

Test Automation

BY:

P.M.A.K.WIJERATNE

Introduction to Test Automation

- Test automation refers to the use of software tools and scripts to perform tests on software applications automatically, rather than relying on manual testing methods.
- It involves writing scripts that can be executed repeatedly, allowing testers to perform regression testing, functional testing, and other types of testing efficiently.

Why it is Important to Have Test Automation

- **Efficiency:**

Automated tests can be executed much faster than manual tests, allowing for quicker feedback on the software's quality.

- **Repeatability:**

Automated tests can be run multiple times with the same precision, ensuring consistent results.

- **Coverage:**

Automation allows for a more extensive test coverage as tests can be run on multiple configurations, environments, and data sets.

Why it is Important to Have Test Automation

- **Cost-Effective:**

In the long run, automation reduces the cost of testing by decreasing the amount of manual labor required.

- **Early Bug Detection:**

Automated tests can run as part of continuous integration, catching bugs early in the development cycle.

- **Regression Testing:**

Automated tests are ideal for regression testing, ensuring that new code changes do not adversely affect existing functionality.

How to Select Test Scenarios for Automation

Here are the key considerations involved in selecting test scenarios for automation:

- Scenarios that **need to be run frequently** or for every build/release cycle are ideal candidates for automation.
- Test cases that cover core functionalities and have high impact on business processes should be automated to ensure reliability.
- Automate scenarios where the requirements are not expected to change frequently. Frequent changes can lead to high maintenance costs for automated tests.

How to Select Test Scenarios for Automation

- Tests that take a lot of time when performed manually, such as regression tests, should be automated to save time and resources.
- Scenarios that require multiple sets of data to be tested with the same steps should be automated to handle different data sets efficiently.
- Tests involving complex calculations or multiple combinations of inputs are suitable for automation to ensure accuracy and consistency.

Steps to Select Test Scenarios for Automation

- Review your test suite to identify test cases that align with the key considerations mentioned above.
- Prioritize the identified test cases based on factors such as business impact, frequency of execution, and complexity. High-priority cases should be automated first.
- Assess the feasibility of automating the identified test cases. Consider the technical complexity, availability of test data, and the tools required.
- Develop a strategy that outlines which test cases will be automated, the tools to be used, the timeline, and the team responsibilities.

Steps to Select Test Scenarios for Automation

- Choose automation tools that are best suited for your application, technology stack, and the type of tests you need to automate. Ensure the tools support the necessary integrations and provide robust reporting features.
- Begin developing automation scripts for the prioritized and feasible test cases. Follow best practices for scripting to ensure maintainability and scalability.
- Regularly review the automated test cases and refine them based on feedback and changing requirements. Ensure that the automated tests remain relevant and effective.

Examples of Scenarios Suitable for Automation:

- Automate login functionalities to ensure consistent user authentication across different scenarios.
- Automate form submissions with various data inputs to verify validation and data processing.
- Automate API testing to verify the responses, data formats, and error handling.
- Automate navigation through the application's UI to ensure links and buttons work correctly.
- Automate the entire regression test suite to ensure new code changes do not break existing functionality.
- Automate tests to measure the application's performance under different load conditions.

When is the Perfect Time to Start Test Automation in a Project

- **Early in the Development Cycle:** Starting automation early helps in integrating it into the development process and ensures that tests are created alongside code.
- **After Initial Stability:** When the application has reached a certain level of stability where the core functionalities are unlikely to change drastically.
- **After Manual Test Cases are Defined:** When manual test cases are well-defined, they can be translated into automated test scripts.
- **When There is Enough Test Data:** Ensure there is sufficient test data available for effective test automation.

How to Select a Test Automation Strategy and Appropriate Tools

1. Define Goals:

Clearly define what you want to achieve with test automation, such as reducing testing time, increasing coverage, or improving accuracy.

2. Evaluate Application:

Assess the application under test to understand its requirements, complexities, and the types of testing needed.

How to Select a Test Automation Strategy and Appropriate Tools

3. Tool Selection Criteria:

- Ensure the tool supports the technology stack of the application.
- The tool should be user-friendly and easy to set up.
- The tool should integrate with existing CI/CD pipelines and other tools used in the development process.
- Choose tools with good documentation, support, and an active user community.

How to Select a Test Automation Strategy and Appropriate Tools

4. Popular Automation Tools:

- Selenium: For web application testing.
- Appium: For mobile application testing.
- JUnit/TestNG: For Java applications.
- Cucumber: For behavior-driven development (BDD).
- Jenkins: For continuous integration and continuous deployment (CI/CD).

How to Select a Test Automation Strategy and Appropriate Tools

5. Create a Framework:

Develop a robust test automation framework that includes guidelines, best practices, and reusable components.

6. Start Small:

Begin with a small set of test cases to automate and gradually expand as the process becomes more refined.

7. Continuous Monitoring and Improvement:

Regularly review and update the automated tests and the framework to ensure they remain effective and aligned with the project's needs.

What types of automation that can be performed

- Various types of automation can be performed to ensure that different aspects of an application are functioning as expected.
- Each type of automation has its own focus and purpose, and selecting the right type of automation is crucial for effective testing.

What types of automation that can be performed

1. Unit Testing
2. API Testing
3. UI Testing (Graphical User Interface)
4. Functional Testing
5. Regression Testing

What types of automation that can be performed

1. Unit Testing

- **Purpose:** To verify that individual units or components of the software function correctly.
- **Scope:** Focuses on the smallest parts of an application, such as functions or methods.
- **Tools:** JUnit, NUnit, TestNG, xUnit, etc.
- **Example:** Testing a function that calculates the sum of two numbers to ensure it returns the correct result.

What types of automation that can be performed

2. API Testing

- **Purpose:** To validate the functionality, performance, security, and reliability of APIs.
- **Scope:** Focuses on testing the endpoints, requests, responses, and integrations of an API.
- **Tools:** Postman, SoapUI, RestAssured, JMeter, etc.
- **Example:** Sending a POST request to an API endpoint and verifying that it returns the correct status code and response data.

What types of automation that can be performed

3. UI Testing (Graphical User Interface)

- **Purpose:** To ensure that the user interface behaves as expected and that users can interact with the application without issues.
- **Scope:** Focuses on the visual elements and user interactions, such as buttons, forms, and navigation.
- **Tools:** Selenium, Appium, TestComplete, Katalon Studio, etc.
- **Example:** Automating the login process in a web application by entering a username and password and verifying that the user is redirected to the correct page.
- .

What types of automation that can be performed

4. Functional Testing

- **Purpose:** To verify that the application functions according to the specified requirements.
- **Scope:** Covers end-to-end functionality, often involving multiple components or systems.
- **Tools:** QTP/UFT, Selenium, TestComplete, etc.
- **Example:** Testing an e-commerce checkout process from adding items to the cart to completing the purchase.

What types of automation that can be performed

5. Regression Testing

- **Purpose:** To ensure that recent changes to the codebase do not negatively affect existing functionality.
- **Scope:** Involves re-running previously executed test cases on a new code version.
- **Tools:** Selenium, Ranorex, TestComplete, etc.
- **Example:** Re-running tests on a new software build to confirm that recent bug fixes have not introduced new issues.

What are the challenges of test automation?

- High initial setup costs.
- Maintenance of test scripts due to frequent changes in the application.
- Choosing the right tool and framework.
- Handling dynamic elements in the UI.
- Ensuring test data availability and management.

What are the best practices for writing automated test scripts?

- Keep tests independent and repeatable.
- Use descriptive names for test cases.
- Maintain a good balance between UI and API tests.
- Implement proper exception handling and logging.
- Regularly review and refactor test scripts.

Thank You!

