Test Automation Frameworks

LTTS GLOBAL ENGINEERING ACADEMY

L&T Technology Services

Sneha Anand
99003525

Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
	12/03/2021	Sneha Anand			

Contents

Test Automation:

Test automation is the practice of running tests automatically, managing test data, and utilizing results to improve software quality. It is a software testing technique that performs using special automated testing software tools to execute a test case suite.

Benefits of Test automation:

1. Increased testing efficiency,
2. Increased testing effectiveness,
3. Faster time to market.

Feasibility of Test Automation:

While doing a feasibility Study for test automation the factors to be considered and percentage of importance are as shown below:

- Technical analysis (20)
- Complexity(15)
- Application stability(15)
- Test Data(15)
- Application size(10)
- Reusability of Automation scripts(10)
- Execution across Environments(15)

Test Automation for Mobile Application:

Mobile automation, as the name suggests, refers to 'automation' that is done on mobile devices. Automation is the process whereby one automates testing of an application - in this case a mobile application - which can be a WAP site or an app. This can be done by using tools and helps in reducing the testing time cycle.

Automated software testing is a cost-effective solution for mobile applications since they need to be tested on various handsets and produce high-quality, robust and reliable software with the ever-growing complexity of technology and under massive competitive pressure. Test automation offers a significant value-add by enabling testing to be done in parallel.

Different Ways to Automate Mobile Testing:

HANDWRITTEN TEST SCRIPTS

Typically, this is the best choice when you know what you're doing and when you have programming-capable people doing the test scripts. Plenty of options are available for test automation frameworks, tools and integration — both commercial and open-source offerings

RECORD-PLAYBACK APPROACH

In this method, the tester needs to write any code instead of just to record the user's actions. However, the tester will need to do coding to fix things that go wrong or fine-tune the automation behaviour. This method is easier than writing a complete test script from scratch because you already have the complete code. This approach is less error-prone because nothing needs to be written in code, but it is typically more limited in functionality

AUTOMATIC TEST EXERCISERS

Automatic test exercisers provide a great way to smoke-test applications. No specific tests are needed; rather, the focus is on testing user-interface logic. Automatic test exercisers yield the least exact results but provide quick feedback on each iteration of an app.

Automation Approaches:

1. Two common test automation approaches:

- User-agent based testing
- Device-based testing

2. User-agent based testing:

User-agent based testing utilizes the user-agent identifier string sent by the browser to spoof a browser on a device. This approach can be used for executing mobile web applications.

3. Device-based testing:

Device-based testing on the other hand involves running the application under test directly on the device. This approach can be used for all types of mobile applications.

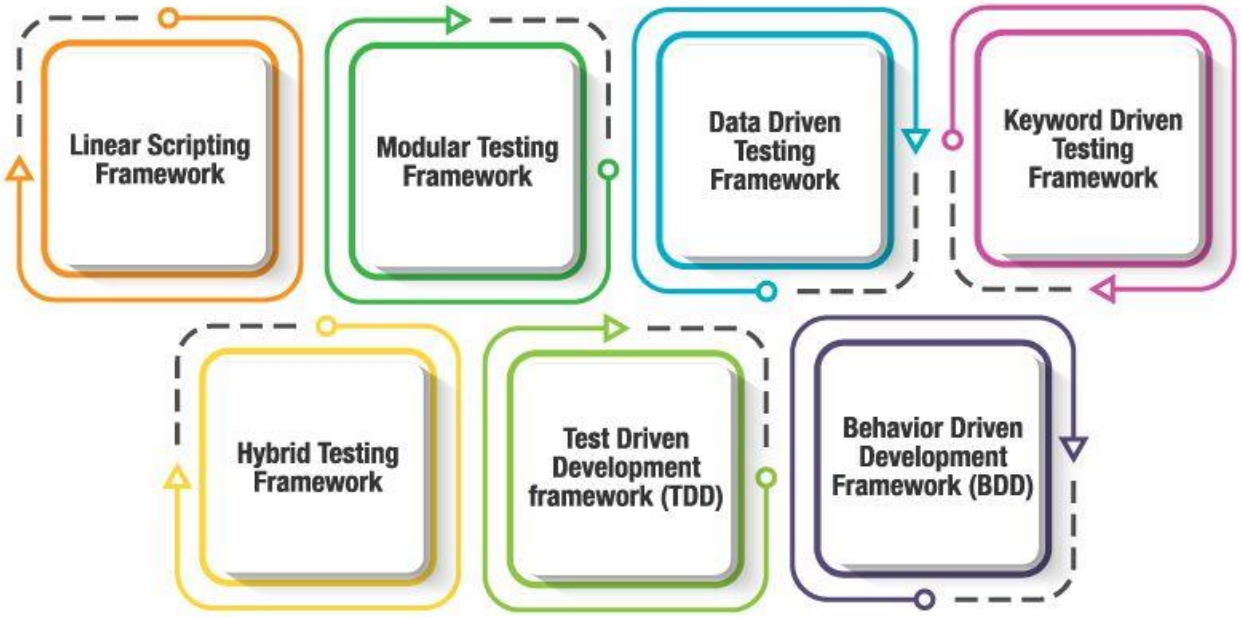
Automation Framework:

A Test Automation Framework is a set of guidelines like coding standards, test-data handling, object repository treatment etc. Which when followed during automation scripting produces beneficial outcomes like increased code re-usage, higher portability, reduced script maintenance cost etc. It integrates various functions like libraries, generic functions, test data, properties of test environment, and various reusable modules.

Automation Frameworks:

- Enhances efficiency during the design and development of automated test scripts by enabling the reuse of components or code
- Provides a structured development methodology to ensure uniformity of design across multiple test scripts to reduce dependency on individual test-cases
- Enables reliable issue and bug detection and delivers proper root-cause analysis with minimum human intervention for the system under test
- Reduces dependence on teams by automatically selecting the test to execute according to test scenarios
- Refines dynamically test scope according to changes in the test strategy or conditions of the system under test
- Improves utilization of various resources and enables maximum returns on efforts

Types of Test Automation Framework:



Linear Automation Framework:

It is the simplest of all Testing Automation Frameworks and also known as **"Record & Playback"**. In this Automation Testing Framework, Tester manually records each step (Navigation and User Inputs), Inserts Checkpoints (Validation Steps) in the first round. He then, plays back the recorded script in the subsequent rounds. In this type, the creation, and execution of test scripts are done individually for each test case individually. Testers capture each test step such as browsing, navigation, user inputs, enforcing checkpoints. Testers then play the scripts to carry out the tests.

Modular Driven Framework:

In the modular testing framework, testers create test scripts module wise by breaking down the complete application under test into smaller, independent tests.
Abstraction is the concept on which this framework is built. Based on the modules, independent test scripts are developed to test the software. Specifically, an abstraction layer is built for the components to be hidden from the application under tests
In simple words, testers divide the application into multiple modules and create test scripts individually. These individual test scripts can be combined to make larger test scripts by using a master script to achieve the required scenarios. This master script is used to invoke the individual modules to run end to end test scenarios.
The main reason for using this framework is to build an abstraction layer to safeguard the master module from any changes made in individual tests.In this framework, testers write function libraries to use it whenever required. This is AKA modularity framework or module-based framework.

Data- Driven Framework:

The data-driven test automation framework is focused on separating the test scripts logic and the test data from each other.
It allows us to create test automation scripts by passing different sets of test data. The test data set is kept in the external files or resources such as MS Excel Sheets, MS Access Tables, SQL Database, XML files, etc.,
The test scripts connect to the external resources to get the test data.
By using this framework we could easily make the test scripts work properly for different sets of test data.
This framework significantly reduces the number of test scripts compared to the module-based framework.
This framework gives more test coverage with reusable tests and flexibility in the execution of tests only when required and by changing only the input test data.
It is reliable in terms of no impact on tests by changing the test data but it has its own drawbacks such as testers who work on this framework needs to have the hands-on programming knowledge to develop test scripts

Keyword Driven Framework:

It is also known as table-driven testing or action word based testing.
In Keyword-driven testing, we use a table format to define keywords or action words for each function or method that we would execute.It performs automation test scripts based on the keywords specified in the excel sheet.
By using this Framework, testers can work with keywords to develop any test automation script, testers with less programming knowledge would also be able to work on the test scripts.
The logic to read keywords and call the required action mentioned in the external excel sheet is placed in the main class. Keyword-driven testing is similar to data-driven testing. Even though to work on this framework doesn't require much programming skills but the initial setup (implement the framework) requires more expertise.

Hybrid Driven Framework:

Hybrid Test automation framework is the combination of two or more frameworks mentioned above. It attempts to leverage the strengths and benefits of other frameworks for the particular test environment it manages. Most of the teams are building this hybrid driven framework in the current market.

Behavior Driven Development Framework:

The purpose of this Behaviour Driven Development framework is to create a platform that allows everyone (such as Business Analysts, Developers, Testers, etc.,) to participate actively. It requires increased collaboration between Development and Test Teams. It doesn't require the users to be acquainted with a programming language. We use non-technical, natural language to create test specifications. Some of the tools available in the market for Behaviour Driven Development is JBehave, Cucumber, etc.