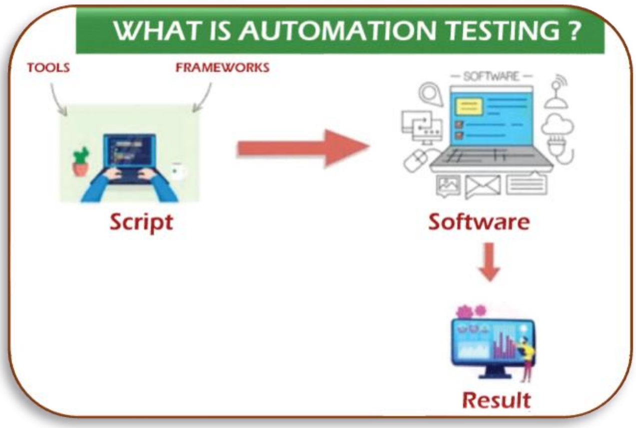
AUTOMATION TESTING

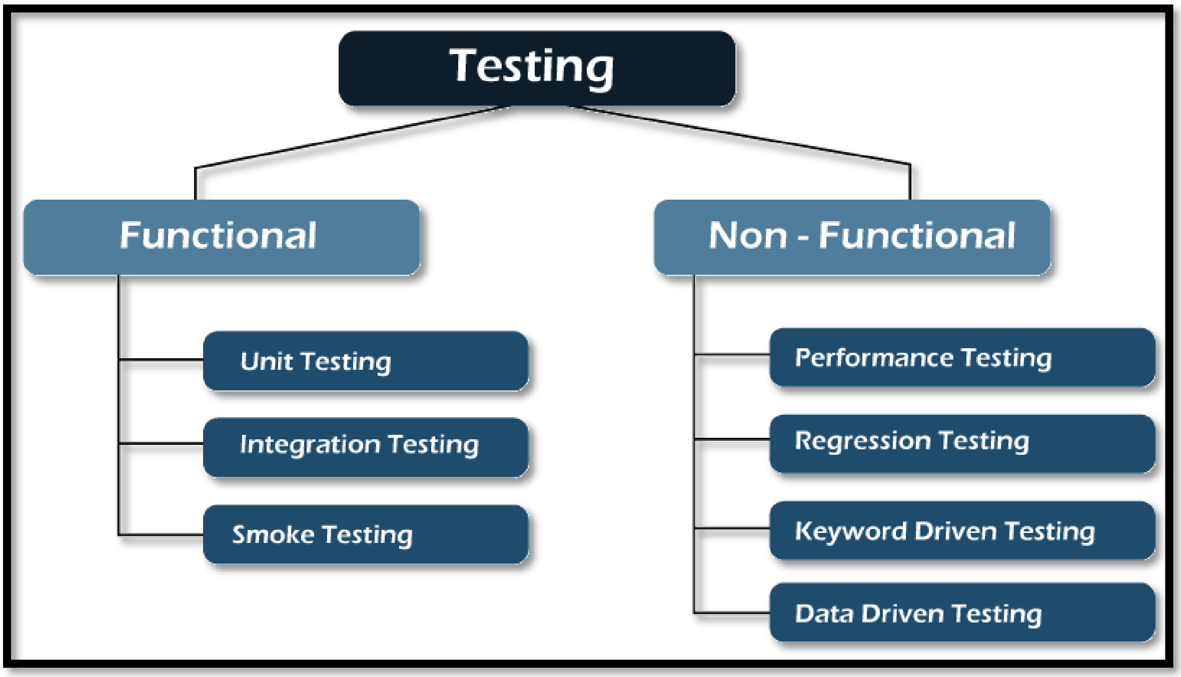
Automated Testing is a **technique** where the **Tester writes** scripts on their own and uses suitable Software or Automation Tool to test the software.

- It is **an Automation Process** of a Manual Process.
- It allows for executing repetitive tasks without the intervention of a Manual Tester.
- It is used to automate the testing tasks that are difficult to perform manually.
- Automation tests can be run at any time of the day as they use scripted sequences to examine the software.
- Automation tests can also enter test data compare the expected result with the actual result and generate detailed test reports.
- The goal of automation tests is to reduce the number of test cases to be executed manually but not to eliminate manual testing.
- It is possible to record the test suit and replay it when required.



❖ Automation testing types

Testing is grouped under **two types**: ***functional** and **non-functional***



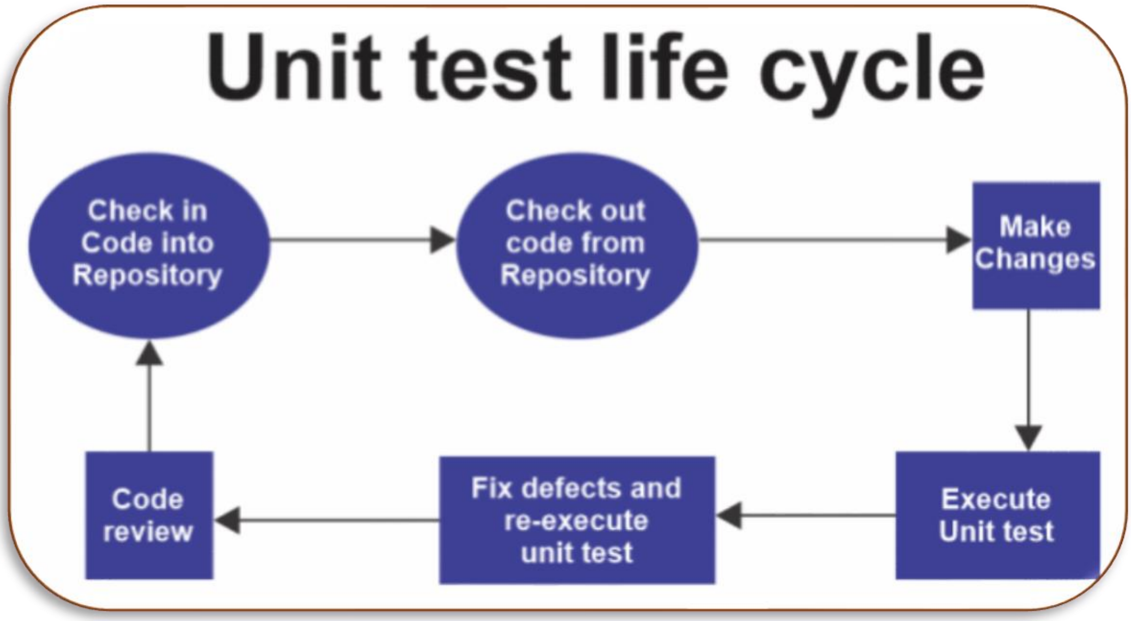
✧ Functional testing

- ✓ The first test performed by tester on newly revised software is called functional testing, which verifies all the software functions' features per user requirement.
- ✓ This testing works on the real-world business application and obtaining the expected output from a given input.
- ✓ All application functions are tested and involve smoke, unit, and integration testing.

a) Unit testing

- The unit is the smallest component of the software that functions individually.

- Unit testing simplifies the testing of the whole software, where each software element is fully tested before the final version is out.
- Unit testing depicts how the code performs at each part and has a faster execution time.
- It's the favourite of developers because it consumes less time and assure the working of each part of the software.
- Before automation testing, the developers write the code for testing, but now there is no need.
- The unit testing technique is divided into three broad categories: White box testing, Black box testing and Grey box testing.


Unit test life cycle
Check in Code into Repository
Check out code from Repository
Make Changes
Execute Unit test
Fix defects and re-execute unit test
Code review

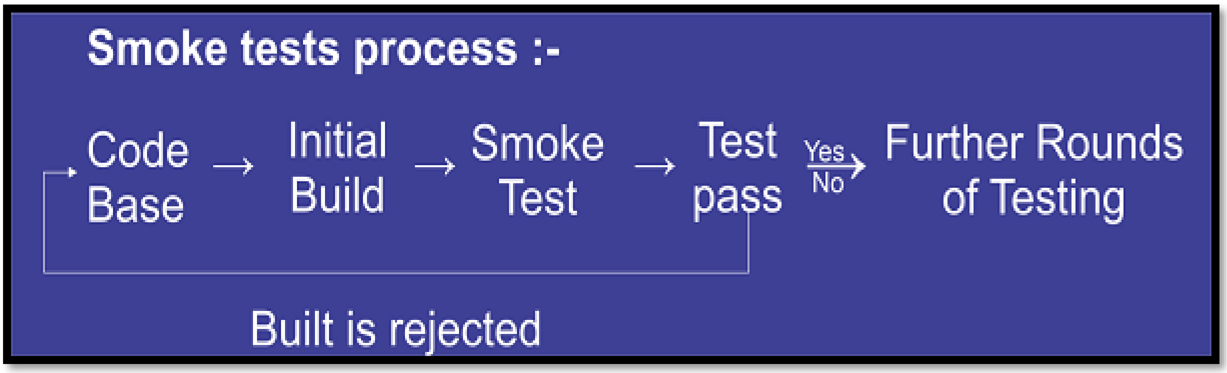
b) Integration testing

- Integration testing is more complicated to set up compared with other tests.
- All the modules of the application communicate with each other to perform tasks.

- Therefore, testers group them for testing and exposing the flaws in maintaining the interaction between these modules.
- Another name for this testing is I&T or string testing, considered end-to-end.

c) Smoke testing

- This testing checks and defines the product's stability (whether stable or not).
- If the product result is unstable, it is called an 'unstable build' and sent back to developers, where they run more test cases to find out the root cause of the problem.
- The smoke test works like this:-



✧ Non-functional testing

✓ Non-functional testing focuses on how well application functions are doing, not on what the product does.

✓ It is the opposite of functional testing, where application elements like reliability, usability, performance, etc., are tested.

✓ Some types of non-functional testing are reliability testing, load testing, compatibility testing, performance testing, security testing etc.

a) Performance testing

- This non-functional testing tests the software's stability, responsiveness and speed under the workload.
- It finds out the potential issues faced by critical software and medical programs used by the user, like slow operation of software under stressful circumstances.
- It finds hurdles in the performance of software and removes them to increase the ability of software to deliver the best results to the end user.

b) Regression testing

- When some changes are made to the code of software or application, it needs to be tested to determine whether the software is working as before the change; for this purpose, testers use automation regression testing to automate scripts, applications of workflows, plans and other activities.
- It tests the system or software workflow after its updation and functional error.

c) Keyword driven testing

- Keyword-driven testing tests the application using the data files consisting of the keywords related to the application, representing a set of actions needed to carry out the step.
- Here these specific keywords are identified and connected with the specific action.
- Therefore during testing, when these keywords are used, their related actions will automatically be done.

- This keyword testing is a popular choice for many businesses as it's flexible, concise, easy to maintain and reusable.
- Keyword-driven testing is compatible with all kinds of automation tools in the market.
- Instead of programming experts, functional testers can plan the testing before the application is fully developed.

d) Data-driven testing

- In data-driven testing, automation is inbuilt and very effective due to the few facilities provided, like the reusability of code, change in the script doesn't affect the test cases, and this testing can be carried out in the phase of the software development cycle.
- It provides consistency in results and reduces the investment of time and resources.
- Test cases use the data separately stored in the table or spreadsheet format, and testers have multiple data sets for testing.

❖ Automation Testing Process

1. Test Tool Selection:

There will be some criteria for the Selection of the tool. The majority of the criteria include: Do we have skilled resources to allocate for automation tasks, Budget constraints, and Do the tool satisfies our needs?

2. Define Scope of Automation:

This includes a few basic points such as the Framework should support Automation Scripts, Less Maintenance must be there, High Return on Investment, Not many complex Test Cases.

©Topperworld

3. Planning, Design, and Development:

For this, we need to Install particular frameworks or libraries, and start designing and developing the test cases such as NUnit, JUnit, QUnit, or required Software Automation Tools

4. Test Execution:

Final Execution of test cases will take place in this phase and it depends on Language to Language for .NET, we'll be using NUnit, for Java, we'll be using JUnit, for JavaScript, we'll be using QUnit or Jasmine, etc.

5. Maintenance:

Creation of Reports generated after Tests and that should be documented to refer to that in the future for the next iterations.

❖ Popular Automation Tools

1.Selenium: Selenium is an automated testing tool that is used for Regression testing and provides a playback and recording facility. It can be used with frameworks like JUnit and Test NG. It provides a single interface and lets users write test cases in languages like Ruby, Java, Python, etc.

2.QTP: Quick Test Professional (QTP) is an automated functional testing tool to test both web and desktop applications. It is based on the VB scripting language and it provides functional and regression test automation for software applications.

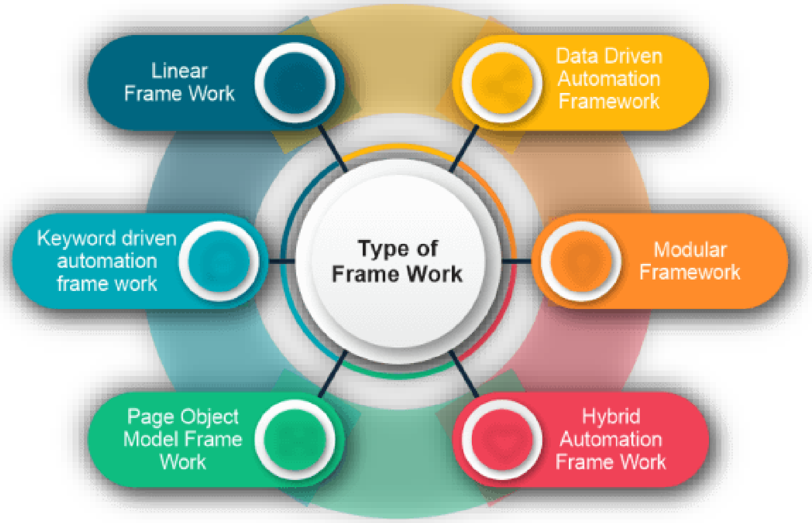
3.Sikuli: It is a GUI-based test automation tool that is used for interacting with elements of web pages. It is used to search and automate graphical user interfaces using screenshots.

4.Appium: Apium is an open-source test automation framework that allows QAs to conduct automated app testing on different platforms like iOS, Android, and Windows SDK.

5.Jmeter: Apache JMeter is an open-source Java application that is used to load test the functional behavior of the application and measure the performance.

❖ The framework used in automation software testing

Automation testing is executed on a few frameworks:


Linear Frame Work
Data Driven Automation Framework
Keyword driven automation frame work
Type of Frame Work
Modular Framework
Page Object Model Frame Work
Hybrid Automation Frame Work

◆ Linear Framework

One of the simplest frameworks that act as a record and playback model. In this tester write the simple code to run the test cases without sequential steps and modularity.

Data-driven automation framework

The data-driven automation framework can perform both negative and positive test cases. All the test case data inputs are stored in the extension files and tables from where values are read during the execution of test scripts.

Modular automation framework

A modular automation framework is best suited to run large test scripts as it divides the test scripts into independent modules that hierarchically interact with each other. These small independent modules are tested easily because it's easy to create required test scenarios.

Hybrid automation framework

Hybrid is always a combination; here, keyword-driven and data-driven frameworks are combined in which test data and keywords are externalized. Test data is stored in an excel or properties file, whereas keywords are maintained in a separate java file.

Page object model framework

In this framework, the tester doesn't need to write the code repeatedly because an object for the UI element is created that can be recalled later for testing. This feature of the POM framework results in less code usability and verbosity and reduces time consumption in writing test scripts.

◆ Keyword-driven automation framework

In KDF, the keywords are separated for a common set of functions and instructions, due to which automation speeds up. In this scripting technique, keywords are associated with actions like closing and opening a browser, mouse-click events, and others.