



TEST AUTOMATION FRAMEWORKS

INTRODUCTION ON BENEFITS, FRAMEWORK TYPES
AND AUTOMATED TESTING TOOLS

ITSE 1392



MAY 15, 2021
VY HOANG

Table of Contents

I. INTRODUCTION	2
Definition of test automation framework	2
Why need to follow framework guidelines?.....	2
Benefits of test automation framework	2
II. COMMON TYPES OF FRAMEWORKS	3
Linear Automation Framework	3
Modular Based Testing Framework	4
Data-Driven Framework.....	4
Keyword-Driven Framework	4
Hybrid Testing Framework	5
III. STEPS TO EVALUATE AND CHOOSE A FRAMEWORK	5
IV. COMMON LANGUAGES FOR TEST AUTOMATION FRAMEWORK	6
Python.....	7
Java	7
JavaScript.....	8
C#.....	8
PHP	8
V. COMPARISON ON SOME POPULAR AUTOMATED TESTING TOOLS	8
VI. CONCLUSION	11
BIBLIOGRAPHY	12

I. INTRODUCTION

Definition of test automation framework

A test automation framework is a set of guidelines used for creating, designing, implementing and maintaining test cases. It comprises a combination of the interlinked components that are designed to support the software testing lifecycle more efficiently.

Why need to follow framework guidelines?

Framework guidelines may include coding standards, test data handling, object repository treatment, utility tools... These guidelines are not the rules, so they are not mandatory to follow.

However, the main goals of using testing automation framework are to maximize the test efficiency while to minimize the cost with the beneficial outcomes such as the increased code re-usage, the higher portability, the reduced script maintenance cost. Therefore, following the guidelines helps keep the framework organized in a standard format and gain the desired results.

Benefits of test automation framework

Selecting the right test automation framework for the product development helps provide the following benefits.

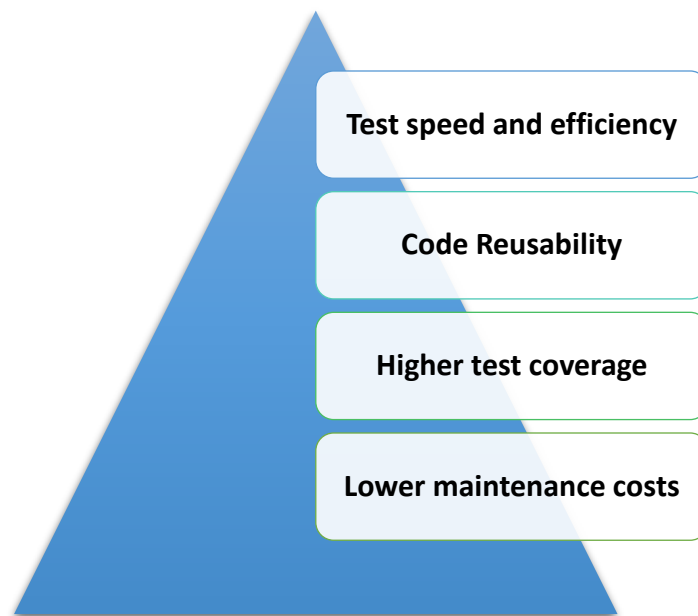


FIGURE 1. BENEFITS OF TEST AUTOMATION FRAMEWORK

- **Test speed and efficiency improvement:** Faster validation for testing phases, useful to detect problems or bugs early during the development and enhance the team's efficiency.
- **Reusability:** Scripts, test suite once defined can be reused for other projects or user cases.
- **Higher test coverage, test case repetition reduced:** Increase the new tests covering for new features or the complex test cases and avoid the tedious testing process that are limited in the manual testing approach.
- **Lower maintenance costs, lower risks:** Once a proper test automation framework is set up, it can save a considerable amount of expenses on the testing resources overall the project, assist creating a stable test automation environment, and improve the accuracy with the lowest human error rate.

II. COMMON TYPES OF FRAMEWORKS

Every framework has its own structure, techniques, advantages, and disadvantages. Below are the five common types of test automation frameworks. (Introducing frameworks, 2013)

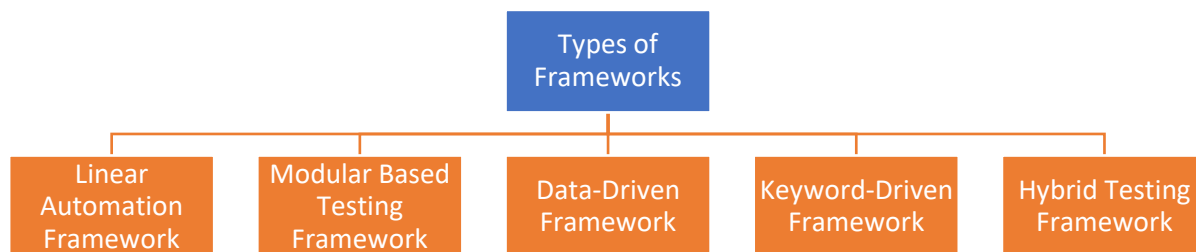


FIGURE 2. TYPES OF FRAMEWORKS

Linear Automation Framework

This framework is also referred to a record-and-playback framework. It is suitable for testing simple applications. In this testing process, testers do not need to write code as the testing steps are written in a sequential order with navigation, user input or checkpoints, and then it plays the script back automatically to conduct the test.

Advantages: This is the simplest framework. Expertise in test automation is not needed as it does not require the custom code written.

Disadvantages: The test script contains hard-coded data which does not allow reusability, so the maintenance is really a big problem.

Modular Based Testing Framework

The modular based testing frameworks are the test library architecture framework or also known as the structure scripting/ functional decomposition. The modular framework allows to break down the whole application into individual modules or functionality. Each module has its own test script level and then they are combined to build in the larger set of tests.

Advantages: As test scripts are written individually for each module, it is convenient for debugging, the script maintenance and scalability.

Disadvantages: It requires the technical knowledge and additional time in analyzing the test cases as well as identifying the reusable flows. Hard-coded data is still needed to use this framework.

Data-Driven Framework

This framework uses the test data from the separated files externally with the script files which can be passed as parameters and loaded into the variables inside the test scripts. The test data files can be Excel spreadsheets, Text files, CSV files, SQL tables.

Advantages: Multiple data sets can be executed with the tests and various test scenarios can be performed by altering the test data into the external data file. Also, the hard-coded data can be avoided so any changes to the test scripts do not affect the test data.

Disadvantages: Highly experienced testers are required to identify the relevant usage of test data to the test scripts framework design.

Keyword-Driven Framework

It is also known as the table-driven testing. This framework is ideal for small projects or applications. Similarly, the test data and scripts are kept in separated files. The automation test scripts executed are based on the keyword actions such as “click”, or login”...

Advantages: Maximum the code reusability with the test scripts built independently. The keyword-driven approach provides the abstraction from the complexities and technical challenges, so it is easy to maintain. The keyword-driven approach allows executing and creating

automated test cases for non-technical testers, business analysts, and SMEs¹ (Subject Matter Expert) to write automated tests.

Disadvantages: It takes the initial time and knowledge of designing to set up the framework . Without good documentation and knowledge sharing, the new team members have to put efforts to understand the framework and its design.

Hybrid Testing Framework

This framework allows the combination of the two frameworks as mentioned above to leverage the advantages and remove the weaknesses of each framework.

Advantages: Hybrid framework leverages all the advantages of all types of the related frameworks.

Disadvantages: Though it mitigates the weaknesses of the other approaches, it loses its generosity and is very specific to the test automation solution AUT², which reduces the reusability of its components across the multiple AUTs.

III. STEPS TO EVALUATE AND CHOOSE A FRAMEWORK

Every application or project is different, so it is crucial to have a sufficient assessment before coming to the decision of picking up a test automation framework. There are some suggested criteria to evaluate and choose the suitable framework. (Thom Garrett, Bernie Gauf, Elfriede Dustin, 2009)



FIGURE 3. STEPS TO EVALUATE AND CHOOSE A FRAMEWORK

¹ SME: a person who has special skills or knowledge on a particular job or process.

² AUT: Application Under Test

- Identify the tool requirements and criteria (the framework criteria)
- Identify a list of tools that meet the criteria
- Assign a weight to each tool criterion based on importance or priority.
- Evaluate each tool candidate and assign a score.
- Multiply the criterion weight by each tool candidate score to get the tool's overall score for comparison.
-

More concrete criteria could be considered for evaluating a testing tool as follow:

- **Usability:** Can the testing team easily adapt to the training and master the tool? Does it require a lot of programming skills? Or is there any user support or training?
- **Documentation:** Consider the features available for the documentation process with easily searchable and exportable test results and the test suites maintenance.
- **Script languages, Cross-platform:** Is the tool working with the various scripting languages, running on various browsers?
- **Version control and Integration:** Can testing be managed historically, on different files or folders, and can it revert local changes if needed? Is it able to integrate with other bug tracking tools or support different types of operating systems?
- **Value for money:** is the price clear, flexible for the features, capabilities or user cases? Is there a big difference of saving amount on the testing resources?

IV. TOP LANGUAGES FOR TEST AUTOMATION FRAMEWORK

Among hundreds of programming languages, there are some most popular languages used in designing and developing the test automation scripts such as Python, Java, JavaScript, C#, PHP... These languages are considered to be syntax-friendly and easier-to-understanding. Each of them solves various purposes. (Kumar, 2020)

According to the popularity of programming languages in Google analyzed by Github, the index shows Python is still ranked as the top programming language for tutorial searches up to May 2021. (Github, n.d.)

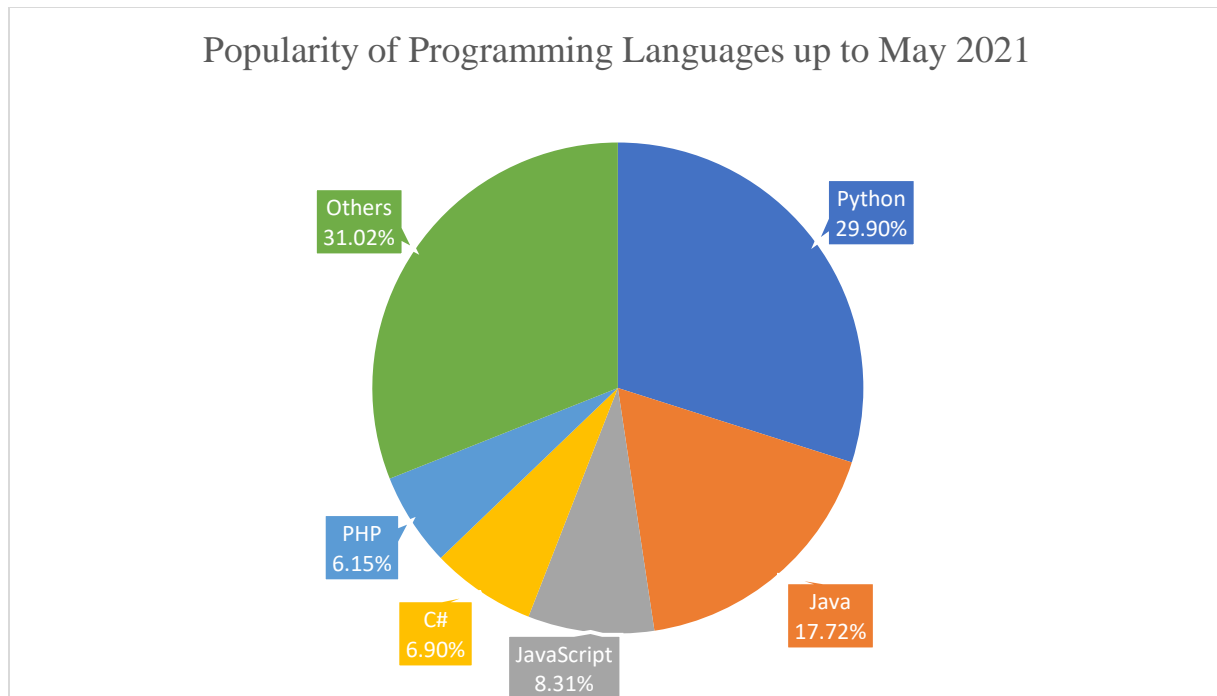


FIGURE 4. POPULARITY OF PROGRAMMING LANGUAGES. SOURCES: GITHUB, GOOGLE TRENDS

Python

Python is an open-source programming language. It is used commonly for data visualization, building web applications and more. Its libraries such as NumPy, SciPy, TensorFlow are used for developing machine learning algorithms. Python frameworks for test automation include Python Selenium, PyUnit and pytest are popular to perform cross-browser testing on desktop and mobile devices.

Many manual testers find Python helpful in learning coding as its human-readable syntax in its scripts.

Java

Java is a general-purpose programming language that is owned by the Oracle Corporation. With platform of test frameworks, packages and educational resources, Java is one of the best programming languages for the test automation. It is trusted and adopted by most organizations due to the stability in Java codes. JUnit with Selenium WebDriver is a popular unit testing framework developed using Java for the automated testing of the web products.

It also provides a powerful command line, an easy build integration, IDE support.

JavaScript

It is a text-based programming language for test automation for web applications. JavaScript with Selenium WebDriver could perform the automated UI testing of applications. It also provides end-to-end testing with the efficiency in its structured functions and patterns, making the test script compact.

C#

It is a well-known programming language created by Microsoft. It uses the .NET framework in processing the test scripts for automation testing. All the applications that run on Windows, iOS, or Android can be effectively tested with the integration of C# programming language.

C# also has the ability to integrate with the Selenium WebDriver to make the automation testing job easier and more convenient.

PHP

The acronym for Hypertext Processor and a server-side scripting language, PHP is used for unit testing and end-to-end testing in the behavior-driven³ and test-driven⁴ development.

V. COMPARISON ON SOME POPULAR AUTOMATED TESTING TOOLS

With a variety of the automated testing tools on the market today, it is crucial to select the right tool for the right project or application to maximize the beneficial outcomes of the test automation. Here are the top automated testing tools suggested by the test solution experts. (Singh, 2020)

³ Behavior-driven development (BDD): a software development process that encourages collaboration among developers, QA, and non-technical or business participants in a software project. (Wikipedia, Behavior-driven development, n.d.)

⁴ Test-driven development (TDD): a software development process relying on software requirements being converted to test cases before software is fully developed and tracking all software development by repeatedly testing the software against all test cases. (Wikipedia, Test-driven development, n.d.)

No	Tool	Pros	Cons
1	Selenium	<ul style="list-style-type: none"> ✓ Automation testing tool for web application ✓ Support various browsers, operating systems, and a considerable amount of programming languages. ✓ Create complex and advanced automation scripts ✓ Support parallel test execution reducing the test execution time. ✓ License: Open-source 	<ul style="list-style-type: none"> ✗ Cannot support non-web-based applications like Oracle apps ✗ Poor documentation ✗ Not reliable technical support ✗ Need at least the basic programming skills to utilize the framework
2.	Appium	<ul style="list-style-type: none"> ✓ Automate any mobile app from any language and any test framework ✓ Testing any mobile app from any language and any test framework ✓ Drive iOS, Android, and even Windows apps using the WebDriver protocol ✓ License: Open-source 	<ul style="list-style-type: none"> ✗ Run both iOS and Android only on macOS; Windows only supports Android. ✗ Requires some technical and programming skills ✗ Installation needs multiple steps and can be complex
3.	Cucumber	<ul style="list-style-type: none"> ✓ Code can be executed on different frameworks like Selenium. ✓ Test code is written in simple English called Gherkin.⁵ ✓ It supports languages like Ruby, Java, Scala, Groovy, etc. ✓ Fill gaps between business and testers. 	<ul style="list-style-type: none"> ✗ Mostly used for the unit testing in the BDD⁶ environment, so it requires programming skills in Java or JavaScript. ✗ Writing code to handle every scenario is complex and time-consuming.

⁵ Gherkin: a business readable language which describes business behavior without going into details of implementation. It uses plain language to describe use cases and allows users to remove logic details from behavior tests. (Guru99, n.d.)

⁶ See footnote #3 above

No	Tool	Pros	Cons
		✓ License: Open-source	
4.	SoapUI	<ul style="list-style-type: none"> ✓ Scripts can be reused easily. ✓ Has drag and drop, point-and-click test generation. ✓ Allows asynchronous testing. ✓ Cool reporting dashboard ✓ License: Open-source 	<ul style="list-style-type: none"> ✗ Bad cache management and collection ✗ Not helpful with the web UI⁷ testing and mobile app testing.
5.	TestComplete	<ul style="list-style-type: none"> ✓ Scripts can be reused easily. ✓ Creates complex automated test scripts without writing a single line of code. ✓ Easy-to-use visual record and playback. 	<ul style="list-style-type: none"> ✗ Lacks some areas like loading the script library during run time. ✗ Need experience to use it. ✗ Not have option in the issue tracking to connect to the defect tracking tool like JIRA. ✗ Though it supports to various scripting languages like Python, JavaScript, C# but not support for Java which is widely used in most devices. ✗ License: Proprietary

⁷ UI: user interface

VI. CONCLUSION

According to the MarketsAndMarkets report, the global automation testing market size is expected to grow to USD 28.8 billion by 2024 (Markets, 2019). With the benefits previously mentioned above, test automation frameworks prove to be the major significance in building robust product with the increased quality and speed.

Though there are not mandatory rules of applying a test automation framework, it is essential to follow the proper guidelines and criteria when evaluating and choosing the right framework to achieve the most beneficial outcome: a standardized test scripts which can be reusable in various scenarios, cost and time reduction on test execution, better maintenance of the test code, the higher and more accurate rate in test coverage.

Testing is an essential part in any software development cycle; thus, it is also a critical matter to determine on how the tests are run. A thorough analyzing and understanding on different types of automation frameworks are highly recommended before selecting one framework for a project or an application development.

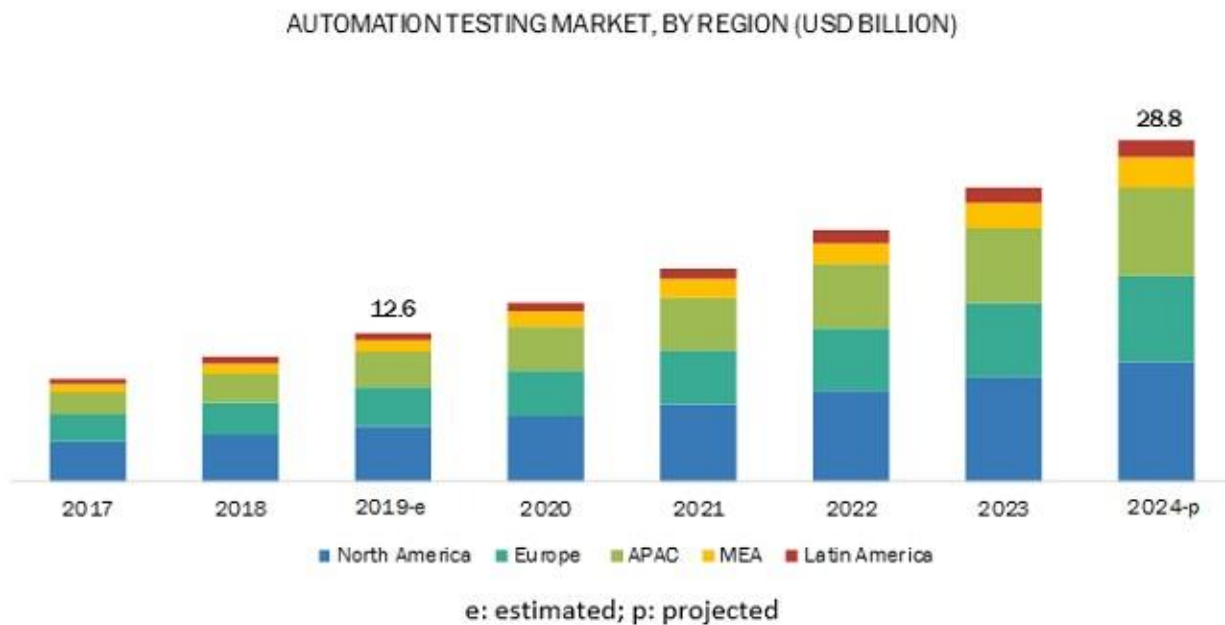


FIGURE 5. AUTOMATION TESTING MARKET SIZE. SOURCE: MARKETSANDMARKETS (MARKETS, 2019)

BIBLIOGRAPHY

- Aebersold, K. (n.d.). *Test Automation Frameworks*. Retrieved from Smart Bear Blog:
<https://smartbear.com/learn/automated-testing/test-automation-frameworks/>
- Github, P. (n.d.). *PYPL PopularitY of Programming Language*. Retrieved from PYPL Github:
<https://pypl.github.io/PYPL.html>
- Guru99. (n.d.). *Gherkin Language: Format, Syntax & Gherkin Test in Cucumber*. Retrieved from Guru99:
<https://www.guru99.com/gherkin-test-cucumber.html>
- Guru99.com. (n.d.). *Test Automation Framework: What is, Architecture & Types*. Retrieved from Guru99:
<https://www.guru99.com/test-automation-framework.html>
- Introducing frameworks. (2013). In A. Bhargava, *Designing and Implementing Test Automation Frameworks with QTP*. Packt Publishing.
- Kelly, M. (2003). Choosing a test automation framework. *IBM - Developer Works*.
- Kumar, S. (2020, September). *Best Programming Languages in Test Automation*. Retrieved from Dataflog: <https://dataflog.com/read/best-programming-languages-test-automation/9411>
- Markets, M. A. (2019). *Automation Testing Market - Global Forecast to 2024*. Retrieved from Markets And Markets: <https://www.marketsandmarkets.com/Market-Reports/automation-testing-market-113583451.html>
- Singh, S. (2020). *Top 10 Automation Testing Tools in 2021*. Retrieved from Net Solutions:
<https://www.netsolutions.com/insights/top-10-automation-testing-tools/>
- TestingXperts. (2020, March 17). *Test Automation Freamworks - Why, Types, Benefits, Approach*. Retrieved from TestingXperts Blog: <https://www.testingxperts.com/blog/test-automation-frameworks>
- Thom Garrett, Bernie Gauf, Elfriede Dustin. (2009). The right tool for the job. In *Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality*. Addison-Wesley Professional.
- Wikipedia. (n.d.). *Behavior-driven development*. Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/Behavior-driven_development
- Wikipedia. (n.d.). *Test-driven development*. Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/Test-driven_development