Article
Game Changing Test Automation Framework

Ramesh Ramani

Tavant Technologies
Introduction

Over the last few years, test automation has matured from crude ‘record and playback’ to ‘hybrid, framework-driven’ processes. Customers’ rising interest in test automation is a clear indication of its significant benefits.

In the last 5-6 years, there has been a considerable upsurge in the usage of commercial testing tools. In their quest to cut costs incurred on commercial tools, an increasing number of customers are looking to migrate to ‘open source’ test automation tools.

One of the key criteria in choosing a test automation tool is the technology on which it is built. All tools do not support all technologies and that poses a challenge in maximizing returns on test automation tools with limited capabilities. For example, an application developed in .NET or Java may require test automation tools to support those technologies. Some tools, however, are available with specific plugins to address such challenges. So, choosing the right tool goes a long way in achieving the expected RoI.

In this article, let’s examine and understand some key questions that need to be considered to achieve higher RoI in test automation.

Did We Select the Right Tests for Automation?

RoI in test automation depends on how frequently the script is used and how likely is it to run in multiple environments without human intervention.

To start with, Smoke or high priority test are the best candidates for automation. Based on project data, priority tests are executed 15 times more than other test cases in a release. Automating the Smoke / high priority test, therefore, yields better RoI.

While selecting regression tests, care should be taken to select those tests that are important and easy to automate to achieve better RoI.

Another important aspect is dependency of the framework on test data or human intervention. It should be completely avoided to ensure that tests are integrated in a continuous integration environment and run continuously without manual intervention.

Is Technology a Challenge?

In test automation, technology is one of the biggest impediments to achieve optimum RoI. With every new patch or release of an OS, browser, languages, or for that matter even test automation tools, the challenges to maintain automation scripts are undeniably compounded.

Between the start and completion of a test automation project, there may be changes in the project environment. In such a case, the scripts that are bound to specific environment have to be reworked. This is when a robust framework and good coding practices facilitate a smooth transition.
A framework independent of test automation tools should be developed. Since each tool has some limitations, it is best to use a combination of tools to achieve the maximum returns on test automation investments.

**Is There a Maintenance Cost for Automation Scripts?**

Change requests have invariably become an integral part of all projects. Today, changes in business requirements impact the functionality and features of applications.

Any alteration in an application has a huge impact on automation scripts, and they have to be modified to accommodate the changes. Very often, the effect on application code change is considered, while the impact on automation code change is ignored.

When a change request is raised, it is important to evaluate its impact on automation scripts. A robust test automation framework mitigates the impact of such changes. Hence, a small cost for maintaining the script goes a long way in optimizing the RoI in test automation.

**Is My Framework Scalable and Flexible?**

Some of the most common test automation frameworks are data-driven, keyword-driven, modular, and hybrid. Also, there are custom built frameworks to meet specific needs of the application.

Spending enough time on choosing the right tools and designing the framework are key to successful test automation. A good and robust framework should take as many hypothetical scenarios as possible into account to achieve desired flexibility.

While it will be difficult to prove due to lack of data, it is general belief that most test automation projects fail in the long run because of their automation frameworks’ inflexibility to adapt to the changes in applications or environment.

The framework also needs to be scalable in running hundreds and thousands of suites of tests in different environments and handling different situations. Many frameworks work well with few hundred tests, but fail to take the load of automating more tests simultaneously. That is why it is very important to create a scalable and flexible framework so the changes can be managed at test case level instead of framework level.

**Conclusion**

A number of organizations take to test automation ambitiously, but fail to achieve the desired RoI. Before embarking on an automation project, selecting the right tool, tests and strategy is very important. Building a test automation framework needs skills such as designing, analysis, and strong foresight on automation.

Considering the development of test automation framework as a typical development project helps achieve optimum RoI.
Our Recommendation

Commercial vs. Open Source Tools

Both commercial and open source tools have advantages and disadvantages. Choice between the two depends on the scenario. For example:

> If the application is complex and there is a dedicated automation team that has very little time to build an automation suite, then commercial tools will be the best choice. Having said that, the cost of standard commercial tools is very high and there are many restrictions in using these tools.

> If the automation scripts are created by many people in a team or if the tool needs to be customized or the scripts need to run in multiple environments, then open source tools will be a better choice. However, a thing to be noted is that the initial learning curve of open source tools is higher than that of commercial tools.
About Tavant Technologies

Tavant Technologies is a specialized IT solutions & services provider that leverages its expertise to provide impactful results to its customers. We have leveraged our unrivaled capabilities and domain insights to create 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

Ramesh Ramani is Director - Independent Software Testing at Tavant Technologies. A technocrat with over 14 years of experience in the IT industry, Ramesh specializes in product and process automation using open source as well as commercial tools. He has customized and built frameworks that are used in testing enterprise-class applications.

As one of the early adopters of open source in software testing, Ramesh has developed a combination testing tool called JWarp, published by the same name at sourceforge.net. He has participated in various conferences and published papers in international journals, including StickyMkinds.com and TicketIT.com.