



TestingWhiz
Code Less, Test More

WHITE PAPER

TEST AUTOMATION FRAMEWORK DESIGN

A Step by Step Guide

Contents

Executive Summary	1
What is an Automation Testing Framework?	1
What are the Types of Test Automation Frameworks?	1
1. Record & Playback	1
2. Data-Driven	2
3. Keyword-Driven	2
4. UI Page Maps	3
5. The Hybrid	3
Why Utilize a Test Automation Framework?	4
What are the Benefits of Utilizing a Test Automation Framework?	4
Things to Consider While Designing a Test Automation Framework	6
1. Independent test scripts from input values	6
2. Restrict a few libraries	6
3. Design the coding standards	6
4. Offer high extensibility	7
5. Reduces framework maintenance	7
6. Framework version maintenance	7
How to Design an Automation Testing Framework?	7
Conclusion	8

Executive Summary

The software testing industry is moving towards automation and it is all because of the benefits that it offers. Automation testing not only reduces the manual efforts, but also speeds up the releases. Also, we have various automation testing tools available in the market. But, what if the application under test requires complex automation testing architecture? Here comes the option to go for the automated testing framework.

This white paper takes you to the complete step by step guide to what is an automation testing framework, its types, why it is required to design and how to design it in the most appropriate way. This will ensure you are able to test the most complex application with ease and speed without compromising on the quality and the outcome of the application.

What is an Automation Testing Framework?

The automation testing framework is an environment to be designed with a set of rules, guidelines, processes, practices, and standards that allow you to execute the well-written automated test scripts. In simple language, it is a system where your test scripts will be automated.

With the test automation framework, the testers can easily create an automated testing environment and define the common outputs of the applications under test. Execution and reporting of the tests also become the part of the framework.

What are the Types of Test Automation Frameworks?

While talking about the test automation frameworks, it is important to learn about its types. Check out the below-mentioned test automation framework types:

1. Record & Playback

It offers an environment to record each test step and verify its execution through the playback functionality.

Pros

- ❖ Lesser efforts
- ❖ No technical expertise required
- ❖ Easy & quick test script development

Cons

- ❖ Hard-coded test data
- ❖ No reusable test scripts
- ❖ Synchronization issues in UI testing

2. Data-Driven

Here, data is stored in external files and it is extracted from the external sources to test the data. Thus, testers do not require writing the code for test data into test scripts repeatedly as testers can execute the same test cases with multiple data sets.

Pros

- ❖ Better test coverage
- ❖ Easy test script editing
- ❖ No hard-coded test data

Cons

- ❖ No real-time business functionality testing
- ❖ Complex specification of association between specific test data & test scripts
- ❖ Coding skills required

3. Keyword-Driven

Data tables and keywords are developed in order to fetch the test data. Here, when the automated tests are executed, the test data is fetched which is attached to a keyword. That keyword calls the test script attached to it for execution.

Pros

- ❖ Test execution can be ordered & executed as per the needs
- ❖ Easy code & keywords reusability
- ❖ Test scripts are independent of the tools/apps

Cons

- ❖ Requires excellent programming skills
- ❖ Updating test scripts for UI changes is a big challenge

4. UI Page Maps

Test scripts are created for the classes on the specific UI page. Test data has the commands to identify the UI objects and to execute the test scripts attached to it.

Pros

- ❖ Better maintenance & reliable tests
- ❖ Reusable UI test scripts
- ❖ Easy test case development

Cons

- ❖ For new UI objects, new test needs to be written
- ❖ For UI object class changes, existing tests need to be updated

5. The Hybrid

A combination of 2 or more automation framework as per the requirements.

Pros

- ❖ Good option for complex apps
- ❖ Offers flexibility between different frameworks

Cons

- ❖ Complexity in initial setup of tests
- ❖ Needs deep test planning & execution

Why Utilize a Test Automation Framework?

These are the implied benefits of having a test automation framework. Certainly, there are many more benefits attached to test automation framework.

What are the Benefits of Utilizing a Test Automation Framework?

Let's check out what invokes the need to design a test automation framework:

To create a new test plan:

Each time a new test plan and strategy is created for the application, a test automation framework needs to be designed.

To accommodate testing functionalities:

For similar kind of applications, same testing functionalities need to be planned and prepared again and again as there is no set of predefined tests available. This requires a framework to be created to accommodate testing functionalities.

To enable duplicate test cases creation:

Having a test framework in place helps create duplicate automated test cases more frequently and easily.

To enable a better understanding of test strategies:

Test automation framework helps testers in understanding the test strategies, the environment and execute tests accordingly.

To unify testing strategies:

Designing a test automation framework helps in unifying the testing strategies, as a result, it reduces chances of errors and productivity issues among testers.

To maintain test cases easily:

With a test automation framework, it becomes easy to maintain test cases and scripts.

Apart from the ones we discussed, here are advanced benefits of utilizing a test automation framework.

1. Leverage different testing tools:

With the test automation framework, users can leverage multiple tools as per their needs. Many organizations have been utilizing Selenium, TestingWhiz, Test runner, BrowserStack and many more which support one or more than one testing framework.

2. Redundant test commands of the tools:

Having a test automation framework in place can help get rid of few low level and unnecessary commands and methods of the tool to make it more powerful for meeting complex test requirements.

3. Independent of the application under test:

Irrespective of the type of application under test, an automation testing framework works perfectly in any test environment.

4. Involvement of the various testing solutions:

As test automation frameworks are independent of the tools, using tools like the ones mentioned above helps in performing multiple types of testing like functional, regression, UI, unit, integration, cross-browser etc.

5. Common testing API:

The basics of coding and databases are quite common. Like any other API, it has easy exception & error handling and code & data maintenance functionalities.

6. Multiple app version testing:

Test automation framework need not require being in each machine to test various versions of the application. Once, it is deployed on a server machine, each version can be tested on a single machine.

7. Test automation framework multithreading:

With the test automation framework, you can execute all tests simultaneously on all the machines across different test environments and app versions.

8. Involved with Continuous Integration:

Test automation framework also supports Continuous Integration methodology to meet continuous testing and delivery cycle.

9. Avoid identical test cases:

A well-designed automation framework also helps avoid automating identical test cases which keep the test cases and resources more optimized.

10. Overall improved efficiency:

With the help of appropriately designed test automation framework, the testers can easily create their test suites and can execute them efficiently.

Things to Consider While Designing a Test Automation Framework

When you plan to design an automated testing framework, you can prepare yourself with the following points:

1. Independent test scripts from input values

While creating test scripts for automation, every time the data entered to perform tests, would not be similar. It may vary based on the logic criteria of the application. Hence, the data should be independently maintained in some data store like databases, MS Excel, XML, etc.

Benefit

This will eliminate the chances of making any change in the existing test scripts, saving a lot of time and efforts.

2. Restrict a few libraries

A library in respect to a software application is created so that multiple programs with little to no connection to each other, can utilize it. The examples can be databases, general application functionalities, etc.

Now, it becomes very important for the testers to have restrictions in using and making changes to the pre-programmed libraries.

Benefit

Pre-programmed libraries have to be invoked at various stages of testing and if the testers are trying to tweak the existing libraries, can break all the code and logic of the test scripts.

3. Design the coding standards

Before designing a test automation framework, the specific coding standards have to be defined in advance in order to maintain the quality and uniformity of the code between testers and developers.

Benefit

Testers and developers can easily understand the code, and can connect to the way it has been developed. Also, they don't need to apply coding individually.

4. Offer high extensibility

Design an automated testing framework in a way that it accepts all the new changes to the existing application and verifies each of the new feature and enhancement.

Benefit

Extensibility in test automation framework will help the testers to qualify the end product in the most appropriate manner and hence, fulfilling the customer expectations.

5. Reduces framework maintenance

The test automation framework should be designed in a way that it becomes an ideal for the application that comes your way and requires very less maintenance.

Benefit

The automation framework that comes with less maintenance requires less time and efforts to be invested in making any necessary changes.

6. Framework version maintenance

The test scripts among the framework and the multiple versions of the framework need to be stored and maintained through a version control tool or a local repository.

Benefit

When the test scripts and framework versions are stored in a different place, it helps the testing team to monitor and handle the changes that are made to the actual code.

How to Design an Automation Testing Framework?

Automation testing framework design is something that depends on each testing team's requirements and ease. There will be some testers who would develop a test automation framework using Selenium or similar kind of test automation tools. And, some testers would build the whole automation testing from the scratch. In fact, the majority of the organizations look for the already build automation testing framework from some niche testing solutions provider.

Conclusion

Automation testing tools work extremely well for smaller testing projects. But, when the application logic is complex, then the tools need customization. Here, the appropriate test automation framework becomes imperative to be designed. Efficiently designed test automation framework helps in overcoming the manual testing hurdles and improves the overall ROI on the test automation.

References

<https://www.cs.colorado.edu/~kena/classes/5828/s12/presentation-materials/ghanakotagayatri.pdf>

<https://www.slideshare.net/saucelabs/test-automation-framework-designs>

https://www.cs.colorado.edu/~kena/classes/5828/s10/presentations/automation_test_frameworks.pdf

<http://www.testing-whiz.com/blog/how-to-choose-automation-testing>

<http://www.evoketechnologies.com/blog/test-automation-framework-design/>

About TestingWhiz

TestingWhiz is a Codeless Test Automation Tool for Web, Database, Web Services/API, and Mobile testing. It is based on a proprietary FAST® (Flexible Automation Scripting Technology) Automation Engine, and comes with 300+ readily available test commands that eliminate long hours of scripting, allowing users to code less and test more.

It helps all small, midsized and large-scale software enterprises as well as individual test engineers and QA analysts to move to the next level of hassle free test automation.

[Download Trial](#) | [Request A Demo](#) | [Pricing](#) | [Contact Us](#)



+1-646-915-0021



info@testing-whiz.com



www.testing-whiz.com

**TestingWhiz LLC, Mack-Cali Centre III, 140 E, Ridgewood Avenue,
Suite 415 ST, Paramus, NJ 07652**

Connect with Us



Disclaimer

The information contained herein has been obtained from our work done in the relevant area with our expertise. TestingWhiz disclaims all warranties as to the accuracy, completeness or adequacy of such information. TestingWhiz shall have no liability for errors, omission or inadequacies in the information contained herein, or for interpretations thereof. The material in this publication is copyrighted. No part of this material can be reproduced either on paper or electronic media without permission in writing from TestingWhiz. Request for permission to reproduce any part of the report may be sent to TestingWhiz.