# VIEWPOINT

## Choosing the right automation tool and framework is critical to project success


- Harsh Bajaj,
*Technical Test Lead ECSIVS, Infosys*

Infosys®

## Introduction

Organizations have become cognizant of the crucial role of testing in the software development life cycle and in delivering high quality software products. As the competition in the IT sector grows stiffer, the pressure to deliver larger number of high quality products with fewer resources in limited time is increasing in intensity.

During development cycles software tests need to be repeated to ensure quality. Every time the source code is modified, test cases must be executed. All iterations in the software need to be tested on all browsers and all supported operating systems. Manual execution of test cases is not only a costly and time-consuming exercise, but it is also prone to error.

Automation testing addresses these challenges presented by manual testing. Automation tests can be executed multiple times across iterations much faster than manual test cases, saving time as well as cost. Lengthy tests which are often skipped during manual test execution can be executed unattended on multiple machines with different configurations, thus increasing the test coverage.

Automation testing helps find defects or issues which are often overlooked during manual testing or are impossible to detect manually – for example, spelling mistakes or hard coding in the application code. Automaton also boosts the confidence of the testing team by automating repetitive tasks and enabling the team to focus on challenging and high risk projects. Team members can improve their skill sets by learning new tools and technologies and pass on the gains to the organization.

The time and effort spent on scientifically choosing a test automation product and framework can go a long way in ensuring successful test execution. Let us take a closer look at various factors involved in the selection process.

## Identify the right automation tool

Identification of the right automation tool is critical to ensure the success of the testing project. Detailed analysis must be conducted before selecting a tool. The effort put in the tool evaluation process enables successful execution of the project. The selection of the tool depends on various factors such as:

- The application and its technology stack which is to be tested
- Detailed testing requirements
- Skill sets available in the organization
- License cost of the tool

There are various functional automation tools available in the market for automating web and desktop applications. Some of these are:

| Tool | Description |
|---|---|
| QTP(Quick Test Professional) / UFT (Unified Functional Testing) | Powerful tool from HP to automate web and desktop applications |
| Selenium | Open source automation tool for automating web applications |
| Watir (Web Application Testing in Ruby) | Open source family of Ruby libraries for automating web browsers |
| Geb | Open source automation tool based on Groovy |

## Comparison Matrix

While analyzing various automation tools, a comparison of key parameters helps select the right tool for the specific requirements of the project. We have created a comparison chart of tools listed above based on the most important parameters for automation projects. Organizations can assign values to these parameters as per their automation requirements. The tool with the highest score can be considered for further investigation.

| Parameters | Criteria | UFT | Selenium | Watir | GEB |
|---|---|---|---|---|---|
| **Ease of Adoption** | License cost | QTP is HP licensed product available through single-seater floating or concurrent licenses | Selenium is an open-source software and is free | Watir is an open-source (BSD) family of Ruby libraries for automating web browsers and is free | GEB is open source software based on Groovy and is free. |
| | Ease of support | Dedicated HP support | User and professional community support available | Limited support on open source community | Limited support on open source community |
| **Ease of Scripting and Reporting Capabilities** | Script creation time | Less | Much more | More | More |
| | Scripting language | VB Script | Java, CSharp, Python, Ruby, Php, Perl, JavaScript | Ruby | Groovy |
| | Object recognition | Through Object Spy | Selenium IDE, FireBug, FirePath | OpenTwebst (web recorder) | GEB IDE |
| | Learning time | Less | Much more | Much more | Much more |
| | Script execution speed | High | Low | Low | Low |
| | Framework | In-built capability to build frameworks such as keyword-driven, data-driven and hybrid | JUnit, NUnit, RSpec, Test::Unit, TestNG, unittest | Ruby supported Frameworks - RSpec, Cucumber, Test::Unit | Grails, Gradle, Maven |
| | Continuous integration | Can be achieved through Jenkins | Achieved through Jenkins | Achieved using Ruby script | Achieved using Grails, Gradle plug-in along with Jenkins Gradle plug-in |
| **Tools Usage** | Non browser-based app support | Yes | No | No | No |
| | Operating system support | Windows 8/8.1/7/ XP/Vista (No other OS) | Windows, MAC OS X, Linux, Solaris (OS support depends on web-driver availability) | Windows 8.1, Linux 13.10, MAC OS X 10.9, Solaris 11.1 (need JSSH compiled) | Windows XP/Vista/7, Linux, DOS (OS support depends on web-driver availability) |
| | Browser Support | IE (version 6-11), Firefox (version 3-24), Chrome (up to version 24) | Firefox, IE, Chrome, Opera, Safari | Firefox, IE, Chrome, Opera, Safari | Firefox, IE, Chrome, Opera, Safari |
| | Device Support | Supports iOS, Android, Blackberry, and Windows Phone via licensed products such as PerfectoMobile and Experitest | Two major mobile platforms iOS and Android | Two major mobile platforms iOS and Android | Driver available for iOS (iPhone and iPad) |

## What is a test automation framework?

A test automation framework is a defined, extensible support structure within which the test automation suite is developed and implemented using the selected tool. It includes the physical structures used for test creation and implementation as well as the logical interaction between components such as:

- Set of standards or coding guidelines, for example, guidelines to declare variables and assign them meaningful names
- Well-organized directory structure
- Location of the test data
- Location of the Object Repository (OR)
- Location of common functions
- Location of environment related information
- Methods of running test scripts and location of the display of test results

A well-defined test automation framework helps us achieve higher reusability of test components, develop the scripts which are easily maintainable and obtain high quality test automation scripts. If the automation framework is implemented correctly it can be reused across projects resulting in savings on effort and better return on investment (ROI) from the automation projects. Let us discuss some key frameworks available in the industry today.

After analyzing the pros and cons of these frameworks, most companies opt for the hybrid framework. This allows them to benefit from multiple best functional test automation frameworks available in the industry.

### Modular Framework

- In the modular approach reusable code is encapsulated into modular functions in external libraries. These functions can then be called from multiple scripts as required.
- This framework is well suited in situations where the application includes several reusable steps to be performed across test scripts.

### Data-driven Framework

- In the modular approach reusable code is encapsulated into modular functions in external libraries. These functions can then be called from multiple scripts as required.
- This framework is well suited in situations where the application includes several reusable steps to be performed across test scripts.

### Hybrid Framework

- The hybrid model is a mix of the data-driven and modular frameworks
- This framework mixes the best practices of different frameworks suitable to the automation need.

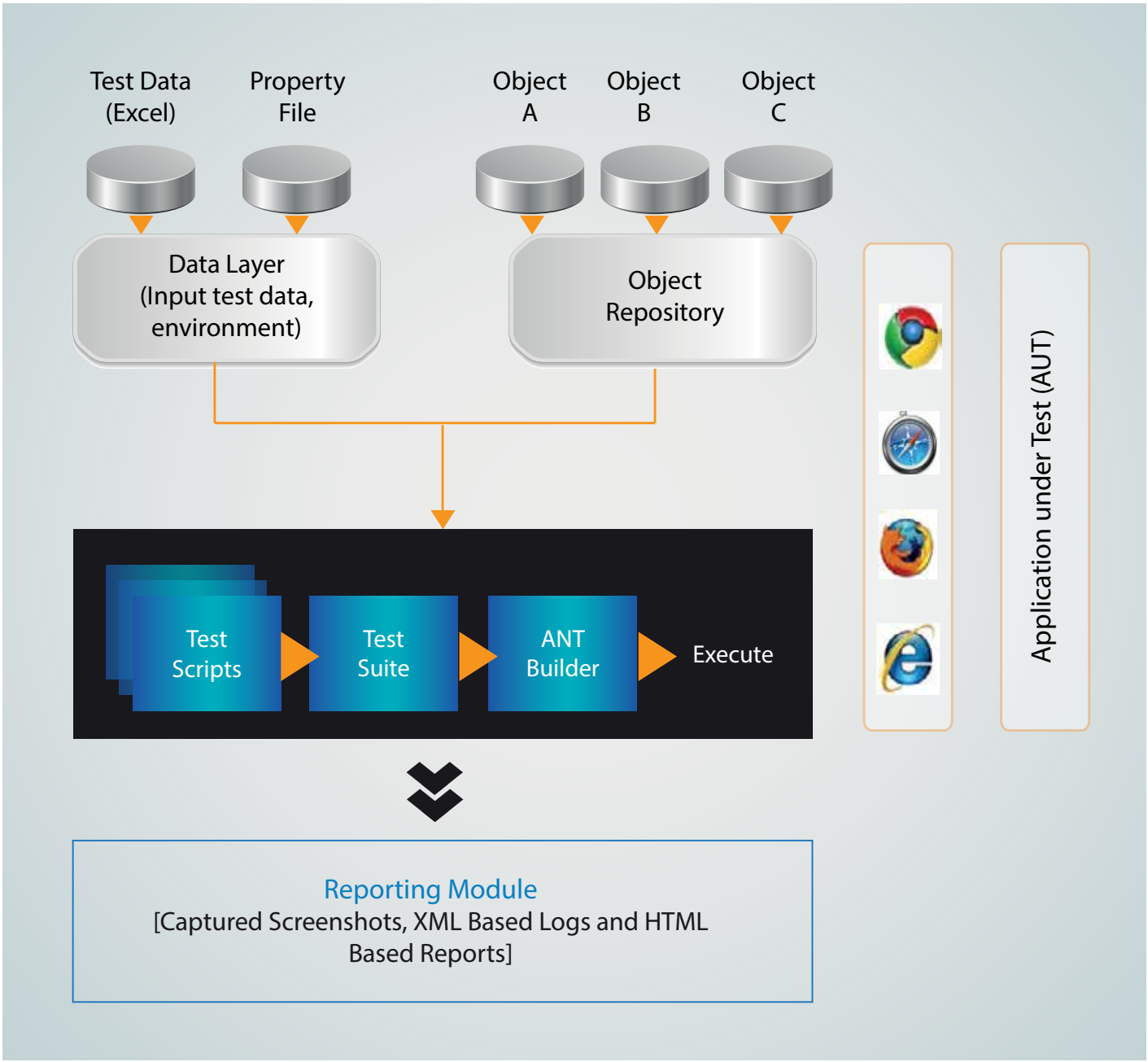
### Keyword-driven Framework

- In the keyword-driven framework, a keyword is identified for every action that needs to be performed and the details of the keyword are given in a spreadsheet.
- This framework is more useful for non-technical users to understand and maintain the test scripts.
- Technical expertise is required to create a complex keyword library
- Creating such a framework is a time-consuming task.

## Implementing Hybrid Framework with Selenium

Now let us see how to implement a hybrid framework with Selenium as an automation tool. The key points in the implementation are:

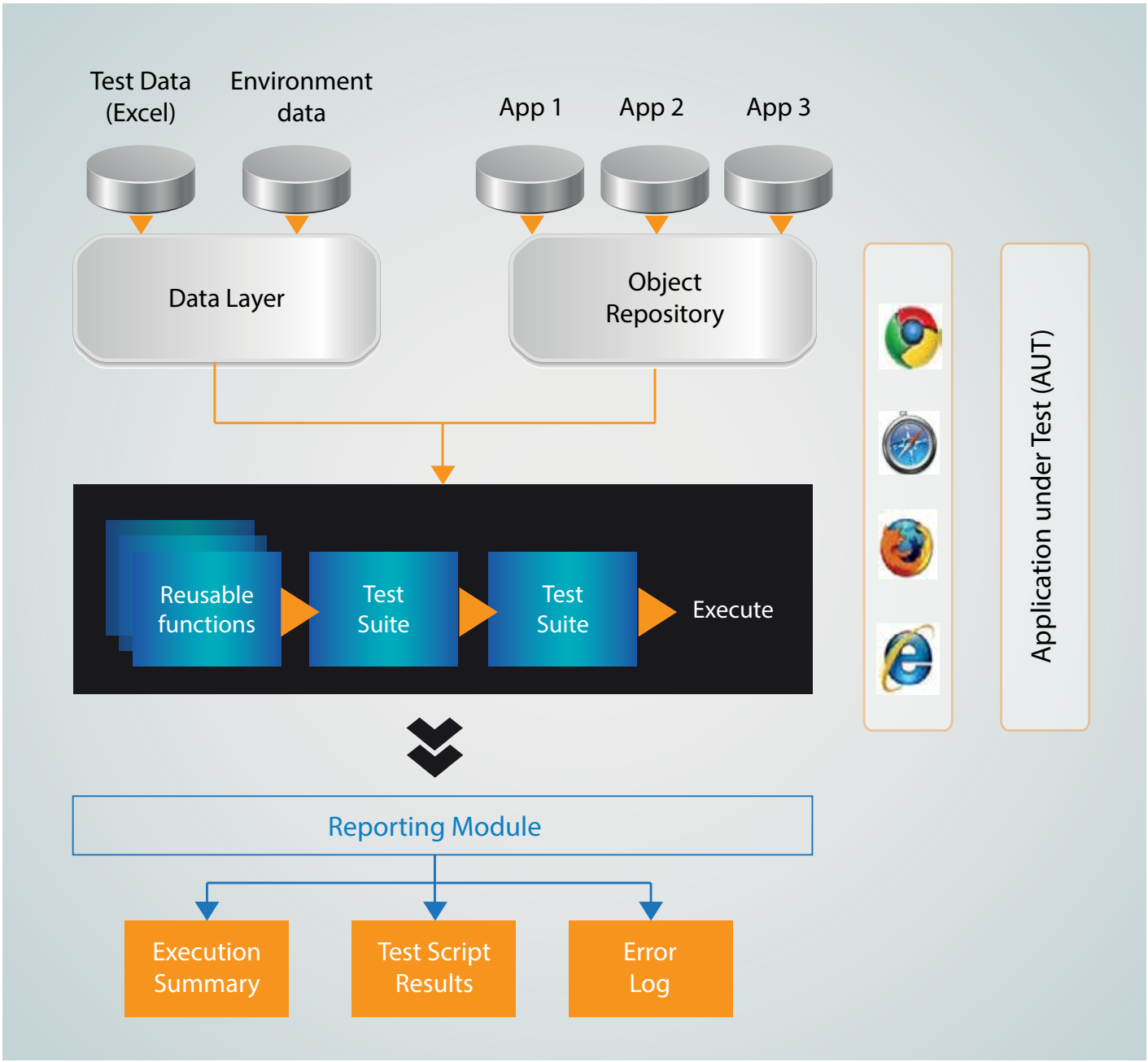
- Store the test data (any user input data) in an Excel file.
- Store the environment related information (for example, QA, UAT, Regression) in a property file.
- Store various objects in the application on which user action needs to be taken in object repository files.
- Test suite contains the logic to verify acceptance criteria mentioned in the requirement.
- Execute the script on various browsers as per the need.
- Generate the reports capturing screenshots and pass/fail results. To get advanced reports in Selenium use any testing framework such as TestNG, JUnit.



## Implementing Hybrid Framework with QTP/ UFT

Let us discuss how to implement a hybrid framework with QTP/ UFT as an automation tool. The key points in the implementation are:

- Store the test data (any user input data) in an Excel file.
- Create a run manager sheet to drive the test execution.
- Store the environment-related information in property file.
- Store the objects in the application on which user actions are taken in object repository files.
- Divide the test cases into modular functions, keeping common functions separately to be used across projects.
- Include main script common functions, object repository and test data. Generate different types of reports as per the business requirements.



## Executing Proof of Concept for the Selected Tool

The last phase of tool evaluation is proof of concept (PoC). The tool selected may satisfy your criteria conceptually, but it is advisable to test the tool using a few scenarios. Almost every tool vendor provides an evaluation version of their tool for a limited time period. The following steps need to be considered during the PoC:

- Choose a few scenarios in such a way that they cover different objects and controls in the application.
- Select the tool(s) based on a comparative study.
- Automate the chosen scenarios using the selected tool(s).
- Generate and analyze various reports.
- Analyze the integration of the tool with other tools such as test management tool available, for example, QC (Quick Center) and with continuous integration tools such as Jenkins.

While evaluating multiple tools, generate a score-card based on various parameters such as ease of scripting, integration, usage, reports generated and chose the tool with the maximum score. When the POC is completed, the team can be more confident about successfully automating the application using the selected tool.

## Conclusion

The process of automation framework design and development requires detailed planning and effort. To achieve the desired benefits, the framework must be accurately designed and developed. Such a framework can then be used across projects in an organization and provides substantial ROI. When choosing an automation framework, it is crucial to ensure that it can easily accommodate the various automation testing technologies and changes in the system under test.

One of the key factors contributing to the accomplishment of any test automation project is identifying the right automation tool. A detailed analysis in terms of ease of use, reporting and integration with various tools must be performed before selecting a tool. Though such selection processes call for focused effort and time, this investment is worth making because of the great impact it has on the success of the automation project.

## About the Author

Harsh Bajaj is a Technical Test Lead with Infosys Independent Validation Services Team. She has led various test automation projects in the telecom domain. She is proficient in various test management and automation tools.

## About the Reviewer

Gautham Halambi is a Project Manager with Infosys and has more than 9 years of experience. He has worked on multiple telecom testing projects and has led and managed large manual and automation testing teams.

## References

www.seleniumhq.org

www.watir.com,

http://www.gebish.org/

http://www.automationrepository.com/

http://en.wikipedia.org/wiki/Keyword-driven_testing

For more information, contact askus@infosys.com