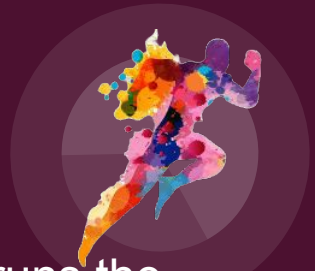
TEST AUTOMATION

- Test Automation.
- Playwright.
- Playwright vs Selenium
- Conclusion

What is automated testing?



- The principle of automated testing is that there is a program that runs the program being tested, feeding it the proper input, and checking the output against the expected output.
- Once the test suite is written, no human intervention is needed, either to run the program or to look to see if it worked; the test suite does all that, and indicates whether the program's output was as expected.
- The main focus of automation is improving the execution portion of testing life cycle. In order to perform automated testing various categories of tools are used.
- Tools: The efficient and time saving way of testing are through the usage of testing tools.
- The tools are used to continuously improve the quality of testing as well as it also helps in reusability and faster performance.

Why Automation Tools?

- The usage of tools make testing easier, faster and more reliable
- To perform repeated execution of test scripts
- Beneficial when the script keeps on changing (regression)
- Allows to perform a fast test execution on different OS platforms
- Allows to execute the test scripts on different machines with different configuration simultaneously
- Automation is an advantage in performance testing, when the script has to be executed for many users together
- Tools are mainly used to facilitate processes
- Helps to uncover those defects that are created accidentally or inadvertently
- Provides highest test coverage limits
- Runs all day and night in an unattended mode.

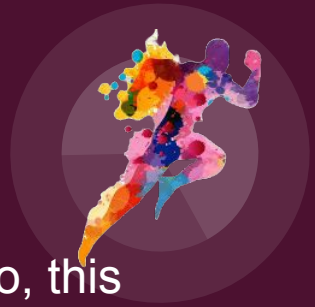
Why Automation Tools?

- System continues running even if a test case fails.
- Write out meaningful logs.
- One point maintenance.
- Easy to update reusable modules.
- Text strings stored in variables easy to find and update.
- Automated most important business functions first.
- Quickly add scripts and modules to the system for new features.
- Don’t waste time with very complex features, keep it simple.
- Track components of the automated testing system in a database.
- Test case management - store test cases in a database for maintenance purposes
- Track tests that pass, as well as test that fail.

When to Automate

- **Mainly automation comes in to picture at the level of System testing. Automation is only possible in stages when the application works 100%.**
- **The primary purpose of an automated test is to verify that a requirement, once validated, functions properly in successive builds or modifications of the AUT (Application Under Test).**
- **Furthermore, because automated testing reduces the time necessary to perform regression testing, this is where its benefits are realized the most.**
- **What should be automated**
  - ✓ Automation should be performed mainly to test the critical paths of the AUT.
  - ✓ First we should automate the primary functions that will be performed by the targeted end-users.
  - ✓ Slowly, we should add the not-so-critical portions of the application as time permits. The scripts should be designed in such a way that they are flexible and easy to update.
  - ✓ Early in the project we should not automate testing of such things as login, user preferences, or other options, status bars, help screens, or any other areas of the application that development will pay little attention to until later in the project.

How to conduct Automation



- The first thing to be done here is to identify an appropriate tool. So, this can be achieved after a tool evaluation is performed.
- This tool evaluation should be conducted by using two or three tools to automate some complex and easy scenarios; this activity should also include the people who are going to participate in automation.
- An appropriate structured testing methodology should be designed because the automated testing methodology is independent of the tool and the tool’s main role is to support that methodology.

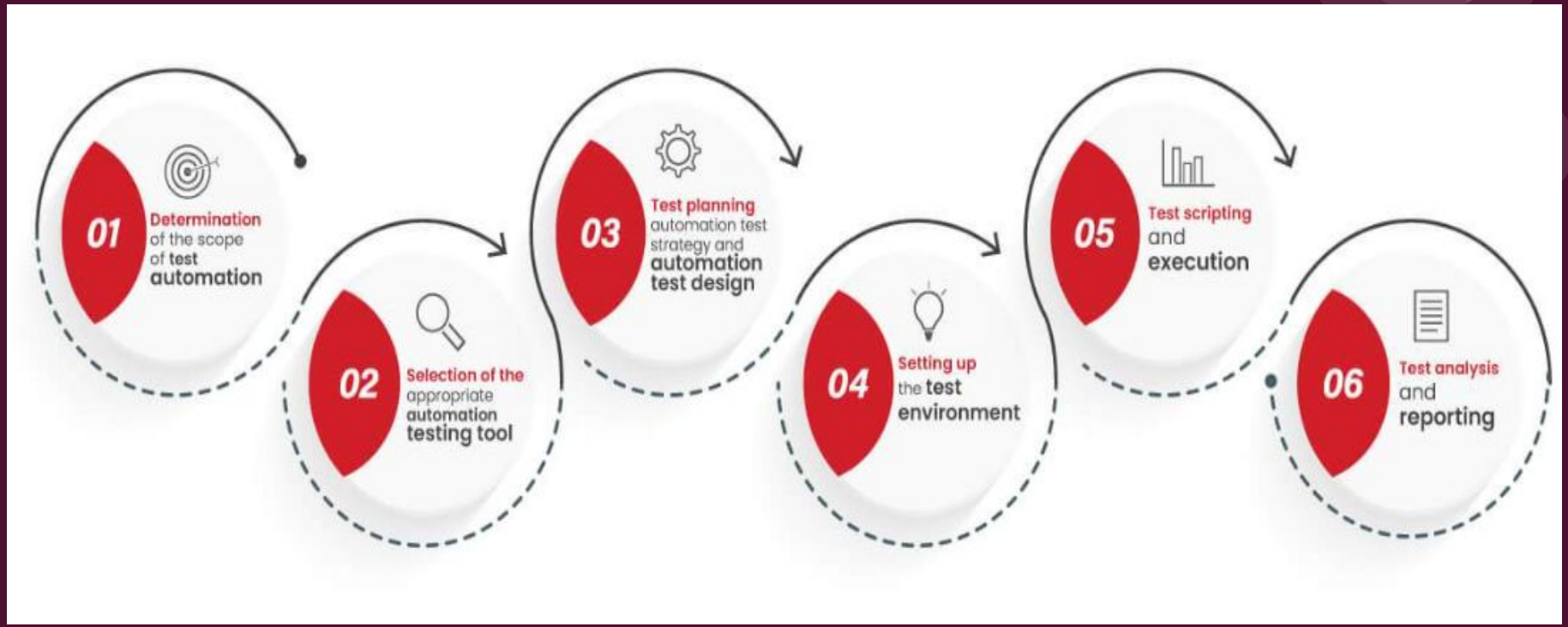
Automated Testing Methodologies

- **Reusable modules**
  - ✓ The basic building block is a reusable module.
  - ✓ These modules are used for navigation, manipulating controls, data entry, data validation, error identification (hard or soft), and writing out logs.
  - ✓ Reusable modules consist of commands, logic, and data.
  - ✓ They should be grouped together in ways that make sense.
  - ✓ Generic modules used throughout the testing system, such as initialization and setup functionality, are generally grouped together in files named to reflect their primarily function, such as "Init" or "setup".
  - ✓ Others that are more application specific, designed to service controls on a customer window, for example, are also grouped together and named similarly.
  - ✓ All the modules that service the customer screen are organized in one file, or library. That way, when the customer screen is modified for any reason, updates to the testing system are located in one place; hence the principle of One Point Maintenance comes into existence.

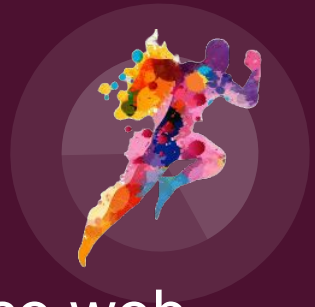
Automated Testing Methodologies

- **Test Cases**
  - ✓ The next step in the methodology is to turn reusable modules into automated test cases.
  - ✓ Here, a well-structured manual test case is converted into scripts, which is nothing but reusable modules.
  - ✓ The goal here is to build the reusable modules in a very methodical manner.
  - ✓ The action-response pairs from the test case are scripted into reusable modules.
  - ✓ These pairs also determine the size or granularity of a reusable module, which generally consist of just one action-response pair.
  - ✓ Automating test cases in this manner allows the testing system to take on a very predictable structure.
  - ✓ One benefit of this predictability is the ability to begin building the automated testing system from the requirements early in the software-testing life cycle.

Automation Testing Life Cycle (ATLC)



What is a Playwright?



Playwright, developed by Microsoft, is an open-source web automation framework that provides a high-level API control and is compactable with many browsers such as Chromium, Firefox, etc. It is often used for web testing, scraping, and other browser-related actions.

Features of Playwright

- Cross-browser. Playwright supports all modern rendering engines including Chromium, WebKit, and Firefox.
- Cross-platform. Test on Windows, Linux, and macOS, locally or on CI, headless or headed.
- Cross-language. Use the Playwright API in TypeScript, JavaScript, Python, .NET and Java.
- Test Mobile Web. Native mobile emulation of Google Chrome for Android and Mobile Safari. The same rendering engine works on your Desktop and in the Cloud.

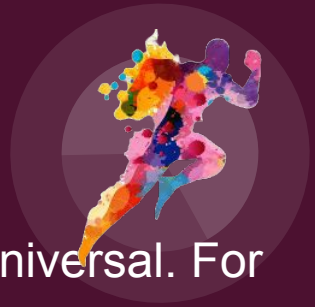
Powerful Tooling

- Auto wait.
- Web-first assertions.
- Tracing
- Test frames, pierce Shadow DOM.
- Codegen. Generate tests by recording your actions. Save them into any language.
- Playwright inspector. Inspect page, generate selectors, step through the test execution, see click points, explore execution logs.
- Trace Viewer. Capture all the information to investigate the test failure. Playwright trace contains test execution screencast, live DOM snapshots, action explorer, test source, and many more.

Advantages of Playwright

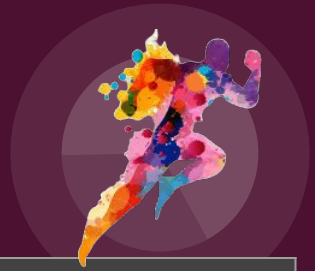
- Cross-browser test support.
- Cross-platform support.
- Cross-language support.
- Headful and Headless modes.
- Mobile support.

Limitations of Playwright



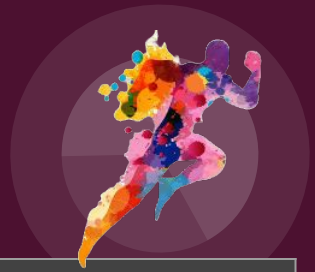
- **Limited Browser Support:** Playwright's browser support is not universal. For some specific browsers, there is a need for an extra workaround.
- **Limited Operating System Support:** Playwright is primarily designed for Windows, Linux, and macOS. It might not support other operating systems.
- **Lower Adoption Rate:** Playwright is not as much adopted as other third-party libraries and extensions like Selenium.
- **Longer Learning Curve:** Its high-level API and advanced features may have a steep learning curve for beginners compared to simpler automation tools.

Playwright vs Selenium



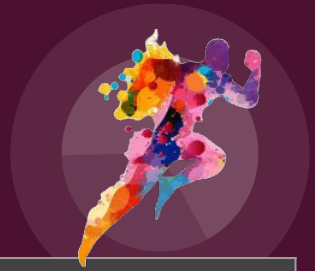
Parameters	Playwright	Selenium
High-level API	Playwright provides a high-level API for easier web browser automation.	Selenium relies on WebDriver API for web browser automation.
Responsive Testing & Device Emulation	It provides responsive testing by enabling the emulation of various devices.	It is often limited in capabilities to provide responsive testing by emulation of devices.
Performance & Stability	Playwright enables strong interaction with browsers and is known for its performance and stability.	The performance and stability of Selenium tests are inconsistent and depend on browser and driver used.

Playwright vs Selenium



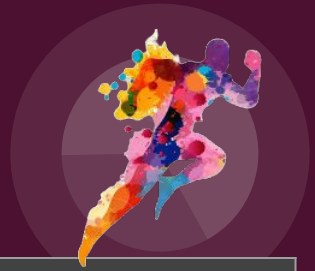
Parameters	Playwright	Selenium
Network Interception	It provides network interception for request/response manipulation.	It has limited support for network interception.
File Handling	It has built-in support for file uploads and downloads. No need for additional workaround.	It doesn’t provide any built-in support for file handling. May requires additional steps.
Mobile Browser Automation	It supports the automation of mobile browsers on Android and iOS.	It has limited support for mobile browser automation.

Playwright vs Selenium



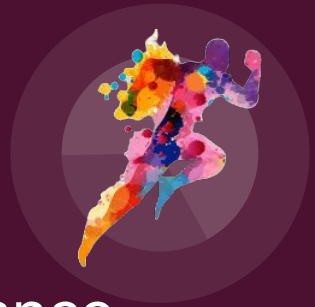
Parameters	Playwright	Selenium
Community & Support	The playwright has a growing community and support from Microsoft.	Selenium is a widely adopted framework and has a large community.
Browser Support	Playwright has extensive support for all kind of browsers. Example: Chrome, webkit, Firefox etc.	Selenium has limited support for all kind of browsers. Examples: Safari, Chrome, Edge, Firefox, etc
Programming language support	Playwright supports programming languages like C#, Python, Java and JavaScript.	Selenium supports programming languages like C#, Python, Ruby, JavaScript and Java.

Playwright vs Selenium



Parameters	Playwright	Selenium
Mobile device support	Playwright provides mobile device support for Android and IOS.	Selenium provides mobile device support with the help of Appium.
Operating System support	Playwright supports Operating Systems like Windows, MacOS, Linux.	Playwright supports Operating Systems like Windows, MacOS, Linux.
Compactible Architecture	Playwright supports 32-bit and 64-bit architecture.	Selenium supports 32-bit and 64-bit architecture.

Conclusion



Playwright’s modern features and excellent performance make it a strong contender, but Selenium’s long-standing reputation and adaptability still hold a valuable place in the world of web automation. The choice between these depends on organization’s structure, specific needs, preferences and tasks which need be automated.

Queries?

Thank You!