

## 1.1 WHY DO WE TEST SOFTWARE?

Software testing today is most effective when it is continuous, indicating that testing is started during the design, continues as the software is built out, and even occurs when deployed into production. Continuous testing means that organizations don't have to wait for all the pieces to be deployed before testing can start. Shift-left, moving testing closer to design, and shift-right, where validation is done by end-users, are also philosophies of testing that have recently gained traction in the software community. Once your test strategy and management plans are understood, automation of all aspects of testing becomes a must to support the speed of delivery that is required.

### 1.1.1 Types of software testing

There are many different types of software tests, each with specific objectives and strategies:

- **Acceptance testing:** Verifying whether the whole system works as intended.
- **Code review:** Are an effective means to confirm that new and modified software is following an organization's coding standards and adheres to its best practices.
- **Integration testing:** Ensuring that software components or functions operate together.
- **Unit testing:** Validating that each software unit performs as expected. A unit is the smallest testable component of an application.
- **Functional testing:** Checking functions by emulating business scenarios, based on functional requirements. Black-box testing is a common way to verify functions.
- **Performance testing:** Testing how the software performs under different workloads. Load testing, for example, is used to evaluate performance under real-life load conditions.
- **Regression testing:** Checking whether new features break or degrade functionality. Sanity testing can be used to verify menus, functions and commands at the surface level, when there is no time for a full regression test.
- **Security testing:** validate that your software is not open to hacker or other malicious types of vulnerabilities that could be exploited to deny access to your services or cause them to perform incorrectly.

- **Stress testing:** Testing how much strain the system can take before it fails. Considered to be a type of non-functional testing.
- **Usability testing:** Validating how well a customer can use a system or web application to complete a task.

In each case, validating base requirements is a critical assessment. Just as important, exploratory testing helps a tester or testing team uncover hard-to-predict scenarios and situations that can lead to software errors.

Even a simple application can be subject to a large number and variety of tests. A test management plan helps to prioritize which types of testing provide the most value – given available time and resources.

Testing effectiveness is optimized by running the fewest number of tests to find the largest number of defects.

**There are many different types of software tests, each with specific objectives and strategies:**

- **Acceptance testing:** Verifying whether the whole system works as intended.
- **Code review:** Are an effective means to confirm that new and modified software is following an organization's coding standards and adheres to its best practices.
- **Integration testing:** Ensuring that software components or functions operate together.
- **Unit testing:** Validating that each software unit performs as expected. A unit is the smallest testable component of an application.
- **Functional testing:** Checking functions by emulating business scenarios, based on functional requirements. Black-box testing is a common way to verify functions.
- **Performance testing:** Testing how the software performs under different workloads. Load testing, for example, is used to evaluate performance under real-life load conditions.

- **Regression testing:** Checking whether new features break or degrade functionality. Sanity testing can be used to verify menus, functions and commands at the surface level, when there is no time for a full regression test.
- **Security testing:** validate that your software is not open to hacker or other malicious types of vulnerabilities that could be exploited to deny access to your services or cause them to perform incorrectly.
- **Stress testing:** Testing how much strain the system can take before it fails. Considered to be a type of non-functional testing.
- **Usability testing:** Validating how well a customer can use a system or web application to complete a task.

In each case, validating base requirements is a critical assessment. Just as important, exploratory testing helps a tester or testing team uncover hard-to-predict scenarios and situations that can lead to software errors.

Even a simple application can be subject to a large number and variety of tests. A test management plan helps to prioritize which types of testing provide the most value – given available time and resources. Testing effectiveness is optimized by running the fewest number of tests to find the largest number of defects.