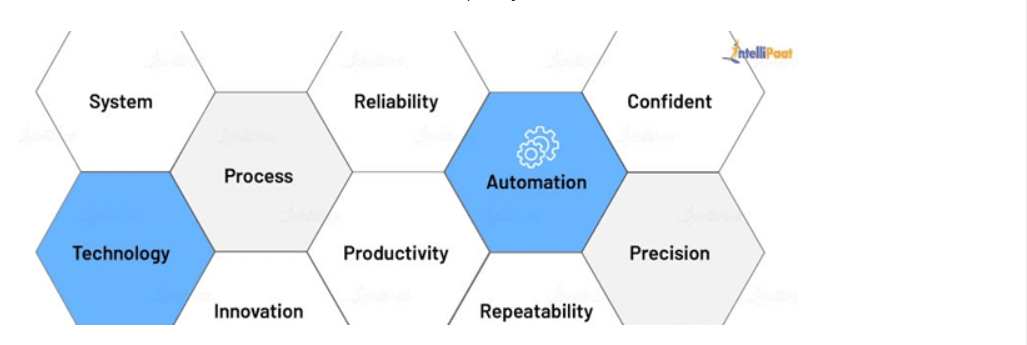
**What is Automation Testing?**

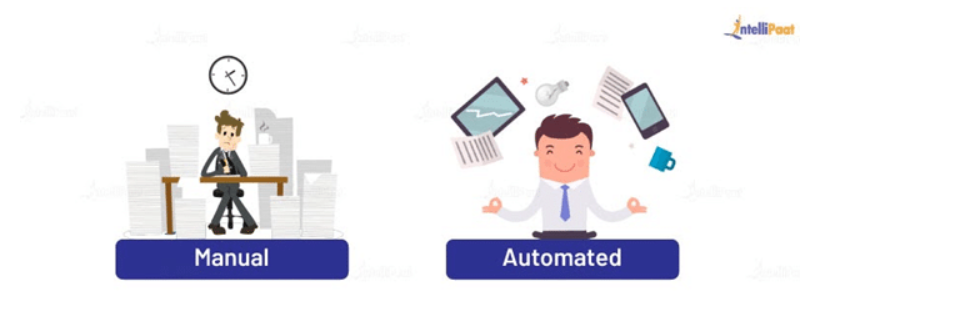
Automation Testing is the method of testing software products with special testing tools and frameworks to minimize human intervention and maximize quality.



Automation Testing is done with the help of automation software, and it controls the flow of the execution of tests as per the written test scripts. They are then compared with predicted outcomes to ensure the quality and reliability of the application. With Automation Testing, one can perform necessary repetitive tasks and those tasks that are hard to achieve with manual testing. Therefore, this type of testing is critical for CI/CD pipelines.

## ****Why Automation Testing?****

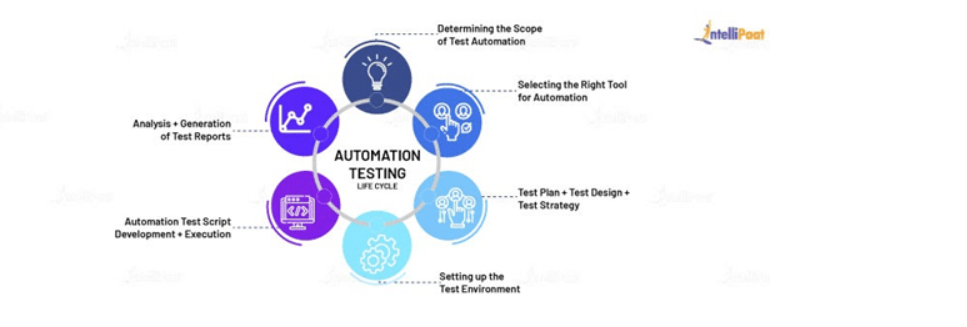
When a company develops a product, it is bound to have defects. So, before the release of the product, the company needs to capture the flaws in it to provide a seamless user experience. It is the responsibility of the testing team to perform various kinds of testing, from functional to non-functional, to ensure the effectiveness, efficiency, and better user experience of the overall product.



Though testers do [Manual and Automation testing](https://intellipaat.com/blog/automation-vs-manual-testing/) both, doing Automation Testing eases a lot of manual work, provides accurate results, and saves up a lot of time, which results in the quicker delivery of the product.

## ****Automation Testing Life Cycle****

To offer the best quality product, an organization should follow up with the step-by-step Automation Testing life cycle to pull off success in the market. Take a look at the below image that shows the six stages of Automation Testing:



### **The Scope of Automation Testing**

Before carrying out the testing process, one must check for the feasibility of automation. Here are the things to consider while identifying the scope of Testing Automation:

1. What are the modules that can be automated?
2. What are the total effective costs and the team size?
3. What are the tests to be automated, and which is the approach to be taken?

### **The Right Automation Tool**

Automation Testing is not possible without the right testing tool. Selecting the right tool is a critical phase in the testing life cycle where one has to consider the following points:

* Familiarity with the tool among the resources on-board
* Total budget and flexibility
* Technologies and programming languages used to build the project
* Choosing a tool that has a support team to take care of any queries and issues

### **Automation Test Plan, Design, and Strategy**

As the name suggests, in this phase, you make a plan, design the architecture, and create a strategy to achieve the goal of test automation.

* **Test plan**: Creation of test standards and procedures, hardware, software, and test data requirements
* **Test design**: Design the test architecture to determine the flow of the test procedures that follow
* **Test strategy**: Select a suitable test automation framework

### **Setting up the Test Environment**

You need to set up a remote machine or a machine where the test cases can be run. It should cover an extensive test coverage range for different test scenarios, different browsers, support desktop, and mobile applications.

### **Automation Test Script Execution**

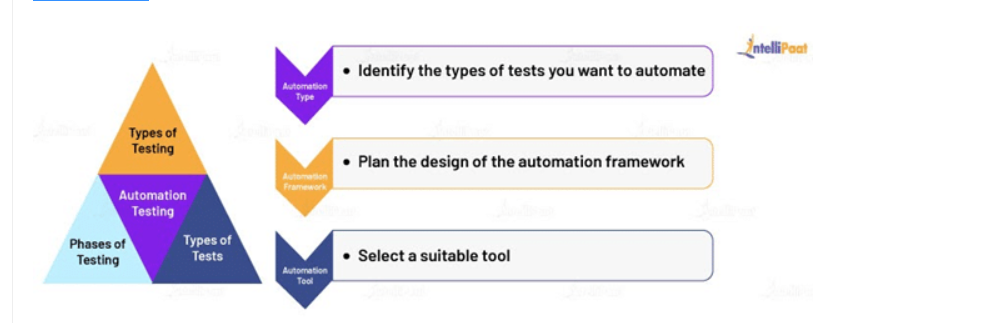
In this step, you will execute the test scripts and check whether the scripts run correctly or not. This step involves all the functional aspects and compatibility across multiple platforms. You also need to generate a bug report if the test case fails.

### **Analysis and the Generation of Test Results and Test Reports**

This is the last phase of the testing life cycle. Here, you will analyze the test reports to determine whether they need additional testing or not. Then, the generation of test results is done to confirm if the test scripts can identify errors in the application. Finally, the test reports are shared with the members/clients who are involved in the project.

## ****Types of Automation Testing****

When you are presented with a product, the knowledge of automation types will aid you to decide which kind of test suites you can use for automating. Automation Testing can be divided into three categories as given below:



### **Functional Testing**

Functional testing is about what the product does and verifies each function/feature of the application. It is based on customers’ requirements. With functional testing, you validate the actions that you perform in the software. It can be done manually as well as automated. An example of functional testing is testing the login functionality of a website.

### **Non-functional Testing**

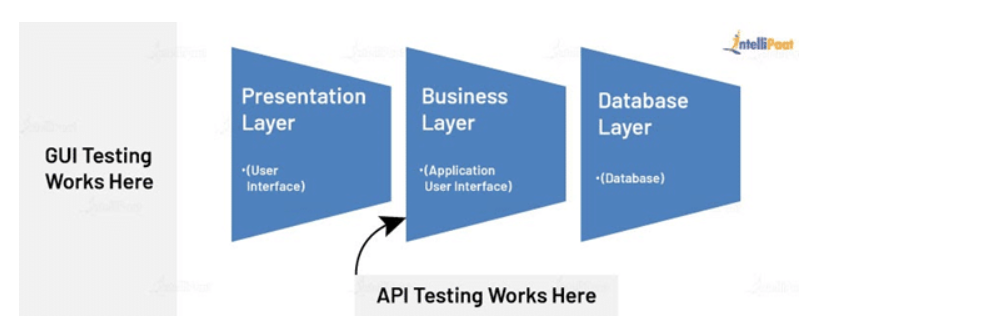
Non-functional testing is about how the product behaves and verifies the non-functional aspects, such as performance, reliability, usability, etc., of the product. It is based on customers’ expectations. With non-functional testing, you validate the performance of the software. It is hard to do manually. An example of non-functional testing is testing how long it takes for the dashboard in a website to load.

## ****Automation Testing Based on the Phases of Testing****

### **Unit Testing**

### A unit refers to the smallest component of the software. For the entire product/software to work well, it is necessary that all the individual parts of the code work as required. Unit testing gives a granular view of how the code is performing. It has a faster execution time since you are testing only chunks of code at a time. Usually, developers prefer to perform unit testing. **API Testing**

API stands for Application Programming Interface. It acts as a middle interface between the UI and the database.



API testing checks the end-to-end functionality of the application. Here, testers will not have access to the source code, and the process does not involve inputs and outputs from the keyboard. Instead, the software is made to send API calls to get the output, and the testers note the system/application response to check the functionality.

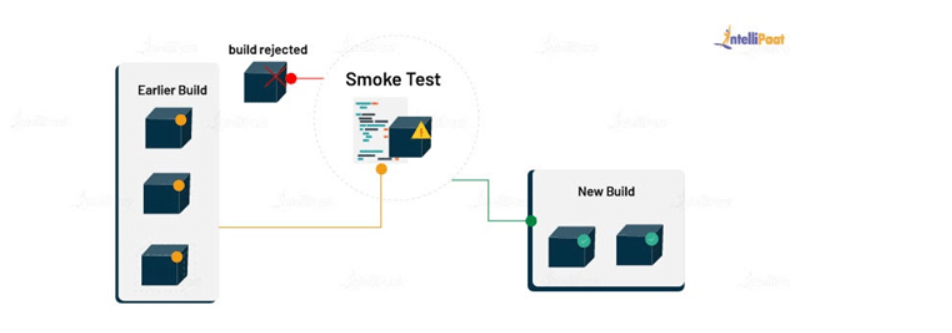
### **UI Testing**

In UI testing, the testers look for the correctness of the visual elements such as fields, buttons, labels, links, text fields, and images on the system’s screen. These elements need to be displayed correctly and work as intended for a better user experience. UI testing also checks for the functionality of the application in handling user actions, which are done via their keyboard, mouse, and other input devices. Its main intent is to provide a friendly user interface (UI) and experience.

## ****Automation Testing Based on the Types of Tests****

### **Smoke Testing**

Smoke testing is also known as build verification testing (BVT). It is done in the initial stages of application testing. Whenever you add a new feature or functionality to the existing build, smoke testing is done that acts as a checkpoint to move on to the next level of testing.



### **Integration Testing**

Integration testing is also known as I&T testing or string testing, or rarely as thread testing. It is to validate the proper communication between all the modules of the application. In other words, since the software is made up of a lot of smaller modules, in integration testing, testers logically group them and test them as groups to expose the flaws while maintaining the interaction between these software modules.

### **Regression Testing**

If you want to be certain whether the developed and tested software works the same way after changes have been made, then you perform regression testing. The changes can be bug fixes, configuration changes, or the enhancement of the software. You achieve this by re-running functional and non-functional tests on the application.

### **Security Testing**

As a tester, you do not want to incur data breaches, loss of revenue, and a blow on reputation due to unauthorized access. To avoid such incidents and prevent malicious intrusion, security testing is done, which unravels the underlying risky threads, vulnerabilities, threats, malware, and other risks in the software application. Security testing helps you spot all the shortcomings and fragilities in the system’s security, which you can prevent timely by coming up with effective solutions.

### **Performance Testing**

Even if your product gets a heavy workload, it needs to work without lag. So, you need to test your product before launching it for the customers to check its responsiveness, speed, stability, reliability, and robustness under a certain workload. An example of [**Performance Testing**](https://intellipaat.com/blog/what-is-performance-testing/) is checking the browser response time and the server request processing time.

### **Acceptance Testing**

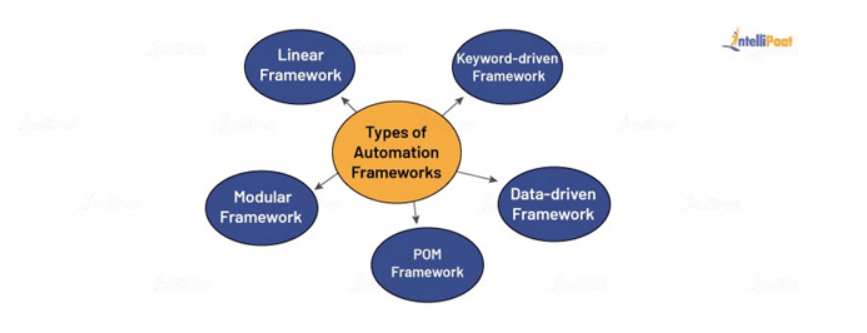
Acceptance testing is the last phase of testing before launching the product into the market. It is done to make sure all the user needs, business requirements, and the client’s demands are met and to determine if the product is fit for delivery or not.

## ****Testing Frameworks****

Let’s look at an example to learn about testing frameworks and their types.

In a classroom, you will have a few instructions to follow such as keeping the decorum of the class and avoiding unnecessary interactions with peers while the class is going on. Though it is possible to continue the class without following the instructions, it will not provide the desired results.

A testing framework does just this during the testing process. It has a set of guidelines for the professionals that include coding standards, repository management, and handling of test data to get beneficiary outcomes such as easy code reuse, reduced time to manage scripts, and high portability.



### **Linear Framework**

This is the simplest framework of all. Under this framework, you need to write a simple code to run the test cases without any modularity or sequential steps. It works as a record-and-playback model.

### **Keyword-driven Framework**

It is a scripting technique where you associate keywords with certain actions, such as opening or closing of a browser, mouse-click events, and others. Later on, in your test scripts, you can call these keywords to perform a specific step. Also, you will have a file where you will maintain all the keywords, along with the actions they perform.

### **Data-driven Framework**

In a data-driven framework, all the test case data inputs are stored in the table or in extension files, including .xls, .xml, .csv, etc. While executing the test scripts, it will read the values from the table. With the help of this framework, you can perform both positive and negative test cases.

### **Page Object Model Framework**

In the POM framework, you will create an object repository for the web UI elements. It allows you to call these methods later on without having to write the code again. Thus, it results in less verbosity, code reusability, and reduced time-consumption to write test scripts.

### **Modular Framework**

The modular framework allows you to divide the test scripts into small, independent modules. These modules will interact with each other in a hierarchical manner to run large test scripts. This will also help you create the required test scenarios and test the scripts individually.

## ****Automation Testing Tools****

An automation tool should be chosen based on the type of testing and the type of framework you are going to implement. There are a lot of tools available in the market for you to choose from as per your requirements. Some of the most widely used automation tools are listed below.

### **Selenium**

It is a tool to test web applications and web browsers. It has multiple powerful tools for testing web applications. Also, it supports multiple platforms and browsers with the help of [Selenium IDE](https://intellipaat.com/blog/tutorial/selenium-tutorial/selenium-ide/).

### **Sikuli**

Sikuli is an open-source automation testing tool used to perform GUI testing.

### **TestNG**

[**TestNG** in Selenium](https://intellipaat.com/blog/tutorial/selenium-tutorial/testng-in-selenium/)is more like a framework than a tool, which supports [Selenium](https://intellipaat.com/blog/tutorial/selenium-tutorial/introduction/), REST Assured, Appium, etc. Testers can generate HTML reports for the tests with their status—passed, failed, or skilled. Later on, they can re-execute the failed test cases.

### **SoapUI**

It is a testing tool used for API testing.

### **Appium**

It is a tool to perform mobile application testing and native app testing.

## ****Benefits of Automation Testing****

If you are learning Automation Testing, it is vital for you to know the benefits of learning it. Like every other type of testing, Automation Testing has its pros and cons. Here, you will learn about the 10 advantages of Automation Testing.



**Benefits of Automation Testing:**

1. It proves to be reliable since it is carried out by efficient testing tools.
2. 70% faster than manual testing, which saves a lot of time for testers and the organization as a whole.
3. Automation Testing avoids human intervention while executing test scripts.
4. It allows re-usability and re-running of test cases.
5. It increases the speed and efficiency of the software testing process.
6. Automation Testing covers all the application features.
7. It also allows for the faster delivery of the product into the market.
8. Faster feedback systems, which come with Automation Testing, results in an easy communication flow between developers and testers and helps detect and fix bugs earlier.
9. It offers improved accuracy as it holds no possibility of human errors.
10. It has the capability to run massive amounts of test cases at a time.