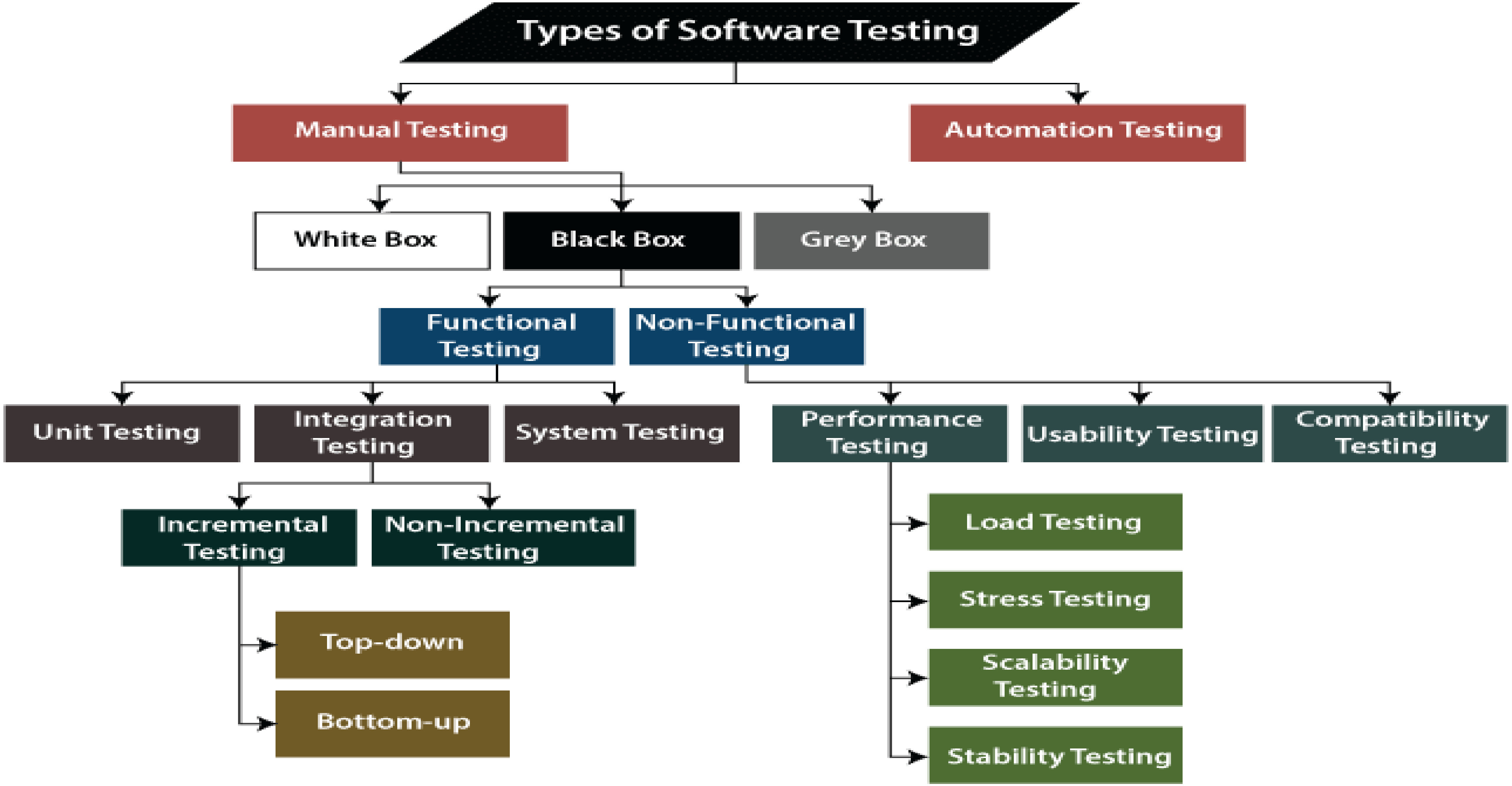
# *Various types of testing*

- Testing is the process of executing a program with the aim of finding errors.
- **Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.
- To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.
- It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.
- . In simple terms, Software Testing means the Verification of Application Under Test (AUT). Some prefer saying Software testing definition as a White Box and Black Box Testing.

## ❖**Why Software Testing is Important?**

- **Software Testing is Important** because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

# Types of Software Testing



# ❖What is Manual Testing?

- Testing any software or an application according to the client's needs without using any automation tool is known as **manual testing**. Manual testing includes testing a software manually, i.e., without using any automated tool or any script.
- This usually includes verifying all the features specified in requirements documents, but often also includes the testers trying the software with the perspective of their end user’s in mind.
- In other words, we can say that it is a procedure of **verification and validation**. Manual testing is used to verify the behaviour of an application or software in contradiction of requirements specification.
- We do not require any precise knowledge of any testing tool to execute the manual test cases. We can easily prepare the test document while performing manual testing on any application.
- At a high level, we need to make the distinction between manual and automated tests. Manual testing is done in person, requires someone to set up an environment and execute the tests themselves, and it can be prone to human error as the tester might make typos or omit steps in the test script.
- In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.
- Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

# ❑What is Automation Testing?

- Automation testing is the process of testing the software using an automation tool to find the defects. . automation testing takes software testing activities and executes them via an automation toolset or framework
- In this process, testers execute the test scripts and generate the test results automatically by using automation tools.so,we can say that whenever we are testing an application by using some tools is known as **automation testing**.
- As the name suggests. In simple words, it is a type of testing in which a tool executes a set of tasks in a defined pattern automatically. This process involves automation of a manual process . It uses specific tools to automate manual design test cases without any human interference
- It is used to re-run the test scenarios, which were executed manually, quickly, and repeatedly
- **Automated testing** is a method in software testing that leverages automation tools to control the execution of tests instead of a human tester.It then compares actual test results with predicted or expected results. Automated testing offers greater efficiency and faster time-to-market for your projects.
- We cannot write the test script or perform the automation testing without understanding the programming language.
- Automation testing is a Software testing technique to test and compare the actual outcome with the expected outcome. This can be achieved by writing test scripts or using any automation testing tool. Test automation is used to automate repetitive tasks and other testing tasks which are difficult to perform manually.
- It takes the pressure off manual testers, and allows them to focus on higher-value tasks – exploratory tests, reviewing test results, etc. Essentially, a machine takes over and implements, mundane, repetitive, time-consuminging tasks such as regression tests. Automation testing is essential to achieving greater test coverage within shorter timelines, as well as greater accuracy of results.
- Automation testing is the best way to enhance the efficiency, productivity, and coverage of Software testing.

# ❖**Functional Testing types include:**

- Unit Testing
- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance Testing

# ❖FUNCTIONAL TESTING

- Functional testing is a type of software testing that validates the software system against the functional requirements/specifications. the purpose of functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the functional requirements.

- functional testing mainly involves black box testing and it is not concerned about the source code of the application. this testing checks user interface, apis, database, security, client/server communication and other functionality of the application under test. the testing can be done either manually or using automation.

## ❑what do you test in functional testing?

- The prime objective of functional testing is checking the functionalities of the software system. it mainly concentrates on –
- **mainline functions**: testing the main functions of an application
- **basic usability**: it involves basic usability testing of the system. it checks whether a user can freely navigate through the screens without any difficulties.
- **accessibility**: checks the accessibility of the system for the user
- **error conditions**: usage of testing techniques to check for error conditions. it checks whether suitable error messages are displayed.

## 1.Unit Testing:

- Unit testing is the first level of functional testing in order to test any software.
- In this, the test engineer will test the modules or units of an application independently or test all the module functionality is called **unit testing .**It focuses on the smallest unit of software design.
- a unit is defined as a single testable function of a software or an application. And it is verified throughout the specified application development phase.
- As the entire protect is divided into various modules called units.each unit should be separately tested to check its functionality and quality .so, unit testing focus on testing these independent units of code to ensure that each unit is correctly working.
- In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.
- It helps developers to know whether the individual unit of the code is working properly or not.
- The primary objective of executing the unit testing is to confirm the unit components with their performance. Here Unit tests are very low level, close to the source of your application.. Unit tests are in general quite cheap to automate and can be run very quickly by a continuous integration server.

## 2. Integration Testing

- Integration testing is testing in which a group of units or components are integrated an d tested to produce output.
- Testing of all integrated modules to verify the combined functionality after integration is termed as Integration Testing

- The objective is to take unit tested components and build a program structure that has been dictated by design
- You need to see that the integrated units are working without errors or not.
- It focuses on the construction and design of the software.
- Integration testing ensures that an integrated cod eunits system meets a set of requirements and works properly or not
- in this testing , where we test the data flow between dependent modules or interface between two features is called **integration testing**.
- The purpose of executing the integration testing is to test the statement's accuracy between each module.
- Integration testing can also be called an **Incremental** Integration **Testing.**

## ❖Incremental Integration Testing

- Whenever there is a clear relationship between modules, we go for incremental integration testing. Suppose, we take two modules and analysis the data flow between them if they are working fine or not.
- If these modules are working fine, then we can add one more module and test again. And we can continue with the same process to get better results.
- In other words, we can say that incrementally adding up the modules and test the data flow between the modules is known as **Incremental integration testing**.

# 3.System Testing

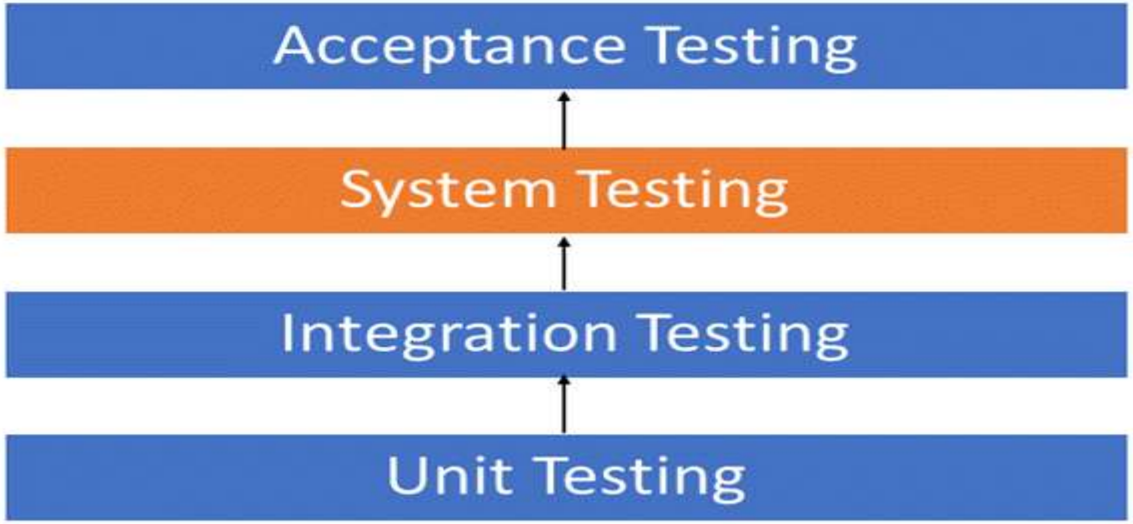
- **System Testing** is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

- System Testing is performed after the integration testing and before the acceptance testing.

- In system testing, is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system

- System test falls under the **black box testing** category of software testing.

- System testing tests the design and behaviour of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).
- **Two Category of Software Testing**
- Black Box Testing
- White Box Testing
- **White box testing** is the testing of the internal workings or code of a software application. In contrast, black box or System Testing is the opposite. System test involves the external workings of the software from the user’s perspective.
- Mostly System Testing is Blackbox



## **4.Sanity Testing** .

Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

The objective is “not” to verify thoroughly the new functionality but to determine that the developer has applied some rationality (sanity) while producing the software.

Example:- For instance, if your scientific calculator gives the result of 2 + 2 =5! Then, there is no point testing the advanced functionalities like sin 30 + cos 50.

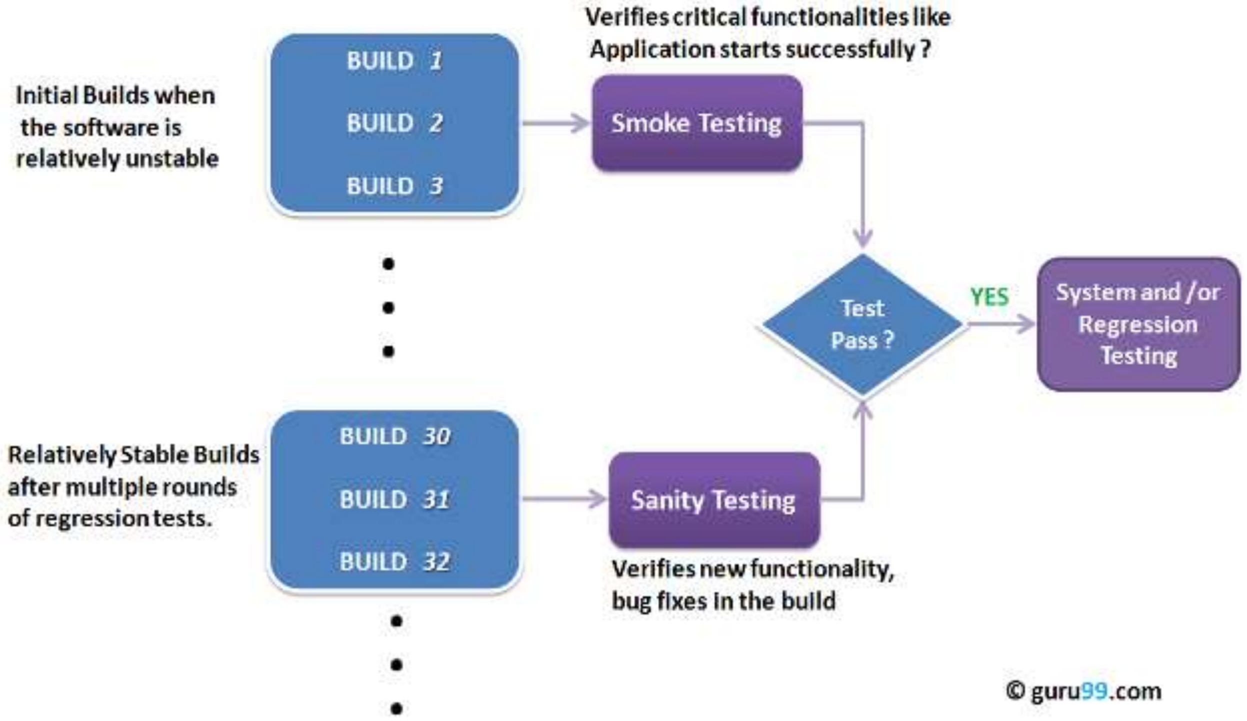
- Sanity testing is performed to ensure that the code changes that are made are working as properly. Sanity testing is a stoppage to check whether testing for the build can proceed or not. The focus of the team during sanity testing process is to validate the functionality of the application and not detailed testing.
- In other words, we can say that sanity testing is performed to make sure that all the defects have been solved and no added issues come into the presence because of these modifications.

- **Functionality of Sanity Testing:**
  The major functionality of sanity testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, software product is rejected by the testing team to save time and money. It is performed only after the software product has passed the smoke test and Quality Assurance team has accepted for further testing.
- Sanity testing is generally performed on build where the production deployment is required immediately like a critical bug fix.
- Sanity is a subset of regression testing

# 5.Smoke Testing

- **Smoke Testing** is a software testing technique performed post software build to verify that the critical functionalities of software are working fine.
- In simple terms, we are verifying whether the important features are working and there are no showstoppers in the build that is under testing.
- It is executed before any detailed functional or regression tests are executed. The main purpose of smoke testing is to reject a software application with defects so that QA team does not waste time testing broken software application.

- In Smoke Testing, the test cases chose to cover the most important functionality or component of the system. The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine.
  **For Example**, a typical smoke test would be – Verify that the application launches successfully, Check that the GUI is responsive … etc.
- Smoke testing is a confirmation for QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”


Verifies critical functionalities like Application starts successfully ?
Initial Builds when the software is relatively unstable
BUILD 1
BUILD 2
BUILD 3
Smoke Testing
Test Pass ?
YES
System and /or Regression Testing
Relatively Stable Builds after multiple rounds of regression tests.
BUILD 30
BUILD 31
BUILD 32
Sanity Testing
Verifies new functionality, bug fixes in the build
© guru99.com

# 6.What is Regression Testing?

- Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product.
- regression testing is defined as a type of software testing to to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done
- confirm that a recent program or code change has not adversely affected existing features.
- This is to ensure that the product works fine with new functionality, bug fixes or any changes to the existing feature. Previously executed test cases are re-executed in order to verify the impact of the change or to know whether new changes have not introduced any new bugs
- . Regression testing is a black box testing techniques
- Regression tests are also known as the Verification Method. Test cases are often automated. Test cases are required to execute many times and running the same test case again and again manually, is time-consuming and tedious too.
- **Need of Regression Testing**
- The **Need of Regression Testing** mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not. Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

❑ **When to apply regression testing**

- Typically, regression testing is applied under these circumstances:
- A new requirement is added to an existing feature
- A new feature or functionality is added
- The codebase is fixed to solve defects
- The source code is optimized to improve performance
- Patch fixes are added
- Changes in configuration

## 7. Interface Testing

- Interface Testing is defined as a software testing type which verifies whether the communication between two different software systems is done correctly.
- A connection that integrates two components is called interface.
- Interface Testing is a type of software testing type that checks the proper communication between two different software systems. Interface is the connection that integrates two components. The interface could be anything like APIs, web services etc. Testing of these connecting interfaces is defined as Interface Testing.
- Interface Testing is performed to evaluate whether systems or components pass data and control correctly to one another. It is to verify if all the interactions between these modules are working properly and errors are handled properly. An interface is actually a software that consists of sets of commands, messages and other attributes that enable communication between a device and a user.

- **Interface Testing includes testing of two main segments**
- Web server and application server interface
- Application server and Database server interface

- **Interface Testing - Checklist**
- Verify that communication between the systems are done correctly
- Verify if all supported hardware/software has been tested
- Verify if all linked documents be supported/opened on all platforms
- Verify the security requirements or encryption while communication happens between systems
- Check if a Solution can handle network failures between Web site and application server

- **Example of Interface Testing**
- Suppose for any xyz application, the interface takes XML file as an input and delivers JSON file as an output. To test the interface of this application, all it requires is the specifications of XML file format and JSON file format.
- With the help of these specifications, we can create a sample input XML files and feed into the interface. And then validating the input (XML) and output (JSON) file with the requirement is Interface testing

# 8.Acceptance Testing

- Acceptance tests are formal tests executed to verify if a system satisfies its business requirements
- An Acceptance Test is performed by the client and it verifies whether the end to end flow of the system is as per the business requirements or not and if it is as per the needs of the end-user.
- Client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).
- . 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.
- The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

## ❖**Types of acceptance test:**

- ➢ Alpha Testing
- ➢ **Beta Testing**

## ❑**Alpha Testing:-**

- Alpha Testing will be conducted at the developer's site. An in-house virtual user environment can be created for this type of testing.
- Alpha Testing will be carried out at the end of the software development phase
- The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user.
- Specialized Testers will perform this testing to check any bugs have occurred. Also, they help to give suggestions to improve product usability in a controlled manner

## ❑Beta Testing

- Beta Testing is a formal type of Software Testing which is carried out by the customer. It is performed in **the Real Environment** before releasing the product to the market for the actual end-users.
- Beta Testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.
- **Beta Testing** is performed by “real users” of the software application in “real environment” and it can be considered as a form of external User Acceptance Testing. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in customer’s environment.
- Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

## ❖Non-functional Testing types include:

- Performance Testing
- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing
- Reliability Testing
- Recovery Testing
- Usability Testing
- Localization Testing
- Compliance testing
- Maintainability testing.
- Portability testing.

## ❖Non-functional Testing types include:

- **NON-FUNCTIONAL TESTING** is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, reliability, load test, performance and accountability etc) of a software application. It is designed to test the readiness of a system as per non-functional parameters which are never addressed by functional testing.
- The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters. The parameters of non-functional testing are never tested before the functional testing
- Non-functional testing is also very important as functional testing because it plays a crucial role in customer satisfaction.
- For example, non-functional testing would be to test An excellent example of non-functional test would be to check how many people can simultaneously login into a software, how many people can work simultaneously on any software.

## 1.performance Testing

- Performance testing examines the speed, stability, reliability, scalability, and resource usage of a software application under a specified workload.
- This term is often used interchangeably with ‘stress’ and ‘load’ testing.
- Performance Testing is done to check whether the system meets the performance requirements. Different performance and load tools are used to do this testing.

## 2.Load Testing

- It is a type of Non-Functional Testing and the objective of Load Testing is to check how much load or maximum workload a system can handle without any performance degradation. This type of non-functional software testing process determines how the software application behaves while being accessed by multiple users simultaneously.

- Load Testing helps to find the maximum capacity of the system under specific load and any issues that cause software performance degradation. Load testing is performed using tools like JMeter, LoadRunner, WebLoad, Silk performer, etc.

## 3.Stress Testing

- This testing is done when a system is stressed beyond its specifications in order to check how and when it fails.
- This is performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to the system or database load.

## 4.Volume Testing

- Volume Testing is a type of Non-Functional Testing performed by the Performance Testing team.
- The software or application undergoes a huge amount of data and Volume Testing checks the system behavior and response time of the application when the system came across such a high volume of data.
- This high volume of data may impact the system’s performance and speed of processing time.

## 5.Security Testing

- Security Testing is done to check how the software, application or website is secure from internal and external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.
- It is a type of testing performed by a special team of testers. A system can be penetrated by any hacking method.
- It also checks how software behaves for any hackers attack & malicious programs and how software is maintained for data security after such a hacker attack. These tests aim to find any potential flaws and weaknesses in the software system that could lead to a loss of data, revenue, or reputation per employees or outsides of a company.

## 6.Compatibility Testing

- This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment.
- Compatibility testing ensures that software can run on a different configuration, different databases, different browsers, and their versions. Compatibility testing is performed by the testing team.

## 7.Recovery Testing

- It is a type of testing which validates how well the application or system recovers from crashes or disasters.
- Recovery Testing determines if the system is able to continue its operation after a disaster. Assume that the application is receiving data through a network cable and suddenly that network cable has been unplugged. Sometime later, plug in the network cable; then the system should start receiving data from where it lost the connection due to network cable being unplugged.

## 8.Usability Testing

- Under Usability Testing, the User-Friendliness Check is done.
- The application flow is tested to see if a new user can understand the application easily or not. Proper help is documented if a user gets stuck at any point. Basically, system navigation is checked in this testing.

## 9.Reliability test

- Reliability test assumes that whether the software system is running without fail under specified conditions or not. The system must be run for a specific time and number of processes. If the system is failed under these specified conditions, reliability test will be failed.

- **Localization testing.** Localization testing checks that the product meets the expectations of a local audience. For example, if you localize a mobile app designed for the USA to the Chinese market and translate it in Chinese, you will have to use this type of testing.
- **Compliance testing.** This type of software testing verifies that the product meets international standards of software development, usually developed by global companies. For instance, it is important for a mobile application to be compliant to the App Store or Google Play regulations.
- **Maintainability testing.** The ability of a program to safely go through changes and updates is crucial. Maintainability testing measures how well the system copes with changes.
- **Portability testing.** It determines how easy it is to transport a software component or application from one hardware or OS to another.

# What are the benefits of Software Testing?

- Here are the benefits of using software testing:
- **Cost-Effective:** It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
- **Security:** It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
- **Product quality:** It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
- **Customer Satisfaction:** The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

# *Automation testing , its pros and cons*

- Automation testing is the process of testing the software using an automation tool to find the defects. . automation testing takes software testing activities and executes them via an automation toolset or framework

- In this process, testers execute the test scripts and generate the test results automatically by using automation tools.so,we can say that whenever we are testing an application by using some tools is known as **automation testing**.

- As the name suggests. In simple words, it is a type of testing in which a tool executes a set of tasks in a defined pattern automatically. This process involves automation of a manual process . It uses specific tools to automate manual design test cases without any human interference

- It is used to re-run the test scenarios, which were executed manually, quickly, and repeatedly

- **Automated testing** is a method in software testing that leverages automation tools to control the execution of tests instead of a human tester. **?**. It then compares actual test results with predicted or expected results. Automated testing offers greater efficiency and faster time-to-market for your projects.

- We cannot write the test script or perform the automation testing without understanding the programming language.

- Automation testing is a Software testing technique to test and compare the actual outcome with the expected outcome. This can be achieved by writing test scripts or using any automation testing tool. Test automation is used to automate repetitive tasks and other testing tasks which are difficult to perform manually.

- It takes the pressure off manual testers, and allows them to focus on higher-value tasks – exploratory tests, reviewing test results, etc. Essentially, a machine takes over and implements, mundane, repetitive, time-confusing tasks such as regression tests. Automation testing is essential to achieving greater test coverage within shorter timelines, as well as greater accuracy of results.

- Automation testing is the best way to enhance the efficiency, productivity, and coverage of Software testing.

- Although some types of testing, such as regression or functional testing can be done manually, there are greater benefits of doing it automatically. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what's been found, and this information can be compared with earlier test runs. Automation developers generally write in the following programming languages: C#, JavaScript, and Ruby.

- Many software businesses will have an appointed QA (quality assurance) automation tester. They design and write the test scripts in the beginning stages. The QA automation tester will work with automation test engineers and product developers to actually test the software and products. They will form a team and control the test automation initiatives, and use different types of test automation frameworks to establish the best one for successful test automation.

- **Test Automation** is the best way to reasons:
- Test Automation in software testing does not require Human intervention. You can run automated test and unattended (overnight)
- Test Automation increases the speed of test execution
- Automation helps increase Test Coverage
- Manual Testing can become boring and hence error-prone
- It is difficult to test for multilingual sites manually

## ❖**Which Test Cases to Automate?**

- Test cases to be automated can be selected using the following criterion to increase the automation ROI
- High Risk – Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming
- **The following category of test cases are not suitable for automation:**
- Test Cases that are newly designed and not executed manually at least once
- Test Cases for which the requirements are frequently changing
- Test cases which are executed on an ad-hoc basis.

# ❖ Automated Testing Process:

- Following steps are followed in an Automation Process
- **Step 1)** Test Tool Selection
- **Step 2)** Define scope of Automation
- **Step 3)** Planning, Design and Development
- **Step 4)** Test Execution
- **Step 5)** Maintenance

## Step 1) Test Tool Selection

Test Tool selection largely depends on the technology the Application Under Test is built on. For instance QTP does not support Informatica. So QTP cannot be used for testing INFORMATICA applications.

## Step 2): Define scope of Automation

- The scope of automation is the area of your Application Under Test which will be automated. Following points help determine scope:
- The features that are important for the business.

- Scenarios which have **a large amount of data** ,**Common functionalities** across applications.
- Technical feasibility and The extent to which business components are reused.
- **The complexity** of test cases and Ability to use the same test cases for cross-browser testing.

## Step 3:-Planning, Design, and Development

- During this phase, you create an Automation strategy & plan, which contains the following details-
- Automation tools selected
- Framework design and its features
- In-Scope and Out-of-scope items of automation
- Automation testbed preparation
- Schedule and Timeline of scripting and execution
- Deliverables of Automation Testing

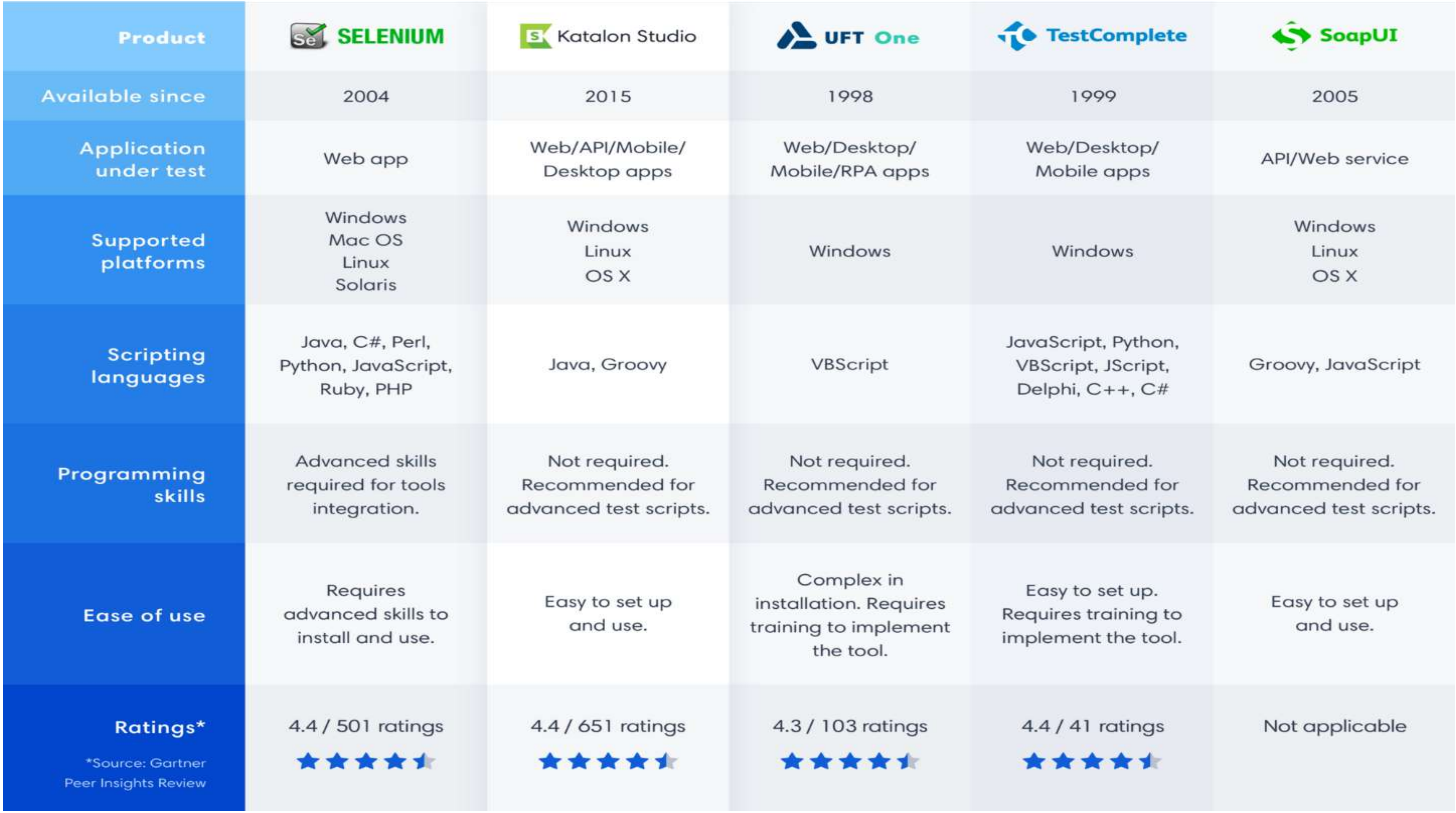
## Step 4:- Test Execution

- Automation Scripts are executed during this phase. The scripts need input test data before there are set to run. Once executed they provide detailed test reports.
- Execution can be performed using the automation tool directly or through the Test Management tool which will invoke the automation tool.

- **Example**: Quality center is the Test Management tool which in turn it will invoke QTP for execution of automation scripts. Scripts can be executed in a single machine or a group of machines. The execution can be done during the night, to save time.

## Step 5:- Test Automation Maintenance Approach

- **Test Automation Maintenance Approach** is an automation testing phase carried out to test whether the new functionalities added to the software are working fine or not. Maintenance in automation testing is executed when new automation scripts are added and need to be reviewed and maintained in order to improve the effectiveness of automation scripts with each successive release cycle.



| Product | SELENIUM | Katalon Studio | UFT One | TestComplete | SoapUI |
|---|---|---|---|---|---|
| Available since | 2004 | 2015 | 1998 | 1999 | 2005 |
| Application under test | Web app | Web/API/Mobile/ Desktop apps | Web/Desktop/ Mobile/RPA apps | Web/Desktop/ Mobile apps | API/Web service |
| Supported platforms | Windows<br>Mac OS<br>Linux<br>Solaris | Windows<br>Linux<br>OS X | Windows | Windows | Windows<br>Linux<br>OS X |
| Scripting languages | Java, C#, Perl, Python, JavaScript, Ruby, PHP | Java, Groovy | VBScript | JavaScript, Python, VBScript, JScript, Delphi, C++, C# | Groovy, JavaScript |
| Programming skills | Advanced skills required for tools integration. | Not required. Recommended for advanced test scripts. | Not required. Recommended for advanced test scripts. | Not required. Recommended for advanced test scripts. | Not required. Recommended for advanced test scripts. |
| Ease of use | Requires advanced skills to install and use. | Easy to set up and use. | Complex in installation. Requires training to implement the tool. | Easy to set up. Requires training to implement the tool. | Easy to set up and use. |
| Ratings*<br>*Source: Gartner Peer Insights Review | 4.4 / 501 ratings<br>★★★★⯪ | 4.4 / 651 ratings<br>★★★★⯪ | 4.3 / 103 ratings<br>★★★★⯪ | 4.4 / 41 ratings<br>★★★★⯪ | Not applicable |

# ❖**Automated Testing Pros**

- 70% faster than the manual testing
- Wider test coverage of application features
- Reliable in results and Ensure Consistency
- Saves Time and Cost ,Improves accuracy
- Human Intervention is not required while execution
- Increases Efficiency and Better speed in executing tests
- Re-usable test scripts and Test Frequently and thoroughly
- More cycle of execution can be achieved through automation
- Early time to market
- Automated testing helps you to find more bugs compare to a human tester
- As most of the part of the testing process is automated, you can have a speedy and efficient process
- Automation process can be recorded. This allows you to reuse and execute the same kind of testing operations
- Automated testing is conducted using software tools, so it works without tiring and fatigue unlike humans in manual testing
- It can easily increase productivity because it provides fast & accurate testing result
- Automated testing support various applications
- Testing coverage can be increased because of automation testing tool never forget to check even the smallest unit

- reduced costs: One aim of automating the testing process is to reduce the overall resources spent on this process, thereby cutting testing costs as a whole. Additionally, test automation facilitates quick and early detection of application bugs and faults, allowing for correction early on, which also serves to cut down the total cost of developing the final product;
- **cuts down on QA process times by up to 80%, which reduces lead time of launching the product;**
- generates quicker Return On Investment (ROI);
- improves overall organizational efficiency;
- faster Feedback: Testing for new features can take a long time with manual testing. Test automation helps reduce the feedback cycle and helps provide relevant information faster, allowing for the product to get to the market in a shorter time period.
- **Detailed reporting capabilities** - Automation testing uses well-crafted test cases for various scenarios. These scripted sequences can be incredibly in-depth, and provide detailed reports that simply wouldn’t be possible when done by a human. Not to mention providing them in a shorter amount of time.
- **Improved bug detection** - One of the main reasons to test a product is to detect bugs and other defects. Automation testing makes this process an easier one. It’s also able to analyze a wider test coverage than humans may be able to.
- **Simplifies testing** - Testing is a routine part of the operations of most SaaS and tech companies. Making it as simple as possible is key. Using automation is extremely beneficial. When automating test tools, the test scripts can be reused. Manual testing, meanwhile, calls for a single code line to be written for the same test case, each time it needs to be run
- **Speeds up the testing process** - Machines and automated technology work faster than humans. Along with improved accuracy, this is why we use them. In turn, this shortens your software development cycles.
- **Reduces human intervention** - Tests can be run at any time of day, even overnight, without needing humans to oversee it. Plus, when it’s conducted automatically, this can also reduce the risk of human error.

- **Saves time and money** - Testing can be time-consuming. Though automation may require an initial investment, it can save money in the long run to become more cost-effective. Team members use their time in other areas and are no longer required to carry out manual testing in many situations. This improves their workflow.
- **Improves Accuracy**
- Mistakes during manual testing are inescapable. Human testers inevitably lead to human errors when executing test cases manually. One of the advantages of automation testing is that test automation follows strict code when performing tests, meaning the same steps are performed when executing thousands of test cases. Because detailed record-keeping can be kept on all test results, test automation can drastically increase accuracy within repetitive test cases as well as for stress testing.
- **Enhances Test Coverage**
- Imagine executing thousands of complex test cases across multiple systems simultaneously—that's just one of the benefits of automation testing. With test automation, your team can examine all the intricacies of your software application in order to accurately analyze its file contents, memory contents, internal program states and data tables. Manual testing simply doesn't have the time nor the means to promise this level of test coverage.
- **Insightful Reporting**
- You want results for every test case executed—and test automation can deliver on that promise. One of the advantages of automation testing is having the ability to track every test script through visual logs that provide details on the number of test cases scheduled or executed as well as the number of reported defects and how these bugs have been resolved. By setting up test automation efficiently, your team can gather additional testing data, from productivity stats to performance metrics.
- **Increase Productivity**
- Humans can only work so many hours of the day, requiring breaks and a proper sleeping schedule to be their best at work. One of the benefits of automation testing is that testing can happen around the clock without reducing the quality or quantity of test performance. Your test engineers can schedule tests at the end of their 9-to-5 shift and arrive the next day with test automation results ready to review.

# ❖Cons or Disadvantages of Automated Testing :-

- Automated testing is very much expensive than the manual testing.
- It also becomes inconvenient and burdensome as to decide who would automate and who would train.
- It has limited to some organisations as many organisations not prefer test automation.
- Automated testing would also require additionally trained and skilled people.
- Automated testing only removes the mechanical execution of testing process, but creation of test cases still required testing professionals.
- Proficiency is required to write the automation test scripts.
- Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to deadly consequences.
- Test maintenance is costly in case of playback methods. Even though a minor change occurs in the GUI, the test script has to be re-recorded or replaced by a new test script.
- Maintenance of test data files is difficult, if the test script tests more screens.
- **developing time needed**: The first time developing the testing software is a time-taking process.
- **debugging script is challenging**: Not only is debugging tricky, but the bigger issue is that sometimes the errors missed in the script can lead to serious consequences.
- **expensive tools and labor**: Specialized automated software tools come at a premium. Additionally, automated QA engineers are more expensive to hire than their manual counterparts.
- **test program needs support and maintenance**: As and when conditions change, the program's coding needs to be modified and re-tested to verify it is working per requirements. This is a costly process.

# ❖Pitfalls of Automation Testing

- What you should pay attention to:
- Not all test cases need to be automated.
- Time needed for automation testing is comparable to the time spent on software development.
- Before writing automated tests you need to work on test system architecture. Otherwise apart from manual testing you will need to support automated tests.
- When implementing automation, it is impossible to foresee all the risks and challenges you might face in the future.

# Manual vs automation testing

| | | |
|---|---|---|
| Definition | Automation Testing uses automation tools to execute test cases. | In manual testing, test cases are executed by a human tester and software. |
| Processing time | Automated testing is significantly faster than a manual approach. | Manual testing is time-consuming and takes up human resources. |
| Exploratory Testing | Automation does not allow random testing | Exploratory testing is possible in Manual Testing |
| Initial investment | The initial investment in the automated testing is higher. Though the ROI is better in the long run. | The initial investment in the Manual testing is comparatively lower. ROI is lower compared to Automation testing in the long run. |
| Reliability | Automated testing is a reliable method, as it is performed by tools and scripts. There is no testing Fatigue. | Manual testing is not as accurate because of the possibility of the human errors. |
| UI Change | For even a trivial change in the UI of the AUT, Automated Test Scripts need to be modified to work as expected | Small changes like change in id, class, etc. of a button wouldn’t thwart execution of a manual tester. |

| | | |
|---|---|---|
| Cost-effective | Not cost effective for low volume regression | Not cost effective for high volume regression. |
| Test Report Visibility | With automation testing, all stakeholders can login into the automation system and check test execution results | Manual Tests are usually recorded in an Excel or Word, and test results are not readily/ readily available. |
| Human observation | Automated testing does not involve human consideration. So it can never give assurance of user-friendliness and positive customer experience. | The manual testing method allows human observation, which may be useful to offer user-friendly system. |
| Performance Testing | Performance Tests like Load Testing, Stress Testing, Spike Testing, etc. have to be tested by an automation tool compulsorily. | Performance Testing is not feasible manually |
| Parallel Execution | This testing can be executed on different operating platforms in parallel and reduce test execution time. | Manual tests can be executed in parallel but would need to increase your human resource which is expensive |
| Batch testing | You can Batch multiple Test Scripts for nightly execution. | Manual tests cannot be batched. |

| | | |
|---|---|---|
| Programming knowledge | Programming knowledge is a must in automation testing. | No need for programming in Manual Testing. |
| Set up | Automation test requires less complex test execution set up. | Manual testing needs have a more straightforward test execution setup |
| Engagement | Done by tools. Its accurate and never gets bored! | Repetitive Manual Test Execution can get boring and error-prone. |
| Ideal approach | Automation testing is useful when frequently executing the same set of test cases | Manual testing proves useful when the test case only needs to run once or twice. |
| Build Verification Testing | Automation testing is useful for Build Verification Testing (BVT). | Executing the Build Verification Testing (BVT) is very difficult and time-consuming in manual testing. |
| Deadlines | Automated Tests have zero risks of missing out a pre-decided test. | Manual Testing has a higher risk of missing out the pre-decided test deadline. |

|  |  |  |
|---|---|---|
| Framework | Automation testing uses frameworks like Data Drive, Keyword, Hybrid to accelerate the automation process. | Manual Testing does not use frameworks but may use guidelines, checklists, stringent processes to draft certain test cases. |
| Documentation | Automated Tests acts as a document provides training value especially for automated unit test cases. A new developer can look into a unit test cases and understand the code base quickly. | Manual Test cases provide no training value |
| Test Design | Automated Unit Tests enforce/drive Test Driven Development Design. | Manual Unit Tests do not drive design into the coding process |
| Devops | Automated Tests help in Build Verification Testing and are an integral part of DevOps Cycle | Manual Testing defeats the automated build principle of DevOps |
| When to Use? | Automated Testing is suited for Regression Testing, Performance Testing, Load Testing or highly repeatable functional test cases. | Manual Testing is suitable for Exploratory, Usability and Adhoc Testing. It should also be used where the AUT changes frequently. |