This white paper takes you through the different aspects of Test Automation in the current scenario and addresses different issues while deciding onto the automation process. And finally it talks about how Open source is a better fit for Test Automation...
Table of Contents

1 Introduction ......................................................................................................................................................... 3
2 One Time Job versus Repeatability & Flexibility ............................................................................................... 3
3 Everything cannot be automated ...................................................................................................................... 3
4 Tools ................................................................................................................................................................. 3
5 The Start Point .................................................................................................................................................. 4
6 How to identify the perfect tools and automate with right foot forward? ......................................................... 4
7 OpenSource – A dominant factor ....................................................................................................................... 4
8 What next, now? ................................................................................................................................................. 5
1. Introduction

Agile and open source tools have changed the way test automation is being implemented – Ok, but is it true? Well, the fundamental Test automation process itself has changed, thanks to the agile methodologies and its influencing trends. In general, Organization that wants successful test automation implementation needs to have 3Ws & 1H in their mind about Automation.

• Why do we need to automate?
• When do we automate?
• What do we automate?
• How to automate?

But the above mentioned questions are always dependent on one single factor - “cost” attached to automation.

• Tool Cost – License & its annual maintenance & upgrade
• Resource Cost – Human resources that organization would hire & keep
• Maintenance Cost – Automation script enhancement along with application enhancement

Somehow, these background costs tend to overshadow the value of these questions - 3Ws & 1H. Below discussed are few reasons as why cost factor is just a myth and how to clear the notions about it.

Let us first uncover couple of things about automation irrespective of costs attached to them.

Why?

One Time Job versus Repeatability & Flexibility

Many whitepapers have addressed an important question “Why do we need to go for QA automation?” I would not really go into details like, there would be benefit in long repeat cycles between development & testing; consistency between multiple application versions on similar environments etc.

Automation takes in real effort and needs to be treated as a full blown project especially when enterprises are initially building their framework and scripts. One just cannot jump of his/her seat and shout “Let’s do it”. Such, off the fly thoughts would straight away lead organizations to a crash with repercussions of never thinking about automation again.

But, what it needs is a deeper understanding as to what would be the framework, what tool should they use and how many resources do they require (hardware, software & human). A flexible automation system is designed to be used more than once i.e. reusability. The framework is designed to create, maintain & enhance modularity in automation scripts. This helps better return on investment from a long term perspective.

What?

Everything cannot be automated

Well, everything that can be manually done on a box on software can also be automated. But the effort versus output principle defines that anything that requires more effort to automate and in turn has low output should not be automated. So, we fall back to the question “What do we automate?”. The modules are usually identified based upon

• Frequency of usage at customer site
• Complexity of a module i.e. coupling versus cohesiveness

Once the modules have been identified, then we decide what functionality of automation suite is required on those modules. Should it be only data creation and data flow tests in a linear manner OR data that is created by a module/s is being used & updated by another set of modules.

This would also be dependent on the kind of framework they choose (Playback-Record, Data Driven, Keyword Driven, Hybrid approach) and also the type of applications (complexity of modules), flexibility of tools, and setting up environment or lab for a scenario or regression in double quick time.

Tools

In order to cope with Time to Market the application on one side and exhaustive regression cycles on other side, company prefers to opt for long term solution i.e. automation. One usually has two choices i.e.
Test Automation – Are Enterprises ready to bite the bullet?

commercially paid tools or open source ones.

Commercial tools have some advantages like being able to easily get started with, get automation scripts developed faster & better reports but the disadvantage is mostly on purchasing & maintenance of licensing costs. This cost is further added to the development and testing cost.

Open source on other side are zero on cost but have disadvantages that you need to invest time & effort to get it customized to match testing requirements of the application. This investment of time & effort can also be treated as Indirect cost which gets attached to the cost of the application.

When? The Start Point

The real power of automation can be leveraged when you have a framework that supports regression, smoke, functional, system and unit testing activities.

So, a thought provoking question is “When should we start automation? Usually, it is understood and considered that automation should begin once the code is frozen or at least stable to quite an extent. But in actual, the automation design starts off as soon as the design for software takes place. The framework design needs ample amount of thought process, review & dry-runs to support different phases of testing that automation scripts would support. The coding of centralized functions in automation starts off along with development code. Depending upon the tool, one can always integrate development team’s centralized libraries to reduce the rework effort in writing functions such as database accessibility, any communication between n-tier servers etc. The coding of per screen, per module scripts does start off when coding or blue print of the screen or controls is frozen.

How? How to identify the perfect tools and automate with right foot forward?

The first step to move towards automation is to select the tool that suits your needs. You need to answer few questions for selection-

• Does it support your application functionally?
• How much flexible the tool is in relation to handling of 3rd party controls used by application under test?
• Would you like to perform load tests; so in that case does load test tool from the same firm supports your application? This would bring you cost advantage in terms of licensing, license management or central dash board tool from reporting perspective.
• Does it support different Operating Systems under which your application runs?
• Does it use a proprietary language or generic language like Java or .NET Framework or the language that has been used by developers of the project? If you want to stretch automation to perform unit testing then it would be required that development engineers be able to tweak code here & there to get successful results
• Does it schedule tests for you?
• What kind of reporting interface it provides? Does it help you build custom reports?

The questions asked below would help you in moving forward with the kind of tool –

• Do you want to build in-house? In-house tools are best when compared to vendor but source code management and resource dependency can kill the enterprise if any one listed before is mismanaged.
• Do you want to purchase? Nowadays, majority of the companies purchase tool from a provider and build their scripts. The only constraint is the language the tool uses for scripting, licensing renewability for regular updates & fixes on the tool.
• Do you want to go for Cloud/ SaaS model? Here, the vendor does everything for you, the tool, the scripting and its management along with dedicated server slots and so forth.

Open Source – A dominant factor

Automation framework should have flexibility to address different type of testing phases such as unit, functional, smoke & regression. For all web or
Test Automation – Are Enterprises ready to bite the bullet?

Browser based applications, **Watir and Selenium** came out as a blessing. Now, the open source market started coming out with different plug-ins that could integrate into a robust & sturdy framework. For mobile applications, tools like **Appium & Monkey Talk** are doing magic as they have support for various languages. They are not only applicable to native but for HTML5 apps as well.

The mobile world and the desktop world have started integrating and very soon they would unite to form one framework. Solutions have started coming into the market where one can utilize the selenium developed functional automation scripts for performance tests.

Apart from the distinctive advantages of usage of open source, it would indirectly help Managers as—

- The team would be lesser in size but would be giving out more productivity as different iterations or new features get integrated into the project or product.

- The ease of use of automation framework would now give birth to faster, shorter & more reliable cycles of SDLC be it Waterfall, Iterative or Agile.

**What next, now?**

Test Automation implementation will be successful when there is continuous innovation with open source tools. It is a definite advantage to have a vendor who also believes in this philosophy and can provide you with experts, robust frameworks and business focused testing approach. This way, organizations can have Test Automation in lesser cost, effort & time while having complete control over deliverables.

Let’s finish the journey for automation keeping in mind as where to start, how to move ahead and what to take along such that we reach the destination in minimum possible time achieving bigger & better returns.