Software Testing
Manual

Software testing is led by a team or individual who will manually operate a software product and ensure it behaves as expected.

Automated

Software testing is composed of many different tools which have varying capabilities, ranging from isolated code correctness checks to simulating a full human-driven manual testing experience.
Testing Approaches
Testing Approaches

1. White Box Testing
2. Black Box Testing
3. Grey Box Testing
White Box Testing

- Glass Box
- Clear Box
- Structural Testing

White Box Testing is based on applications internal code structure.

In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases.
Black Box Testing

- Behavioral / Specification-Based Testing
- Input-Output Testing

Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.
Grey Box Testing

Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.
Testing Levels
Unit tests are very low level, close to the source of your application.

They consist in testing individual methods and functions of the classes, components or modules used by your software.

Unit tests are in general quite cheap to automate and can be run very quickly by a continuous integration server.
Integration tests verify that different modules or services used by your application work well together.

For example, it can be testing the interaction with the database or making sure that microservices work together as expected.

These types of tests are more expensive to run as they require multiple parts of the application to be up and running.
End-to-end tests

Testing the fully integrated application this is also called as end to end scenario testing.

To ensure that the software works in all intended target systems.

Verify thorough testing of every input in the application to check for desired outputs.

Testing of the users experiences with the application.
Functional tests focus on the business requirements of an application.

They only verify the output of an action and do not check the intermediate states of the system when performing that action.

Testing all the functionalities by providing appropriate input to verify whether the actual output is matching the expected output or not.

It falls within the scope of black box testing and the testers need not concern about the source code of the application.
End-to-end tests

Testing the fully integrated application this is also called as end to end scenario testing.

To ensure that the software works in all intended target systems.

Verify thorough testing of every input in the application to check for desired outputs.

Testing of the users experiences with the application.
Acceptance tests are formal tests executed to verify if a system satisfies its business requirements.

They require the entire application to be up and running and focus on replicating user behaviors.

But they can also go further and measure the performance of the system and reject changes if certain goals are not met.
Performance tests check the behaviors of the system when it is under significant load.

These tests are non-functional and can have the various form to understand the reliability, stability, and availability of the platform.

Performance tests are by their nature quite costly to implement and run, but they can help you understand if new changes are going to degrade your system.
Smoke tests are basic tests that check basic functionality of the application.

They are meant to be quick to execute, and their goal is to give you the assurance that the major features of your system are working as expected.
Questions?

Next: Working with External APIs