Unit 4

Continuous Testing

Computer Science and Engineering Department@SoEEC
SQAT(CSE5310)

Outline:

- Automated vs manual testing
- What is automation
- Quality Assurance vs Quality Control
- Test automation development process, and software quality achievement through automation.
- Different test automation frameworks and libraries developments.
- Quality certification process.
- Selenium framework architecture and its capabilities

Manual Testing

- Manual testing is testing, where the tester can test the application without any knowledge of any programming language.
- In manual testing, the test engineer tests the application like a user to make it bug-free or stable.
- Manual test engineers always search for the fault or bugs in the product before the product released in the market, yet delivered software still has defects.
- And there is a chance that the final software product still has a defect or does not meet the customer requirement, even the manual test engineer do their best.

Automation Testing

- Automation testing is a process of changing any manual test case into the test scripts by using automation testing tools, and scripting or programming language is called automation.
- Automation testing is used to increase the efficiency, effectiveness, and coverage of Software testing.
- Automation test engineer uses automation testing tools to automate the manual design test cases without any human interference.
- And these testing tools can control the execution of tests, access the test data, and compares the actual result against the expected result.

Difference between automation testing and manual testing

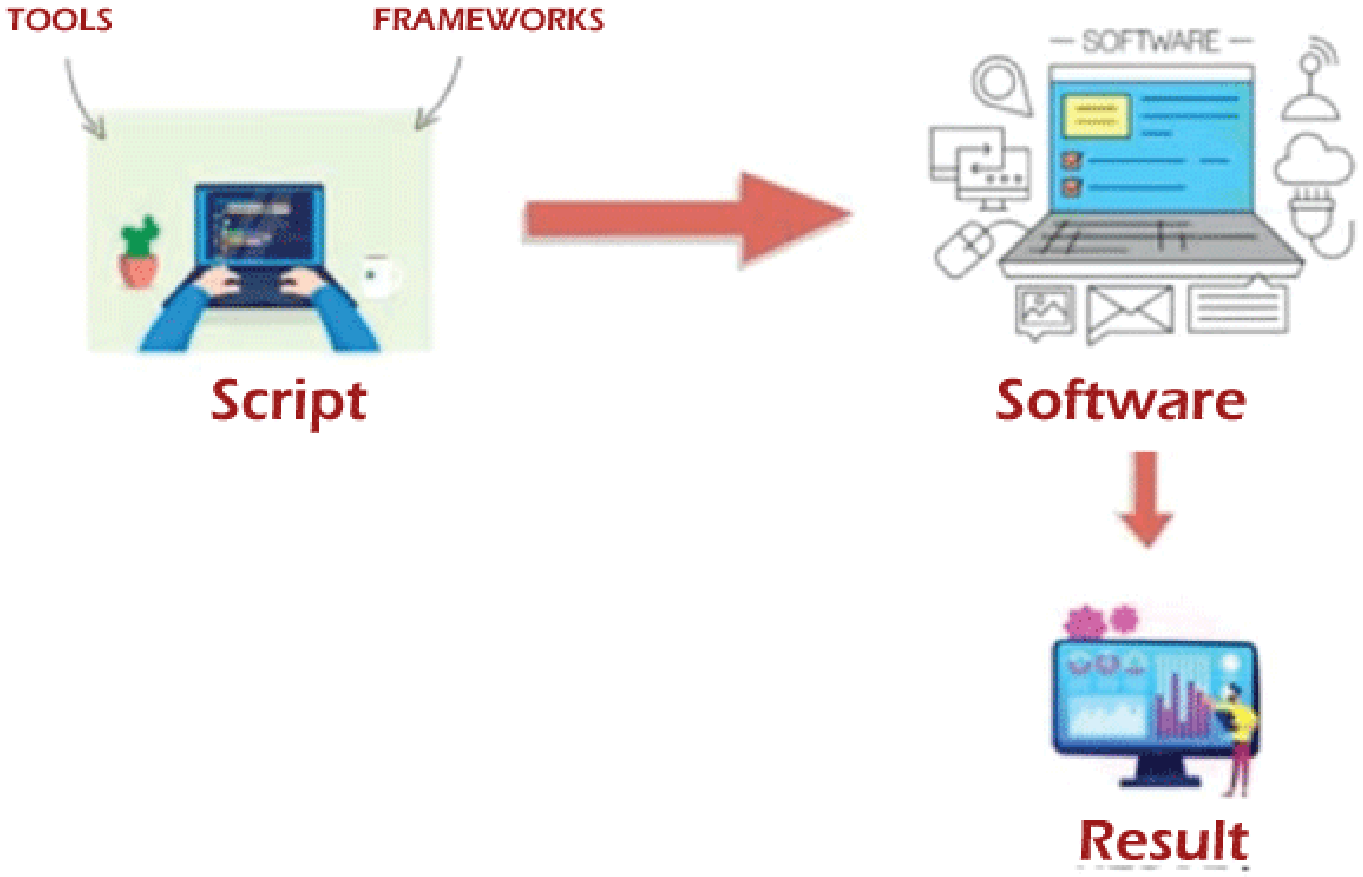
Aspects	Automation testing	Manual testing
Definition	When an application or software is tested with the help of some tools is known as automation testing. Whenever multiple releases or multiple regression cycle is going on the application or software, we will go for automation testing.	It is a type of software testing, which is done by the test Engineer to check the functionality of an application based on the customer requirement.
Reliability	It is reliable because it tests the application with the help of tools and test scripts.	It is not reliable because there is a possibility of human error, which may not be delivered the bug-free application.

Aspects	Automation testing	Manual testing
Reused	The script can be reused across multiple releases.	It could be possible when the test case only needs to run once or twice.
Batch Execution	Batch execution is possible using automation testing because all the written scripts can be executed parallelly or simultaneously.	Batch execution is not possible in manual testing.
Time-saving	The execution is always faster than the manual; that's why the automation testing process is time-saving.	It is time consuming due to the usage of the human resources.
Investment	While using the Automation tool, investment is required.	Human resources needed investment.
Performance testing	To test the performance of the application with the help of load and stress testing, automation test engineer needs to perform Performance Testing.	In manual testing, performance testing is not possible.

Aspects	Automation testing	Manual testing
Programming knowledge	Without having an understanding of programming language, we cannot write the test script.	There is no need to know programming language but should have the product knowledge to write the test case.
Framework	The automation test engineer can use the different types of frameworks like **Data driven, Hybrid, modular, and keyword-driven** to faster the automation process.	There is no need for a framework while using manual testing.
Operating system compatibility	Automation testing can also be performed on different systems with different operating system platforms and various programming languages.	Operating system compatibility is not possible in manual testing because the different tester is required to perform such tasks.
Regression testing	Whenever the code changes happen due to the enhancement of the release, then automation test engineer performs the regression testing.	When the test engineer executes the test case for the first time, it may be useful, but there is a possibility that it will not catch the regression bugs because of changing requirements frequently.

What is Automation Testing

- Another software testing method is automation testing, which is used some specific tools to execute the test scripts without any human interference. It is the most acceptable way to enhance the efficiency, productivity, and test coverage of Software testing.
- With the help of an automation testing tool, we can easily approach the test data, handle the test implementation, and compares the actual output against the expected outcome.



TOOLS

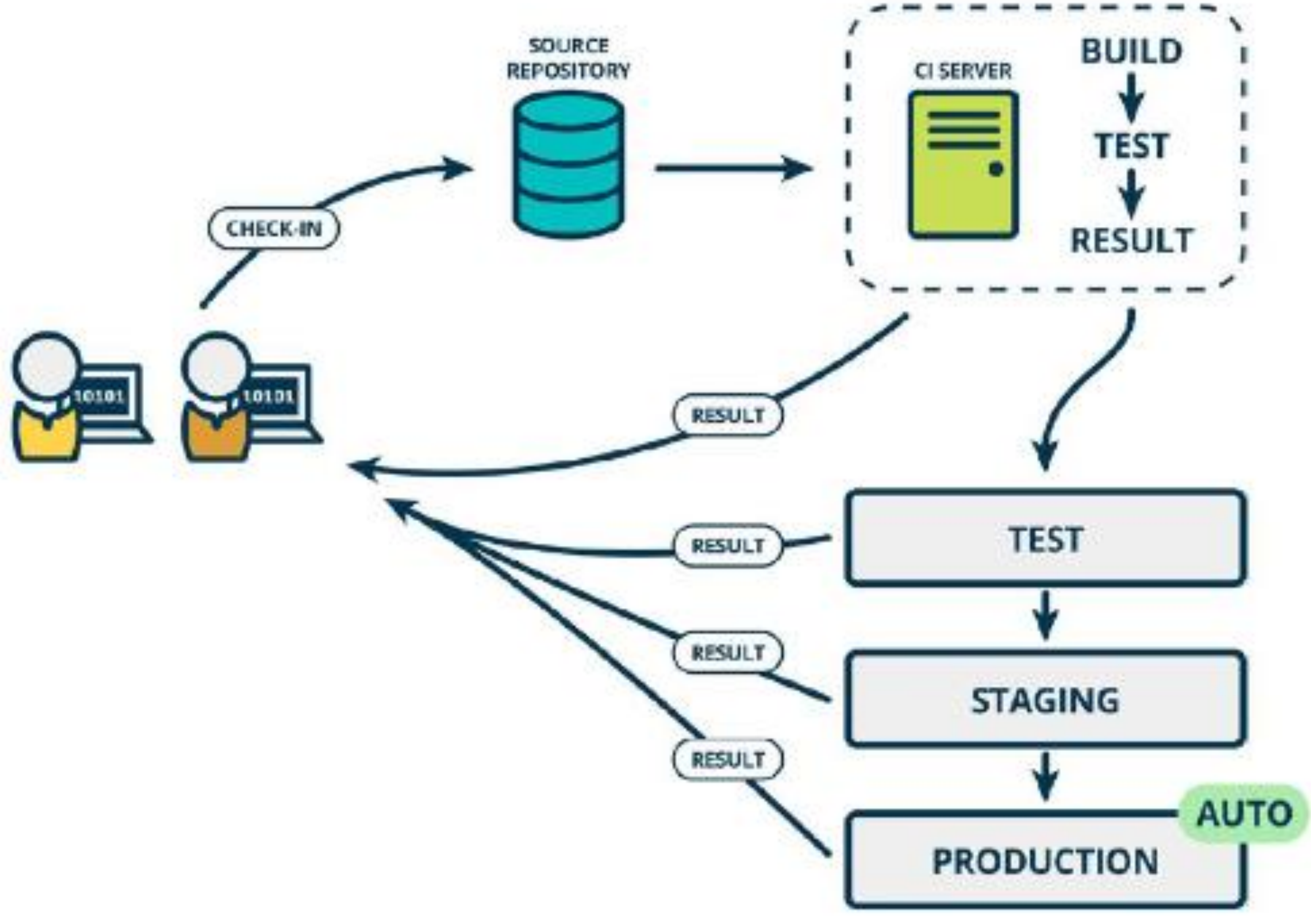
FRAMEWORKS

Script

Software

Result

Automated Testing



Modern Automated Testing

- **Automation testing** refers to the automatic testing of the software in which developer or tester write the test script once with the help of testing tools and framework and run it on the software. The test script automatically test the software without human intervention and shows the result (either error, bugs are present or software is free from them).
- Automation testing needs manual effort when creating initial scripts, and further process is performed automatically to compare the actual testing result with expected results.
- In **automation testing**, the test automation engineer will write the test script or use the automation testing tools to execute the application. On the other hand, in manual testing, the test engineer will write the test cases and implement the software on the basis of written test cases.
- In test automation, the **test engineer** can execute repetitive tasks and other related tasks. In manual testing, it is a tedious process to implement the repetitive take again and again.
- In other words, we can say that the main concentration of **test Automation** is to change the manual human activity with systems or devices.
- The **automation testing** process is a time-saving process as it spends less time in exploratory testing and more time in keeping the test scripts whereas enhancing the complete test coverage

Why do we perform Automation Testing?

- In software testing, automation testing is required to test the application because it offers us a better application with less effort and time.
- To implement the automation testing, we required pretty a considerable investment of resources and money.
- The execution of automation testing provides us various advantages, which are as discussed below:
  - **Reusability**
  - **Consistency**
  - **Running tests anytime (24/7)**
  - **Early Bug detection**
  - **Less Human Resources**

1. Reusability

We can re-use the test scripts in automation testing, and we don't need to write the new test scripts again and again. And, we can also re-create the steps which are detailed as the earlier ones.

2. Consistency

As compared to manual testing, automation testing is more consistent and way faster than executing the regular monotonous tests that cannot be missed but may cause faults when tested manually.

3. Running Tests 24/7

In automation testing, we can start the testing process from anywhere in the world and anytime we want to. And even we can do that remotely if we don't have many approaches or the option to purchase them.

4. Early Bug Detection

We can easily detect the critical bugs in the software development process's initial phases by executing automation testing. It also helps us spend fewer working hours to fix these problems and reduce costs

5. Less Human Resources

To implement the automation test script, we need a test automation engineer who can write the test scripts to automate our tests, rather than having several people who are repeatedly performing the tedious manual tests.

Test automation

- Test automation is the process of using software tools to execute pre-scripted tests on a software application before releasing it into production. Automation testing ensures that the software meets the required quality standards, reduces the time-to-market, and saves costs.
- The development process of test automation involves identifying the test cases that need to be automated, selecting the automation tool, creating the automation scripts, executing the scripts, and analyzing the results.

Benefits of Test Automation

- Test automation provides numerous benefits such as improved accuracy, increased efficiency, and faster feedback. Automated tests can be executed repeatedly, ensuring that the software works as expected after each change.
- Automation testing also helps in identifying defects early in the development cycle, reducing the cost of fixing them at later stages. Additionally, automation testing enables better collaboration between developers and testers by providing a common language for discussing issues.

Challenges in Test Automation

- Although test automation offers several advantages, there are also some challenges associated with it. One of the biggest challenges is selecting the appropriate automation tool for the project. The tool should be capable of testing all the features of the software and should integrate well with other tools used in the development process.
- Another challenge is maintaining the automation scripts. As the software evolves, the scripts must be updated to reflect the changes. This requires significant effort and may result in increased maintenance costs.

Software Quality Achievement through Automation

- Test automation plays a critical role in achieving software quality. It ensures that the software meets the required functional and non-functional requirements and performs as expected under different conditions.
- Automated testing also helps in identifying performance issues, security vulnerabilities, and other defects early in the development cycle, reducing the risk of releasing faulty software into production. Overall, test automation is an essential component of the software development process that helps in achieving high-quality software.

Best Practices for Test Automation

- To ensure successful test automation, it is essential to follow best practices such as identifying the right test cases for automation, selecting the appropriate automation tool, creating modular and reusable automation scripts, and using **version control** for managing the scripts.
- It is also important to maintain a balance between manual and automated testing, as both have their own advantages and limitations. Finally, it is crucial to continuously evaluate and improve the automation process to ensure that it remains effective and efficient.

Conclusion-Test Automation

- Test automation is a vital part of the software development process that helps in achieving high-quality software. Although there are some challenges associated with automation testing, following best practices can help in overcoming them.
- By identifying the right test cases, selecting the appropriate automation tool, creating modular and reusable scripts, and continuously evaluating and improving the process, organizations can achieve significant benefits such as improved accuracy, increased efficiency, and faster feedback.

Quality Assurance vs Quality Control

Software quality assurance is (also known as QA) a sequence of tasks to prevent defects and ensure that the techniques, methods, approaches, and processes are designed for a specific application must be implemented correctly. This is an ongoing process within the development of a software system.

The development of units of an application is checked under the quality assurance specifications in the sequence of their development.

Quality assurance test ensures the development of high-quality software because of its main focus on the high-quality processes, good quality management system and periodic conformance audit during the development of software. It is a managerial tool includes planned and systematic activities and documentation to prevent problems related to quality.

The responsibility of quality assurance is not of any specific team, but it is a responsibility of each member of the development team.

1. Quality assurance prevents defects.
2. Quality assurance is process oriented.
3. Quality assurance is proactive in a process and preventive in nature.
4. Quality assurance is a managerial tool.
5. Each developer is responsible for quality assurance.

Quality Control

Quality Control also known as QC is a sequence of tasks to ensure the quality of software by identifying defects and correction of defects in the developed software. It is a reactive process, and the main purpose of this process is to correct all types of defects before releasing the software. The process is done by eliminating sources of problems (which cause to low the quality) through the corrective tools so that software can meet customer's requirements and high quality.

The responsibility of quality control is of a specific team which is known as a testing team that tests the defects of software by validation and corrective tools.

1. Quality Control provides identification of defects.
2. Quality Control is product oriented.
3. Quality Control is a corrective tool.
4. Testing team is responsible for Quality control.
5. Quality Control is a reactive process.

Difference between Quality Assurance and Quality Control

Points	Quality Assurance	Quality Control
Definition	QA is a group of activities which ensures that the quality of processes which is used during the development of the software always be maintained.	QC is a group of activities to detect the defects in the developed software.
Focus	The focus of QA is to **prevent defects** in the developing software by paying attention to processes.	The focus of QC is to **identify defects** in the developed software by paying attention to testing processes.
How	Establishment of the high-quality management system and periodic audits for conformance of the operations of the developing software.	Detecting and eliminating the quality problem elements by using testing techniques and tools in the developed software.
What	QA ensures prevention of quality problem elements by using systematic activities including documentation.	QC ensures identification and elimination of defects by using processes and techniques to achieve and maintain high quality of the software.

Points	Quality Assurance	Quality Control
Orientation	QA is **process oriented**.	QC is **product oriented**.
Type of process	QA is a proactive process. It concerns to improve development so; defects do not arise in the testing period.	QC is a reactive process because it concerns to identify defects after the development of product and before its release.
Responsibility	Each and every member of the development team is responsible for QA	Only the specific testing team is responsible for QC

Test Automation Framework:

Architecture & Types

What is Framework in Automation Testing?

A Test Automation Framework is a set of guidelines like coding standards, test-data handling, object repository treatment etc… which when followed during automation scripting produces beneficial outcomes like increased code re-usage, higher portability, reduced script maintenance cost etc. These are just guidelines and not rules; they are not mandatory and you can still script without following the guidelines. But you will miss out on the advantages of having a Framework.

Types of Test Automation Frameworks

Below are the different types of Automated Testing Frameworks:

1) Linear Scripting

2) The Test Library Architecture Framework.

3) Module Based Testing Framework

4) The Data-Driven Testing Framework.

5) The Keyword-Driven or Table-Driven Testing Framework.

6) The Hybrid Test Automation Framework.

1) Linear Scripting – Record & Playback

- Linear Scripting Framework is a basic level test automation framework that is in the form of ‘Record and Playback’ in a linear fashion. This framework is also known as the ‘Record and Playback’ framework. This type of framework is used to test small-sized applications.
- In this type, the creation, and execution of test scripts are done individually for each test case individually.
- Testers capture each test step such as browsing, navigation, user inputs, enforcing checkpoints. Testers then play the scripts to carry out the tests.

1) Linear Scripting – Record & Playback

Advantages of Linear Scripting Automation Framework:

- Can generate test scripts (Record and playback) without planning much or consume much time
- Coding knowledge is not required
- A quick way to generate test scripts

Disadvantages of Linear Scripting Automation Framework:

- Lack of reusability due to auto-generated scripts
- Hard coding the data doesn’t allow us to run with multiple data sets
- Maintenance is high – It requires a lot of effort to do even small changes.

2) The Library Architecture Framework

- Library Architecture Testing framework - “Structured Scripting” or “Functional Decomposition” It is based on the modular framework with some additional advantages.
- In the modular testing framework, we divide the application under test into modules whereas here we identify the common tasks and grouped them into functions. Once the functions are grouped then these groups will be kept in a library. The test scripts reuse these libraries to create new test cases.

Advantages of a Library Architecture Testing Framework:

- Script maintenance is simple
- Easy to scalable
- Functions library is reusable and it can be reusable

Disadvantages of a Library Architecture Testing Framework:

- Coding skills are required
- It takes more time to prepare test scripts
- A fixed set of test data is hardcoded within the scripts

3) Module Based Testing Framework

- Module based Testing Framework is based on one of the popularly known OOPs concept – Abstraction. The framework divides the entire "Application Under Test" into a number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts took together builds a larger test script representing more than one modules.
- These modules are separated by an abstraction layer in such a way that the changes made in the sections of the application doesn't yield affects on this module.

3) Module Based Testing Framework

Advantages:

- The framework introduces the high level of modularization which leads to easier and cost-efficient maintenance.
- The framework is pretty much scalable
- If the changes are implemented in one part of the application, only the test script representing that part of the application needs to be fixed to leave all the other parts untouched.

Disadvantages:

- While implementing test scripts for each module separately, we embed the test data (Data with which we are supposed to perform testing) into the test scripts. Thus, whenever we are supposed to test with a different set of test data, it requires the manipulations to be made in the test scripts.

4) The Data-Driven Testing Framework

- The data-driven test automation framework is focused on separating the test scripts logic and the test data from each other. It allows us to create test automation scripts by passing different sets of test data.
- The test data set is kept in the external files or resources such as MS Excel Sheets, MS Access Tables, SQL Database, XML files, etc.,
- The test scripts connect to the external resources to get the test data. By using this framework we could easily make the test scripts work properly for different sets of test data.
- This framework significantly reduces the number of test scripts compared to the module-based framework. This framework gives more test coverage with reusable tests and flexibility in the execution of tests only when required and by changing only the input test data.
- It is reliable in terms of no impact on tests by changing the test data but it has its own drawbacks such as testers who work on this framework needs to have the hands-on programming knowledge to develop test scripts

4) The Data-Driven Testing Framework

Advantages of a Data-Driven Framework:

- It supports multiple data sets
- Modifying the test scripts won’t affect the test data
- No need to hardcode test data
- Saves time by executing more tests

Disadvantages of a Data-Driven Framework:

- Require coding skills
- Setting up the framework and test data takes more time
- Need experienced automation testers to design framework

5) Keyword Driven Testing Framework

- It is also known as table-driven testing or action word-based testing. In Keyword-driven testing, we use a table format to define keywords or action words for each function or method that we would execute.
- It performs automation test scripts based on the keywords specified in the Excel sheet.
- By using this Framework, testers can work with keywords to develop any test automation script, testers with less programming knowledge would also be able to work on the test scripts.
- The logic to read keywords and call the required action mentioned in the external Excel sheet is placed in the main class. Keyword-driven testing is similar to data-driven testing.
- Even though to work on this framework doesn’t require much programming skills but the initial setup (implementing the framework) requires more expertise.

5) Keyword Driven Testing Framework

Advantages of Keyword-Driven Frameworks:

- No need to be an expert to write test scripts
- It is possible to reuse the code. We can point the different scripts to the same keyword
- Even though application changes, test scripts don’t change.
- Tests can be designed before developing the application
- Test scripts work independently of an application under test with basic modifications
- Not dependent on test tools

Disadvantages of Keyword-Driven Frameworks:

- Take more time to design
- The initial cost is high
- Employees with good test automation skills needed

6) Hybrid test automation Testing Framework

Hybrid Test automation framework is the combination of two or more frameworks mentioned above. It attempts to leverage the strengths and benefits of other frameworks for the particular test environment it manages. Most of the teams are building this hybrid driven framework in the current market.

Quality Certificate

A quality certificate is a document issued by an institution outside the company and which certifies that its production processes meet the quality parameters necessary to go on the market

The quality certificate is a means of verification of the processes that a company has developed for the development and / or production of its products and services, which certifies compliance with appropriate quality standards to be offered to consumers. These standards can adhere to a standard, or be specific to the entity that grants the certificate.

International quality certificates

An institution that has established quality standards, is the International Organization of Standardization (International Organization for Standardization, better known as ISO), who established parameters for their quality management certificates in ISO 9001, which emphasizes the management of quality for the production of goods and services.

- It should be noted that the companies that undergo the review of their processes to obtain this certificate, do so voluntarily, since they are international reference standards that provide guarantees on their production management, and not a legal regulatory requirement.

Characteristics of the quality certificates

- As mentioned above, the quality certificates are voluntary and applied by entities outside the company, independent and autonomous. They must ensure compliance with significant quality parameters that are transversal to economic and industrial activities.
- Some of its characteristics:
    - They are applied to a specific process related to the elaboration and / or production of goods and services.
    - They adhere to one or more quality standards (it can be ISO 9001).
    - They provide recommendations in the event that the submitted institution does not approve the quality management processes.
    - They are extended for a set period, which is why they must be monitored from time to time.
    - They indicate the data of the certifying institution.

Benefits of having a quality certificate

- Quality certificates provide guarantees to companies and reliability to their clients, making it an excellent tool to boost business prestige and reputation. This allows companies to implement continuous improvement processes in their organizations and thereby position themselves by the quality of the products they offer.
- In addition, if the company wants to have guidelines to approach optimal quality management, submitting to a review by an expert institution will help it to obtain feedback on what to correct and thus work with a clear objective.

Common Quality Management Frameworks

- **ISO 9001:** The most commonly used set of requirements for designing a QMS. **QMS certification** against ISO 9001 is recognized worldwide.
- **AS9100:** This is a standard that is based on ISO 9001 and has additions designed for use in the Aerospace Industry. A QMS can be certified by a third party to comply with this standard.
- **ISO 13485:** This is a standard published by the ISO for use by companies that want to design a QMS for medical devices, and the requirements for regulatory purposes surrounding them.

The most common quality management frameworks

ISO 9001

AS9100

ISO 13485

- **Lean:** The core idea is to maximize value by eliminating waste. The main idea is that anything that adds cost to a product, but not value, is waste and should be controlled or eliminated. This supports a QMS, but cannot be used to design or certify a QMS.
- **MBNQA:** The Malcolm Baldridge National Quality Award recognizes US organizations for performance excellence. The award has a set of requirements against which a company could design and assess a QMS, but apart from the award there is no ongoing certification against these requirements.
- **Six Sigma:** This is a set of tools and techniques used for process improvement. It is used in many organizations to support the QMS by helping to improve processes, but Six Sigma does not define a QMS, and the QMS cannot be certified to Six Sigma.
- **TQM:** Total Quality Management consists of practices designed to improve the process performance of a company. The techniques help improve efficiency, problem solving and standardization of processes. These techniques are used to aid in quality management, but do not provide a framework for a Quality Management System, and cannot be certified to.
- **IATF 16949:** This document includes requirements for the application of ISO 9001 for automotive production and service part organizations. A QMS designed using these requirements can also be certified against them.

Selenium Architecture

- **What is Selenium?**
- **What is Selenium Architecture?**
  - Selenium IDE
  - Selenium RC
  - Selenium WebDriver
  - Selenium Grid
- **Why is Selenium so preferred?**
  - Browser Compatibility
  - Language Compatibility
  - Prompt Implementation
- **Top Features of Selenium**
- **How to use Selenium for Web Automation?**
- Conclusion

Selenium Architecture

What is Selenium?

- Selenium is an automation testing tool or to be precise, a framework. It is an open-source framework that has been designed for the automation testing of web applications. Also, Selenium is a flexible testing tool that allows the automation tester to write testing scripts in Selenium in various programming languages such as Python, Java, etc.
- Selenium supports various web browsers such as Safari, Firefox, Opera, and Chrome wherein the Selenium test scripts, written in different languages can be run easily. It also supports cross-platform browsing, i.e. the test cases can be run simultaneously across different platforms. The platforms supported by Selenium are Windows, Mac OS, Linux, and Solaris. Selenium has become the top automation testing tool as it allows developers to create robust and flexible automation suits.
- Every day, thousands of applications are deployed to the web. Now the testing teams have to be always ready to ensure that these applications are performing up to the mark even outside the development environment. To perform this testing, a user-friendly and robust framework is needed. Selenium's amazing suite has helped millions of app deployments easy. To understand how this has been possible, we will have to understand the architecture of Selenium.

Selenium Architecture

- The Selenium architecture is comprised of the following components:
  - Selenium IDE
  - Selenium RC
  - Selenium WebDriver
  - Selenium Grid

Selenium IDE

Selenium IDE stands for Selenium Integrated Development Environment. It was created by Shinya Kasatani of Japan and later went through further development. Selenium IDE is a Firefox plugin that allows the developers to record and playback the scripts. One of the advantages of Selenium IDE is that it does not require any programming knowledge. Simply knowing HTML and DOM would suffice. Usually, the Selenium IDE is used as a prototyping tool because of its simplicity.

Selenium RC

Selenium RC stands for Selenium Remote Control. It is sometimes referred to as Selenium 1 because it was the first flagship testing framework and remained a preferred web automation testing tool for a long time. The Selenium RC bases its automation functions on Javascript. The languages supported by Selenium RC are Python, Perl, Ruby, Java, C#, and PHP. Also, almost all the available web browsers out there are supported by Selenium RC. It is to be noted though that Selenium RC has been officially deprecated.

Selenium WebDriver

- Selenium Web driver is an automation framework that allows automating user actions with modern-age web browsers and communications with the browsers through a set of open-source APIs. The implementation of Selenium WebDriver is carried out through a browser-centric drive.
- **Operating systems supported by Selenium WebDriver:** Windows, Mac OS, Linux, Solaris
- **Programming languages supported by Selenium WebDriver are:** Java, C#, PHP, Python, Perl, Ruby, and Javascript
- **Web browsers supported by Selenium WebDriver are**: Mozilla Firefox, Internet Explorer, Google Chrome 12.0.712.0 and above, Safari, Opera 11.5 and above, Android, iOS, HtmlUnit 2.9 and above.
- **The architecture of Selenium WebDriver consists of the following:**
  - Selenium Client Library
  - JSON WIRE PROTOCOL Over HTTP Client
  - Browser Drivers
  - Browsers

1- Selenium Client Library

The Selenium Client Library consists of various language libraries for Java, Ruby, Python, and other supported languages.

2- JSON WIRE PROTOCOL Over HTTP Client

JSON denotes Javascript Object Notation. This component of the Selenium WebDriver plays an important role in the Selenium automation process by transferring data between the server and a client on the web.

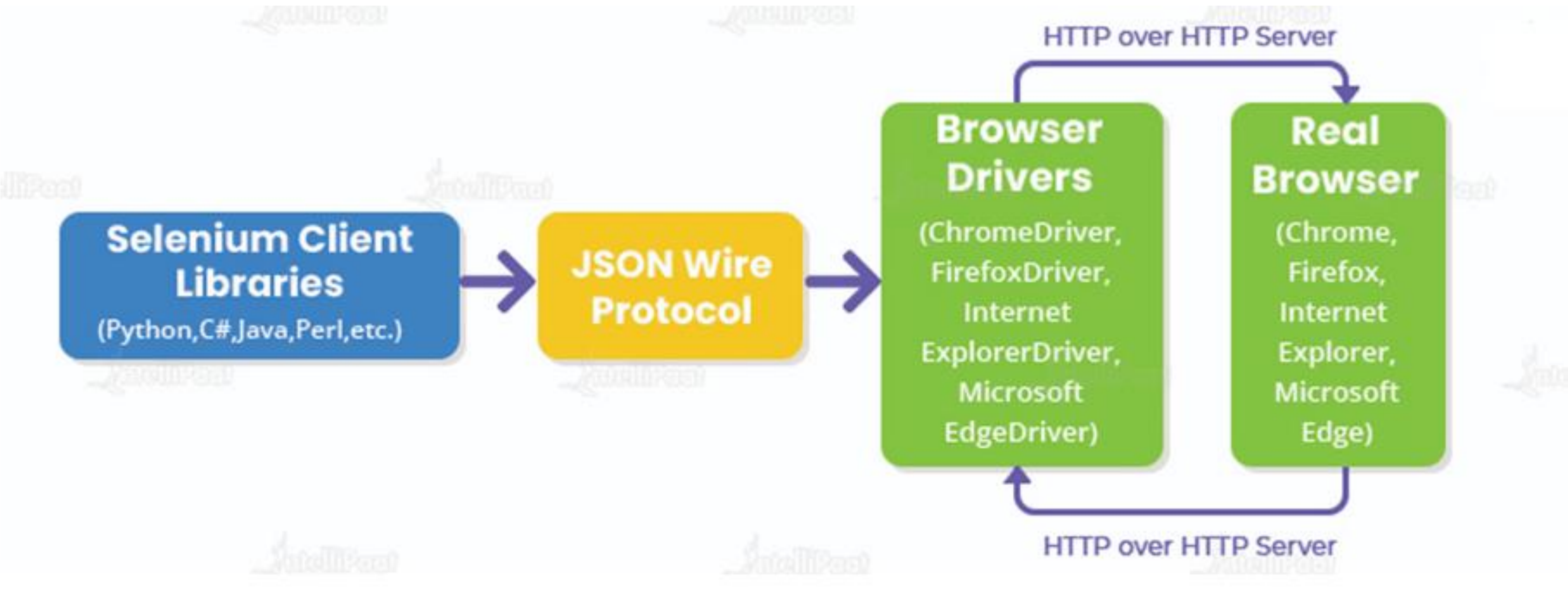
3- Browser Drivers

Browser drivers are used to carry out the communication between the Selenium WebDriver and the respective browser. The Browser drivers ensure that no details are revealed to the browser regarding the internal logic of the functionalities of the browser.

4- Browsers

As already discussed above, the browsers supported are Firefox, Safari, Chrome, and more.

Selenium WebDriver Architecture Diagram



HTTP over HTTP Server
Selenium Client Libraries
(Python,C#,Java,Perl,etc.)
JSON Wire Protocol
Browser Drivers
(ChromeDriver, FirefoxDriver, Internet ExplorerDriver, Microsoft EdgeDriver)
Real Browser
(Chrome, Firefox, Internet Explorer, Microsoft Edge)
HTTP over HTTP Server

Selenium Grid

- This component of the Selenium suite works together clubbed with the Selenium RC. It is used to run parallel tests on machines against their respective browsers. Selenium. The working of Selenium Grid depends on the browsers and operating systems supported by the entire framework. Since almost all browsers and operating systems are supported by Selenium, it is easier for the Selenium Grid to run multiple tests at the same time against different machines with different browsers.

Why is Selenium so preferred?

From the discussed features offered by Selenium and its architecture, you might already have an idea as to why it is so popular. Now, let us discuss in depth why Selenium is a top preferred automation testing framework.

Browser Compatibility

Selenium’s compatibility with almost all the available browsers such as Firefox, Safari, Opera, Chrome, etc. makes it a favorite tool among automation testers and developers. Also, the cross-browser compatibility offered by Selenium WebDriver is the primary reason why it is preferred.

Language Compatibility

Selenium offers ample freedom to write codes and test scripts in various languages. It is not possible for testers to learn almost every programming language. Hence its all-language compatibility feature has made Selenium a loved automation testing tool.

Prompt Implementation

Due to little or almost no dependency on middle servers, the process of automation testing is very fast. No middleware servers are required to communicate with the browsers. Also, the JSON is lightweight which adds to the fast execution of automation testing.

Features of Selenium

Some of the features offered by the Selenium suite are mentioned below:

Cross-Browser Testing- Selenium allows conducting tests on the same website with the same scripts and codes on different browsers.

Parallel testing- The scripts on one or multiple browsers can be carried out simultaneously.

Modern Technology support- In Selenium, all types of applications can be tested irrespective of the technology used in those applications.

Web Page Automation- Even those web pages can be automated through Selenium in which the content varies based on the user's actions.

How to use Selenium for Web Automation?

The following steps can be followed to run Selenium test scripts :

- The first step involves creating an instance of WebDriver for the respective Browser
- Now move to the web page which needs to be automated
- Fix an HTML element on the Web page
- Perform a random action on the HTML element. You can use the 'Sendkeys' method to perform the action.
- Run the automation testing on the page and record the results using a test framework.

Thank You!!