How to understand Automation Frameworks?

Rupendra Ragala

Linear Automation Framework (Record-and-Playback Framework)

Explanation:

This is the simplest framework where test cases are created sequentially by recording user actions and replaying them. No modularity or scripting knowledge is needed.

Use Cases:

- Quick testing for small projects with minimal test cases.
- Ad-hoc testing for GUI-based applications.

Tools:

- Selenium IDE
- UFT (Unified Functional Testing)

Modular Testing Framework

Explanation:

The application under test is divided into separate modules, and test scripts are written for each module. This improves maintainability and reusability.

Use Cases:

- Testing applications with multiple independent components, like an e-commerce platform.
- Scenarios where specific functionalities need isolated testing.

Tools:

- Selenium WebDriver
- TestNG/JUnit

Data-Driven Testing Framework

Explanation:

This framework separates the test script logic from the test data. It allows testers to execute the same set of actions with different sets of data inputs.

Use Cases:

- Form validation with multiple data sets.
- Applications requiring extensive input/output combinations.

Tools:

- Apache POI (for Excel handling in Java)
- Selenium WebDriver
- Robot Framework

Keyword-Driven Testing Framework

Explanation:

This approach uses keywords to represent actions (e.g., "click", "enter text") and decouples test scripts from the testing logic. Keywords are mapped to functions in the automation tool.

Use Cases:

- Testing complex workflows with non-technical teams involved.
- Applications with repetitive actions across modules.

Tools:

- Selenium
- Katalon Studio
- UFT

Hybrid Testing Framework

Explanation:

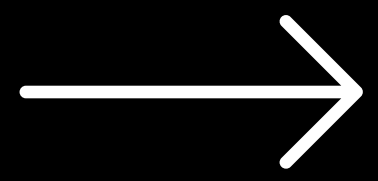
A combination of two or more frameworks (e.g., data-driven and keyword-driven). This framework leverages the strengths of each integrated framework.

Use Cases:

- Applications requiring extensive testing with diverse requirements.
- Projects with multiple teams, including non-technical testers.

Tools:

- Selenium with TestNG
- Katalon Studio
- Appium



Behavior-Driven Development (BDD) Framework

Explanation:

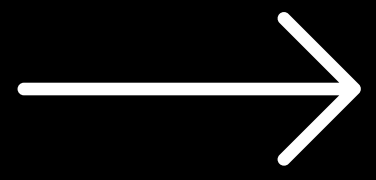
Focused on collaboration between developers, testers, and business analysts. Test cases are written in natural language using Gherkin syntax.

Use Cases:

- Agile projects with frequent stakeholder involvement.
- Complex systems requiring clear communication between teams.

Tools:

- Cucumber
- SpecFlow
- JBehave



Test-Driven Development (TDD) Framework

Explanation:

This framework focuses on writing tests before the actual application code. Tests guide development and ensure robust code.

Use Cases:

- Development-driven testing in Agile or DevOps environments.
- Ensuring minimal bugs in the development phase.

Tools:

- JUnit
- NUnit
- TestNG

Library Architecture Testing Framework

Explanation:

Extends the modular framework by creating common functions and libraries for reusable code. Tests are written using these libraries, enhancing maintainability.

Use Cases:

- Projects with complex reusable components.
- Systems with high interdependencies across modules.

Tools:

- Selenium WebDriver
- Appium
- RestAssured

Scriptless Testing Framework

Explanation:

Focuses on reducing code by allowing non-programmers to create and execute tests using visual workflows or drag-and-drop functionalities.

Use Cases:

- Teams with limited programming expertise.
- Rapid test creation for large-scale applications.

Tools:

- Tricentis Tosca
- TestComplete
- Leapwork

CI/CD Integrated Automation Framework

Explanation:

Combines test automation with continuous integration/continuous delivery pipelines. Automates the build, testing, and deployment process.

Use Cases:

- DevOps projects with frequent releases.
- Automated regression testing in Agile workflows.

Tools:

- Jenkins
- GitLab CI/CD
- Bamboo
- Selenium/Grid

Follow for

more

Information

Thank You