**Software tests research**

**Student:** Jorn Kersten

**Student number:** 483331

**Class:** DB-S3-DB03

Tabel of contents

[What are software tests? 3](#_Toc124509431)

[Where do you need software tests for? 3](#_Toc124509432)

[Types of tests 3](#_Toc124509433)

[Unit tests 3](#_Toc124509434)

[Integration tests 3](#_Toc124509435)

[Functional tests 4](#_Toc124509436)

[Performance tests 4](#_Toc124509437)

[Security tests 5](#_Toc124509438)

[Acceptance tests 5](#_Toc124509439)

[User acceptance tests 6](#_Toc124509440)

[Regression tests 6](#_Toc124509441)

[Conclusion 7](#_Toc124509442)

[Sources 7](#_Toc124509443)

# What are software tests?

Software testing is an essential step in the software development process that helps ensure that the final product is of high quality, meets the requirements of its users, and is free of bugs. There are many different types of software tests that can be performed, each with their own specific purpose and benefits. In this research document, we will discuss the different types of software tests that are available, their purposes, and when to use them.

# Where do you need software tests for?

Software tests are used to ensure that a program or application functions as intended and to identify and fix any bugs or issues. They can be used at various stages of development, including during unit testing, integration testing, and acceptance testing. Additionally, they can be used to verify that the software continues to function correctly after updates or changes have been made.

# Types of tests

Before you start testing, it is useful to know what tests there are and which ones you should use in which context. The commonly used tests are:

1. Unit testing
2. Integration testing
3. Functional testing
4. Performance testing
5. Security testing
6. Acceptance testing
7. User acceptance testing
8. Regression testing

## Unit tests

Unit testing is a type of software testing that focuses on testing individual units or components of a software system. These units are typically small, self-contained portions of code, such as functions or methods. The goal of unit testing is to ensure that each unit of code behaves as expected and that there are no defects or bugs in the code. Unit tests are typically written by developers as they write the code, using a unit testing framework. These tests are usually automated and are run every time the code is changed or updated to ensure that changes do not break existing functionality.

Unit tests should test only the functionality of the unit being tested and should not have any dependencies on external resources or other units. This allows for tests to be run quickly and easily, making it easier to identify the source of any issues that arise.

## Integration tests

Integration testing is a type of software testing that focuses on testing how different components of a system work together. The goal of integration testing is to ensure that the interactions between different components of a system are working as expected and that there are no defects or bugs in the integration of the components.

Integration testing is typically done after unit testing and before system testing. It is done to check if the interfaces between the different components are working as expected, and to ensure that the system as a whole is working as expected.

Integration tests are usually automated and are run every time the code is changed or updated. The tests are designed to exercise the interfaces between the different components of the system, and they may involve testing data flows, message passing, or other forms of communication between components.

Integration testing can be done in various ways, such as:

* Top-down integration testing, where the top-level component is tested first and then lower-level components are tested as they are integrated.
* Bottom-up integration testing, where the lowest-level component is tested first and then higher-level components are tested as they are integrated.
* Big-Bang integration testing, where all the components are integrated and tested at once.
* Incremental integration testing, where components are integrated and tested in small batches.

## Functional tests

Functional testing is a type of software testing that focuses on testing the functionality of a system against its requirements. The goal of functional testing is to ensure that the system is working as expected and that all the required features and functions are working correctly.

Functional testing is usually done by the QA (Quality Assurance) team and is typically done after integration testing and before system testing. It is done to check if the system is working as per the requirements mentioned in the requirement specification document.

Functional tests are usually automated and are run every time the code is changed or updated. The tests are designed to exercise the various features and functions of the system, and they may involve testing data inputs, user interactions, and system outputs.

## Performance tests

Performance testing is a type of software testing that focuses on testing the performance and scalability of a system. The goal of performance testing is to ensure that the system can handle the expected load and usage, and that it can scale to meet the needs of the users.

Performance testing is typically done by the QA (Quality Assurance) team and is usually done after functional and system testing. It is done to check how the system behaves under different loads and usage scenarios, such as heavy traffic, high concurrent users, or large data inputs.

Performance tests are usually automated and can be run on a regular basis or as needed. They are designed to exercise the system under different loads and usage scenarios, and they may involve simulating real-world user interactions, network traffic, or other factors that can affect the performance of the system.

Performance testing can be done in various ways, such as:

* Load testing, which tests the system under normal and peak loads to check for performance bottlenecks or issues.
* Stress testing, which tests the system under extreme loads to check for the system's stability and ability to handle unexpected usage scenarios.
* Endurance testing, which tests the system over a prolonged period of time to check for memory leaks, performance degradation, or other long-term issues.

Performance testing helps to identify issues with the performance of the system, such as bottlenecks, memory leaks, or scalability issues. It also helps to ensure that the system can handle the expected load and usage, and that it can scale to meet the needs of the users.

## Security tests

Security testing is a type of software testing that focuses on testing the security of a system. The goal of security testing is to identify vulnerabilities or weaknesses in the system that could be exploited by attackers, and to ensure that the system is properly protecting sensitive data and resources.

Security testing is typically done by the QA (Quality Assurance) team and is usually done after functional and system testing. It is done to check the system's compliance with security standards and regulations, and to identify any potential security vulnerabilities or weaknesses.

Security tests are usually automated and can be run on a regular basis or as needed. They are designed to identify potential vulnerabilities or weaknesses in the system, such as those related to authentication, access control, data encryption, or network security.

Security testing can be done in various ways, such as:

* Penetration testing, which simulates an attack on the system to identify vulnerabilities or weaknesses that could be exploited by an attacker.
* Vulnerability scanning, which uses automated tools to identify potential vulnerabilities or weaknesses in the system.
* Compliance testing, which checks the system's compliance with security standards and regulations.
* Risk assessment, which evaluates the potential risks to the system and its data, and recommends measures to reduce or mitigate those risks.

Security testing helps to identify issues with the security of the system, such as vulnerabilities or weaknesses that could be exploited by an attacker. It also helps to ensure that the system is properly protecting sensitive data and resources, and that it is compliant with security standards and regulations.

## Acceptance tests

Acceptance testing is a type of software testing that focuses on determining whether a system meets the acceptance criteria of the client or end-user. The goal of acceptance testing is to ensure that the system meets the needs and expectations of the client or end-user, and that it is ready for release or deployment. Acceptance testing is typically done by the client or end-user, but can also be done by a QA (Quality Assurance) team or a representative of the client. It is usually done after functional, system, and integration testing and is the final phase of testing before the system is released or deployed.

Acceptance tests are usually defined by the client or end-user and are based on the acceptance criteria outlined in the requirement specification document. They are designed to ensure that the system meets the needs and expectations of the client or end-user, and that it is ready for release or deployment.

Acceptance testing can be done in various ways, such as:

* User acceptance testing (UAT), which is done by the end-users to ensure that the system is usable and meets their needs.
* Operational acceptance testing, which is done by the client's IT or operations team to ensure that the system can be integrated with their existing systems and can be operated and maintained.
* Contract acceptance testing, which is done by a representative of the client to ensure that the system meets the terms of the contract or agreement.

## User acceptance tests

User Acceptance Testing (UAT) is a type of software testing that focuses on determining whether a system is usable and meets the needs of the end-users. The goal of UAT is to ensure that the system is user-friendly, easy to use, and meets the requirements of the end-users. UAT is typically done by the end-users themselves, or by a representative of the end-users, such as a business analyst or a customer representative. It is usually done after functional, system, integration, and acceptance testing and is the final phase of testing before the system is released or deployed.

UAT tests are usually defined by the end-users and are based on their specific requirements and needs. They are designed to ensure that the system is user-friendly, easy to use, and meets the requirements of the end-users. UAT tests may include testing of the system's usability, functionality, and performance from the end-user perspective. UAT helps to ensure that the system is user-friendly, easy to use, and meets the requirements of the end-users. It also helps to identify any issues or gaps that may have been missed during the previous testing phases and to make sure that the system is ready for production use.

## Regression tests

Regression testing is a type of software testing that focuses on ensuring that changes to a system do not introduce new bugs or issues. The goal of regression testing is to verify that the system continues to function correctly after changes are made. Regression testing is typically done by the QA (Quality Assurance) team and is usually done after functional, system, integration, and acceptance testing. It is done to ensure that any changes made to the system, such as bug fixes or new features, do not affect the existing functionality of the system.

Regression tests are usually automated and are run every time the code is changed or updated. They are designed to exercise the existing functionality of the system and to ensure that it continues to work as expected after changes are made. Regression tests are often a subset of previously run tests that were selected based on the nature of the changes made to the system.

Regression testing can be done in various ways, such as:

* Full regression testing, where all the previously run tests are executed again.
* Partial regression testing, where only a subset of the previously run tests are executed again.
* Selective regression testing, where only the tests that are affected by the changes are executed again.

# Conclusion

In conclusion, Software testing is a crucial step in the software development process that helps ensure that the final product is of high quality, meets the requirements of its users, and is free of bugs. There are many different types of software tests that can be performed, each with their own specific purpose and benefits. By understanding the different types of tests that are available, software developers can make informed decisions about which tests to use and when to use them. The sources discussed in this essay provide a good starting point for learning more about software testing and the different types of tests that are available.

# Sources

<https://www.guru99.com/software-testing-introduction-importance.html>

<https://www.guru99.com/> (for finding info on tests)

<https://www.ibm.com/topics/software-testing>