;
- System data streams.

White-box testing can be done:

- at the **component level**, where code coverage is based on a percentage component code that was tested;
- at the **integration level**, where testing may be based on the system architecture, such as the interface between components;
- on a **different level** in specific cases.

Designing white-box tests may require specialized skills or knowledge, such as knowing how to build code, how to secure data, and how to use tools and integrate their results correctly.



Regression testing - regression testing. It includes the creation of a set of test cases performed periodically, which are used to demonstrate that the system works properly after the change in the code.

Changes may also include changes to the environment, such as a new operating system or database management system.



Smoke testing - smoke test. The goal is to check the basic, most critical aspects of the system's operation. Based on the results of a given test, a decision is made as to whether it can proceed to the next stage and/or broader tests.



Sanity testing - conditional testing.

Narrowly focused testing sufficient to prove that a specific function works according to the requirements declared in the specification. It is used to determine the working capacity of a certain part of an additive after changes made to it or to its environment.



maintenance testing

Maintenance Testing - Tests that are performed during the system lifecycle stage after the system has been developed in the selected environment are called maintenance testing.

Maintenance testing focuses on testing system changes and testing unaltered parts that may have been affected by these changes.

It may be useful to test at several levels using different types of tests.

During the maintenance phase, it may be useful to test at several levels using different types of tests, depending on its capacity.

The range depends on:

- The degree of risk of change;
- The size of the existing system; • Change size.

ATTENTION

Maintenance testing differs from maintainability testing, which determines how easy it is to maintain a system.