Module-5

1. **What is Automation testing?**

* **Automation Testing** is a software testing technique that uses specialized tools or scripts to automatically execute test cases, compare actual outcomes with expected results, and report defects.
* It is primarily used to increase testing efficiency, speed, and accuracy, especially for **repetitive** and **regression** testing tasks.

**Why Use Automation Testing?**

1. **Saves Time and Effort** – Automates repetitive test cases.
2. **Improves Accuracy** – Reduces human error.
3. **Faster Feedback** – Speeds up the development and deployment cycle.
4. **Supports Continuous Testing** – Essential in **DevOps** and **CI/CD pipelines.**
5. **Reusable Test Scripts** – Once created, test scripts can be reused for future testing cycles.

**Common Tools for Automation Testing**

* **Selenium** – Web application testing.
* **JUnit/TestNG** – Unit testing in Java.
* **Appium** – Mobile app testing.
* **Postman/Newman** – API testing.
* **Cypress** – Modern web testing.
* **Playwright** – End-to-end browser testing.

**Types of Tests That Can Be Automated**

* Regression Tests
* Smoke Tests
* Sanity Tests
* Load & Performance Tests
* **UI Tests** (Web and Mobile)

**When Not to Use Automation**

* When the test cases are **newly designed** or **change frequently**
* For **short-term projects**
* For **usability** or **exploratory testing**

1. **Which Are The Browsers Supported By Selenium Ide?**

**Browsers Supported by Selenium IDE:**

Selenium IDE is a browser-based test automation tool that supports the following browsers:

1. Google Chrome
2. Mozilla Firefox
3. Microsoft Edge (via chrome web store)

**Browsers NOT supported by Selenium IDE:**

* Safari
* Internet Explorer
* Opera

1. **What are the benefits of Automation Testing?**

* Automation testing provides many advantages over manual testing, especially for large and complex applications. Here are the key benefits:

1. **Faster Execution**

* Automated tests run much faster than manual testing.
* Ideal for regression testing and frequent test runs during development.

1. **High Test Coverage**

* You can run hundreds or thousands of test cases across different platforms and environments quickly.
* Increases test coverage and improves product quality.

1. **Reusability of Test Scripts**

* Automated test scripts can be reused across different versions of the application.
* Easy to update and maintain tests when changes occur.

1. **Reduced Human Error**

* Removes the risk of manual mistakes during repetitive testing.
* Ensures consistency and accuracy in test execution.

1. **Saves Time and Cost in the Long Run**

* Although initial setup can be time-consuming, it saves time and cost over time due to repeatable test execution.
* Useful in Agile and DevOps environments.

1. **Supports Continuous Integration / Delivery (CI/CD)**

* Automation tools integrate well with CI/CD pipelines.
* Enables frequent releases with better test confidence.

1. **Easy Reporting**

* Tools generate detailed test logs and reports automatically.
* Helps in quick identification of failed cases and faster debugging.

1. **Parallel Execution**

* Tests can be run on multiple devices, browsers, or environments simultaneously.
* Speeds up overall test cycles significantly.

1. **Better Resource Utilization**

* Frees up testers from repetitive tasks so they can focus on exploratory and usability testing.

1. **What are the advantages of Selenium?**

* Of course, here are the main advantages of Selenium in a simple, short format:
* **Free to Use:** It's open-source, so there are no licensing costs.
* **Language Flexibility:** You can write test scripts in many popular languages like Python, Java, C#, and JavaScript.
* **Cross-Browser Support:** Your tests can run on all major web browsers like Chrome, Firefox, and Safari.
* **Works on Any OS:** It is compatible with different operating systems, including Windows, macOS, and Linux.
* **Parallel Testing:** It can run multiple tests at the same time to speed up the testing process.
* **Strong Community:** It has a large and active community, making it easy to find help and resources online.

1. **Why testers should opt for Selenium and not QTP?**

* Testers should opt for **Selenium over QTP** for several strong reasons, especially when working with web-based applications.

1. **Open Source and Free**

* **Selenium:** Completely **open-source** and free to use.
* **QTP/UFT:** **Commercial tool** developed by Micro Focus. Requires **expensive licenses,** which may not be suitable for startups or small teams.

1. **Language Support**

* **Selenium:** Supports **multiple programming languages** like **Java, C#, Python, Ruby, JavaScript, Kotlin.**
* **QTP/UFT:** Supports only **VBScript**, which limits flexibility.

1. **Platform and Browser Compatibility**

* **Selenium:** Can run on **multiple operating systems** (Windows, macOS, Linux) and supports **all major browsers** like Chrome, Firefox, Edge, Safari.
* **QTP/UFT:** Mostly supports **Windows OS** and **limited browsers** (primarily Internet Explorer and Edge).

1. **Web Application Testing**

* **Selenium:** Best suited for **web applications**. It automates across all browsers effectively.
* **QTP/UFT:** Supports desktop and some web applications, but not as strong or fast in cross-browser testing.

1. **Community and Support**

* **Selenium:** Has a **huge community,** lots of online tutorials, forums, and active development.
* **QTP/UFT:** Smaller community with **limited free resources.**

1. **Integration with CI/CD Tools**

* **Selenium:** Easily integrates with **Jenkins,Git, TestNG,** and other DevOps tools.
* **QTP/UFT:** Can integrate, but **more complex** and **less flexible** compared to Selenium.

1. **Parallel & Cross Browser Testing**

* **Selenium Grid** allows **parallel execution** and **cross-browser testing**, which saves time.
* **QTP/UFT** doesn't support this out-of-the-box; it’s **slower and more resource-intensive.**

1. **Flexibility and Customization**

* **Selenium:** Highly customizable and works with frameworks like **TestNG, JUnit, Cucumber, Allure**, etc.
* **QTP/UFT:** More rigid; customization is **limited by the tool’s built-in structure.**