Automation Frameworks

Features of Automation Framework

- Maximum code reusability
- Minimal maintenance effort for the scripts
- Provision for non-technical business users to interact with the framework, in terms of easily driving the test execution, manipulating test data, etc.
- Ease of script development – must provide appropriate support libraries to handle common and repetitive framework tasks
- Test data externalization – to avoid embedding test data within the scripts
- Robust error handling mechanism – to capture any unexpected errors during the test run, without affecting the overall execution
- Detailed reporting mechanism – to help in easy analysis of test execution results
- Integration with other tools as required – for example, Sauce labs, Jenkins tools for Selenium

Types of automation frameworks

- Modular/Function Driven
- Data-driven
- Keyword-driven
- POM (Selenium)
- Test NG/Junit (Selenium)
- Hybrid

Modular/Function Driven

- Involves **identifying reusable test steps** and encapsulating the same into functions in external libraries
- The functions can be **called from multiple scripts** as required
- Example/Exercise:
  - How will you automate the following manual test cases using the modular approach? (A, B, C, etc. refer to the test steps)

TC1	TC2
A	C
B	D
C	E
D	F
E	G

Modular/Function Driven

Steps involved:

1. Identify **repeated/common test steps** across the test cases.
2. Create **reusable functions** for the common steps identified.
3. Store the reusable functions into a **centralized library** for future reference.
4. These libraries are typically called **functional libraries**, because they are specific to the functionality of the AUT.
5. **Create automated test scripts** for each of the manual test cases.
6. **Consume the reusable functions** as required within the test cases.

Modular/Function Driven

Functional libraries:

Functions	Function Steps
CDE()	C
	D
	E

Test scripts:

TS1	TS2
A	CDE()
B	F
CDE()	G

Modular : Pros and Cons

Pros:

- Promotes reusability of code
- Reduces maintenance effort by minimizing duplication of rework effort

Cons:

- Test data is hard coded within the scripts/functions
- Does not provide features for non technical business users to interact with the framework

Data-driven : Overview

Involves **externalizing the test data** from the test automation tool (usually into an Excel sheet).
Example:

E	F	G	H	I	J
Browser	BrowserVersion	Platform	URL	UserName	Password
Firefox	11	Win 7	https://Testenviroment.com	XXXX1	YYYY1
InternetExplorer	32	Win XP	https://Testenviroment.com	XXXX2	YYYY2
InternetExplorer	9	Win 8	https://Testenviroment.com	XXXX3	YYYY3
Chrome			https://Testenviroment.com	XXXX4	YYYY4

Data-driven approach: Pros and Cons

Pros:

- Avoids hard-coding of test data in the scripts
- Non-technical users can easily change the test data using the Excel sheets, without modifying the scripts
- The same script can easily be executed with multiple rows of test data

Cons:

- The end user should be careful to specify the data inputs in the format expected by the framework, otherwise, the results can be unexpected

Keyword Driven

In this approach, a keyword represents **a single basic user action**, like clicking on a button, typing text in a textbox, etc.
These keywords are categorized as keywords, and are also known as **action keywords**.

E.g.: ButtonClick, TypeText, etc.

Example:

Object	Event	data
txtUsername	sendkeys	User1
txtPassword	sendkeys	Pass1
btnSumbit	Click	

TC #	TC Name	Execute ?
1	Login	Yes
2	Inbox	No
3	Submit	Yes

Keyword Driven: Pros and Cons

Pros:

- This approach is largely script-free, hence there is less dependency on automation experts.
- Tests can be automated without any programming knowledge and are highly readable.

Cons:

- Initial efforts and time will be high for developing this framework

Hybrid Framework: Overview

- Most automation frameworks today are Hybrid frameworks.

- Combining the best practices from more than one approach.

- The aim is to avoid the disadvantages of individual approaches

Hybrid Framework

- Modular + Data driven
- Keyword driven + Data driven
- **POM+ Data driven + Test NG + Modular +Maven**