Contents

1 Executive Summary ........................................................................................................................................... 3
2 Current State of Test Automation .................................................................................................................. 3
3 Challenges in Test Automation ....................................................................................................................... 4
4 Selecting the Right Automation Framework .................................................................................................. 6
5 The Tavant Approach ..................................................................................................................................... 7
  5.1 Advantages of FIRE ....................................................................................................................................... 7
  5.2 Under the Hood – A Look at FIRE’s Structure .............................................................................................. 8
  5.3 Generating Return on Investment (ROI) ...................................................................................................... 9
  5.4 Test Automation Success Story ................................................................................................................ 10
6 Conclusion ....................................................................................................................................................... 10
7 References ....................................................................................................................................................... 11
1 Executive Summary

As businesses try to cope with a dynamic, demanding marketplace, the pressure to deliver high quality software in smaller windows of time is ever-increasing. When it comes to technology, organizations are realizing the importance of software testing and the role it plays in developing quality code with speed and agility.

Organizations today face several Quality Assurance (QA) challenges – time constraints in development and test cycles, executing large volumes of test cases, testing diverse legacy applications, and mitigating the impact of ripple effects that arise from configuration changes in application modules. The most efficient way to deal with this situation is to adopt a well-integrated and robust automation solution that can predict and simulate business scenarios. Coupled with an appropriate strategy, approach and process, a sound automation framework can deliver significant benefits. However, it is critical that organizations choose the right approach for test automation.

This paper explores the current state of test automation, highlights the challenges faced by organizations, and underscores the need to select the right framework. It also introduces FIRE – a test automation framework – and takes a look at how it can help businesses become more competitive.

2 Current State of Test Automation

Recent times have seen companies embracing new trends such as e-commerce, social media marketing, Big Data analytics, Cloud-based solutions, mobile marketing, and mobile app development. Best-of-breed vendors help them quickly on-board technology solutions, allowing them to drive greater business value with agility.

However, for companies to attain business agility, they first need to embed agility in their underlying software and business processes. Here, the role of test automation becomes critical. The evolution of test automation has been a long journey from crudely stitched code that provided specific or stop-gap fixes to today’s well-defined and structured automation test suites.

Companies that adopt test automation still contend with several challenges such as identifying the optimum amount of test automation coverage, maintaining up-to-date automation test cases, and creating reusable test scripts. Further, there are several approaches for test automation and most companies are often unsure of which is best suited for their technology landscape.

Choosing the right approach presents a significant challenge as a single automation framework cannot meet all the needs of a project. Additionally,
the implementation of an automation framework depends on the type of application and the end-goals of the automation process. Thus, the adoption of test automation solutions often requires significant customization for different projects, making it a tedious task that involves considerable manual effort. These approaches have led to successful solutions for domains like Gaming, Consumer Lending, Media & Entertainment, Manufacturing, etc. The deployment of the framework has been approximately 30% each for Gaming and Consumer Lending while 14% in manufacturing projects.

Let us examine in detail the challenges in test automation and how companies can choose the right approach to ensure business agility.

3 Challenges in Test Automation

Despite the known business benefits of test automation, it is important to understand that test automation is not a magic wand. There are several factors that can compromise the integrity of a test automation solution. For instance, automation engineers responsible for deciding whether an organization should adopt test automation frequently overlook factors that may impact the success. The most common reasons for failure of test automation are:

- **Lack of experience in framework design and/or implementation**
- **Inability to easily incorporate enhancements**
- **Absence of Proof-of-Concept (POC) or pilot phases**
- **Lack of in-depth analysis on test reports or logs**
- **Lack of clarity on what to test and what to exclude**
- **Absence of exceptions and error recovery functionalities within the framework**
- **Improper maintenance of the object repository**

Further, the decision to automate is complex and fraught with several concerns such as (i) Does the organization need an end-to-end automation suite? (ii) What is the optimum amount of automation? How is it to be achieved? (iii) Are test cases required for all functions in the application—big or small?

There are no one-size-fits-all answers to these questions since they depend on subjective factors such as the teams, projects and the organization itself. The best solution is one that fulfills the unique requirements of an organization and drives collaboration between development, QA and business development teams. This can ensure transparent and effective communication on time frames and deliver the required scope and quality of end-user experience. The absence of efficient communication between
these three teams can lead to critical process failures. Thus, companies first need to ensure that they have identified the ‘right reasons’ for adopting test automation.

Once organizations have identified the need for automation, they face several challenges when it comes to implementation. These include:

> **Poorly designed software architecture** – Just as a building requires a stable foundation, software needs to be developed on robust and sound architecture. The failure to incorporate key scenarios, design common problems, create risk mitigation, and factor in consequences of long-term key decisions can jeopardize the integrity of any application. While companies can leverage modern tools and platforms to simplify the task of building applications, there is a need to proceed with utmost precision and care when designing applications to ensure that all scenarios and requirements are covered. Lack of such careful monitoring can result in poor architecture leading to software that is unstable, unable to support existing or future business requirements and is a challenge to deploy or manage in a production environment.

Consider the following example: A leading software provider had initiated an automation solution without a clear vision, resulting in a poorly designed framework. This led to several complications during the implementation phase as the framework failed to support several objects/flows and hampered the building of scripts. This resulted in significant time and cost spent on fixing the framework instead of focusing on the deliverables.

> **Improper identification of automation goals** – When deciding on the type of automation framework to be implemented, it becomes critical to first identify the automation goals of the organization. The framework should be chosen based on its capabilities to meet these goals. Companies that do not map their goals to the framework’s features run the risk of incomplete automation coverage and increased costs.

> **Choosing the wrong automation framework** – Data-driven and Keyword frameworks are two of the most commonly used frameworks. Each has its advantages as well as disadvantages. Companies need to be aware of the differences between such automation frameworks in order to choose the one best-suited to fulfill their organizational goals.

When planning to select an automation framework, companies must ensure that the framework provides object support, tool support and enables customization. Improper estimation and groundwork can result in a negative automation ROI.
4 Selecting the Right Automation Framework

Let us examine the two most commonly used frameworks – Data-driven and Keyword-driven frameworks.

**Data-driven frameworks** involve fetching test input and output values from data files such as Excel, Concurrent Version Systems (CVS), and Open Database Connectivity (ODBC) source. Here, the test cases are encapsulated in the data file while the test script becomes a driver for data delivery. This framework requires multiple data files for each test case and tester expertise in the scripting language.

On the other hand, a **Keyword-driven framework** requires one to create a set of keywords and data tables that do not depend on the automation tool used by the organization or on the test scripts powering the data and the application being tested. For this framework, the functionalities of the application being tested are captured and translated into meticulous steps for each test. Here, tester expertise in the scripting language as well as knowledge of several special formats and keywords is needed to create customized utilities for the application.

When making the choice between either of these frameworks, companies should be aware that testers face certain limitations. For instance, they are allowed to use only the keywords/methods provided by the specific framework. They need to invest time to understand the workings of the framework such as setting up data, using keywords, etc. Thus, the final choice of the right framework must involve a careful analysis of the goals of the organization along with an understanding of what is available in the market.

The best-fit automation framework is a single standard solution that addresses existing and future challenges, delivers optimum test coverage, and ensures time and cost efficiencies. It should deliver concrete results such as a unified test library, standardized scripting, maximum reusability of scripts, etc. It should simplify user adoption by being intuitive about project requirements and enable all users – including non-technical users – to participate in creating and maintaining test scripts. The right framework should contain:

> Well-designed structures that provide flexibility for enhancements and changes
> In-built team member flexibility that allows any user to add or modify scripts
> Easy-to-use test data and data pool integration
> Cross browser support
Flexible Approach to Test Automation with Measureable ROI

- Remote machine test execution
- Parallel test execution support
- User-friendly customized reports
- Integration with nightly builds
- Integration with bug tracking tools
- Easy maintenance of the object repository

5 The Tavant Approach

To overcome the stated challenges and deliver benefits of automation, Tavant Technologies has designed a ready-to-use automation platform that can be employed in a variety of projects with minimal customization and maintenance. The Framework for Intelligent and Rapid Execution – also known as FIRE – imbibes best practices and implementation techniques from multiple automated testing approaches. Such a framework can facilitate various levels of testing for an enterprise application. Developed using open source tools such as Selenium2/WebDriver and TestNG frameworks, FIRE presents a new type of easy-to-use solution that can automate functional and regression testing.

Based on the ‘Zero-D’ philosophy, which represents no show-stoppers, FIRE provides testing capabilities based on the Six Sigma approach. Its key differentiator is a flexi-hybrid framework that simplifies a broad range of testing needs from unit testing – testing a component in isolation or implementing layered modular testing – to integration testing, which involves testing entire systems made of several components, external interfaces and their integration. This solution approach can integrate and automate all activities, resources, tools, and methodologies used in different phases of testing, thereby actualizing the ‘Zero-D’ philosophy.

5.1 Advantages of FIRE

Some of the key features and advantages of using FIRE for functional automation are as follows:

- Application centralization by enabling a common platform for User Interface (UI), Web Service/Application Programming Interface (API), Mobile, and Database Testing along with a centralized object repository
- Comprehensive testing capabilities based on a flexi-framework that enables rule-based testing, data-driven testing and Data Warehouse and Business Intelligence (DW/BI) testing. It also enables running tests in sequential, parallel or remote machines with a single property change and captures screenshots on test case failure. This enhances user involvement in configuring and executing automated tests.
- Easy test script maintenance and enhancement with options for reusable methods to be written in Excel as a function/methods that can be later called in test scenarios when required. The scripts
and test data can be maintained independently. A reusable library helps in developing test scripts and reduces the time and effort spent on utility building.

- Customizable test reports that are user friendly, detailed and integrate easily along with dashboards that feature defect management tools
- Continuous integration with build tools to allow easy integration of test scripts and automatic scheduling of nightly builds for test execution
- Extensive support capabilities for Android Mobile Web Application automation and behavior-driven development. It also supports parallel, cross-browser, cross-platform and remote execution; enabling faster ROI.
- Configurable and externalized test data pool and test environment settings

For media organizations, the multi-channel effect is phenomenal, opening up several new streams for content delivery to reach a larger audience base

**Advantages of FIRE - Snapshot**

- Application centralization
- Comprehensive testing
- Easy test script maintenance and enhancement
- Customizable test reports
- Continuous integration with build tools
- Extensive support for Android Mobile Web Application automation and behavior-driven development
- Configurable and externalized test data pool and test environment settings.

### 5.2 Under the Hood – A Look at FIRE’s Structure

FIRE caters to three types of users:

- **Novice Users** – Users that possess minimal programming knowledge, can write commands in simple English and execute test scripts.
- **Moderately Skilled Users** – Users with basic knowledge of programming but cannot write plug-ins. These users can utilize existing scripts and execute them.
- **Expert Users** – Users that are fluent with a scripting language, can write their own plug-ins, and integrate with the framework that is used in day-to-day scripts.
5.3 Generating Return on Investment (ROI)

FIRE drives efficiencies in automation resulting in radical cost and time savings. These have a direct impact on Return on Investment (RoI) as shown below:

> **Reduced script development time** – FIRE reduces the time taken for script development by up to 60%. This means that automation scripts that take 500 hours in a manual environment will require only 200 hours when employing the FIRE automation solution. Such tremendous time savings also pertain to updating scripts based on user interface changes.

> **Reduced test cycle time** – FIRE reduces test cycle time by up to 60-70%. For instance, against a manual effort of 520 hours for 1000 test cases, FIRE can do it in approximately 303 hours including test development and automation development. The time taken to execute the tests in this scenario with FIRE is 3 hours against 320 hours taken manually.
5.4 Test Automation Success Story

A large mortgage company was contending with several challenges in managing their mortgage application. They were struggling to handle complex business problems involving innumerable test scenarios and hundreds of variables with the current application. Further, the system was unable to process short-term client requirements such as weekly changes on short release cycles, leading to longer turnaround times. The client was looking for an automation solution that could validate large amounts of input data and deliver a mission-critical system with a high degree of precision.

The client chose Tavant to deliver an automation solution with advanced data processing capabilities. Tavant leveraged its FIRE methodology to design a system that employed a risk-based approach to identify business scenarios. Further, by using a combination of technology stack implementations, Tavant was able to ensure compatibility with disparate systems and enable seamless integration with continuous build tools. The solution helped the client reduce test execution time by up to 74%.

Figure showing the benefits of implementing the automation solution

<table>
<thead>
<tr>
<th>Manual Testing</th>
<th>FIRE Automation Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of scenarios</td>
<td>23000</td>
</tr>
<tr>
<td>Avg. manual execution and validation of scenarios (hrs)</td>
<td>383.33</td>
</tr>
<tr>
<td>Avg. time for updating test cases (hrs)</td>
<td>10</td>
</tr>
<tr>
<td>No. of regression cycles</td>
<td>15</td>
</tr>
<tr>
<td>Total time (hrs)</td>
<td>5900</td>
</tr>
</tbody>
</table>

6 Conclusion

Organizations seeking agility in their business processes need to on-board robust test automation solutions that ensure superior software quality. Successful test automation frameworks are those that are easy to adopt, simple to maintain and enable flexibility in test data maintenance, parallel execution of test cases, remote machine test execution, and cross browser execution. However, organizations need to choose their automation framework with care to ensure that it realizes their automation objectives.
Tavant automation solution framework FIRE leverages the key success levers of available frameworks, thereby empowering organizations to benefit from best practices. The framework has proven advantages for specialized functional automation such as Mobile testing, DW/BI testing, API testing and Web service testing. With cross-browser support, continuous integration capabilities, reusable scripts, and rule-based testing, FIRE presents a host of advantages for organizations seeking to drive value and agility from superior software.

7 References


TestNG. (n.d.). Retrieved from TestNG: www.testng.org

About Tavant Technologies

Tavant Technologies is a specialized IT solutions and services provider that leverages its deep expertise to provide impactful results to its customers. With our unrivaled capabilities and domain insights, we have enabled game-changing results for leading businesses across chosen industry micro-verticals. We are known for our long-lasting customer relationships, engineering excellence and passionate employees. Founded in 2000, we are headquartered in Santa Clara, California and service customers across North America, Europe, and Asia-Pacific.

About the Author

Madan Mohan Reddy B is working as a Test Architect at Tavant Technologies. He has over 13 years of experience in the area of software Automation testing / QA. He has architected and designed in house frame works and worked on varies successful approaches in Automation deliverables. Currently he is a part of Tavant Automation COE.

Santosh Kumar CH. is working as a Test Architect at Tavant Technologies. He has over 13 years of experience in the area of Software Automation / Manual Testing. He Evaluated and recommended test automation tools and determined suitability with various automation tools for project needs. Currently he is a part of Tavant Automation COE.