What is Continuous Testing

Continous Testing is a process of evaluating the functionality of a software application to find any software bugs. It checks whether the developed software met the specified requirements and identifies any defect in the software in order to produce a quality product. It is basically executing a system in order to identify any gaps, errors, or missing requirements contrary to the actual requirements.

Types of Continous Testing

Testing is an integral part of any successful software project. The types of software testing depend on various factors, including project requirements, budget, timeline, expertise, and suitability. The different Types of software testing are the key role where the tester determines the right testing for the apps.

The two main kinds of testing performed by the QA or Software Tester are:

- Functional Testing
- Non-Functional Testing

What is Functional Testing?

Functional Testing is defined as a type of testing which verifies that each **function** of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application. It focuses on manual testing as well as automation testing.

Functional Testing involves checking of the following:

- User Interface
- APIs

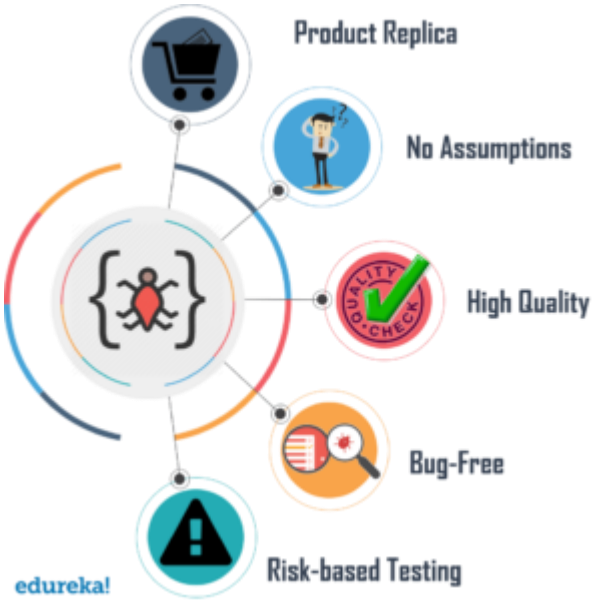
- Database
- Security
- Client/ server applications
- Functionality of the Application Under Test

Functional Testing vs Non-Functional Testing

Functional Testing defines the process of testing and Non-Functional Testing is used to check performance, reliability, usability etc.

Advantages of Functional Testing

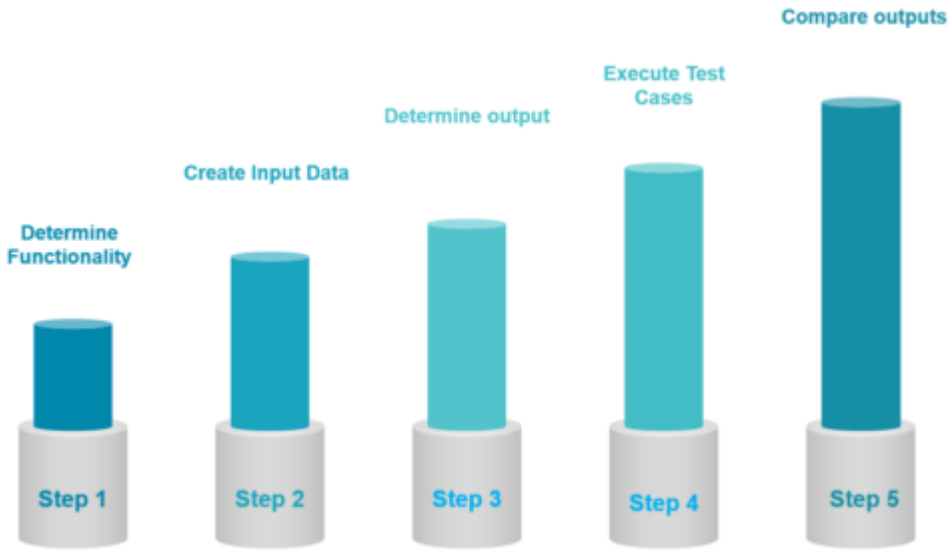
Functional testing is important as it verifies that the system is fixed for release and it is error free. Now let's have a look at some of the advantages of Functional Testing:



- This testing provides a **replica** of what the actual system is i.e. it is a replica of what the product is in the live environment. Testing is focused on the specifications as per customer usage.
- It does not work on any **assumptions** about the structure of the system.
- This testing ensures the delivery of a **high-quality** product which meets the customer requirement and makes sure that the customer is satisfied with the end results.
- It ensures to deliver a **bug-free** product which has all the functionalities working as per the customer requirement.
- **Risk-based** testing is done to decrease the chances of any kind of risk in the product.

Steps involved in Functional Testing

There are certain steps that we need to follow while performing functional testing. The various steps involved are:



- The first step is to determine the **functionality** of the product that needs to be tested and it includes testing the main functionalities, error condition, and messages, usability testing.
- Next step is to create the **input data** for the functionality to be tested as per the requirement specification.
- In the third step, the output is determined for the functionality under test from the requirement specification.
- Prepared **test cases** are executed in the next step.
- The final step is to **compare** the **output** after executing the test case and expected output in order to find whether the functionality is working as expected or not.

Types of Functional Testing

Functional testing has many categories and these can be used based on the scenario. Let's have a look at the most prominent types of functional testing:

1. **Unit Testing** – It is usually performed by a developer who writes different code units that could be related or unrelated to achieve a particular functionality. Code coverage is an important part of unit testing where the test cases need to exist to cover line coverage, code path coverage, and method coverage.
2. **Sanity Testing –** It is done to ensure that all the major and vital functionalities of the application/system are working correctly. This is generally done after a smoke test.
3. **Smoke testing –** It is done after each build is released to test in order to ensure build stability.
4. **Regression tests –** Testing performed to ensure that adding new code, enhancements, fixing of bugs is not breaking the existing functionality or causing any instability and still works according to the specifications.
5. **Integration tests** – When the system relies on multiple functional modules that might individually work perfectly, but have to work coherently when clubbed together to achieve an end to end scenario, validation of such scenarios is called Integration testing.

6. **Usability testing –** Product is exposed to the actual customer in a production like an environment and they test the product. The user's comfort is derived from this and the feedback is taken. This is similar to that of User Acceptance testing.

Tools for Functional Testing

The market is full of a number of automation tools for test management, load testing, GUI testing, functional testing, etc. I would suggest you opt for a tool which is on-demand, easy to learn as per your skills, generic and effective for the required type of testing.

Some of the most preferred tools for functional testing are:

- **Ranorex**
- **Selenium**
- **Test IO**
- **Telerik**
- **CUIT**

Selenium is one of the most preferred tools for testing. Let's have a look at how it is used for automation testing.