

Open Source Software Test Automation Tools: A Competitive Necessity

¹ K. Saravanan & ² E. Poorna Chandra Prasad

^{1,2} Assistant Professor

Sree Vidyanikethan Institute of Management, India.

©Scholedge International Journal of Management & Development (ISSN 2394-3378), Vol.03, Issue 06 (2016) pg103-110.
Published by: Scholedge R&D Center [www.theSCHOLEDGE.org] [Email:editorial@thescholedge.org]

ABSTRACT

Software Testing is one of the critical activities in developing quality software. To enhance test efficiency and to improve repeatability of tests, several testing tools were developed and rolled out by proprietary commercial vendors like HP, IBM, etc., In the past decade, Proprietary software test automation tools dominated the automation market and were extensively adopted. But in recent past, Open Source test automation tools are acknowledged as reliable and are being adopted rapidly by IT companies. When companies started adopting Open source test automation tools, these tools were perceived as Competitive advantage to keep them abreast of technology developments, reduce cost, train testers, test in-house developed applications and so on. But today, we are witnessing Open Source test automation tools evolving as industry standard and have become a competitive necessity in IT Industry. This paper explores on how Open Source Test Automation tools have become a Competitive Necessity.

Keywords: Proprietary Software, Open Source Software, Testing Tools, I.T. Industry, Competitive Necessity, Software Testing, Test Automation.

INTRODUCTION

Software Testing is one of the key business processes of Software Development. It is a Quality control activity performed during the entire software development life cycle & also during software maintenance. In the initial phases of software development, developers will be performing reviews and unit testing to ensure quality product is delivered to the testing team. This is followed by Testing team performing System testing and coordinating Acceptance Testing.

Companies that are developing and delivering quality software to customers are spending several millions of dollars in performing testing activity effectively and efficiently. With Software testing gaining importance in the development of software, many companies that are specialized in providing testing services have emerged and are successful in their endeavour.

Manual Software Testing: An Introduction

Any software developed will undergo rigorous testing to ensure that it is bug free. Testing is performed by a human sitting in front of a computer carefully going through application screens, trying various usage and input combinations, comparing the results to the expected behaviour and recording their observations. Manual tests are repeated often during development cycles for source code changes and other situations like multiple operating environments and hardware configurations.

In general, Manual Testing effort is a systematic activity and follows a process. Test Process involves software test professionals analyzing and prioritizing the requirements to be tested, preparation of a test plan, setting the test environment, identification of test scenarios, preparation of test cases and test data, test execution and defect reporting. Periodic reports on the quality of software during test is generated and introspected to improve the quality of software.

Manual testing is the effective way of performing testing & is unavoidable in software development. Depending on the quality of manual tester, various test scenarios will get created and tested such that the software becomes robust. Though manual testing is effective, it becomes monotonous and boring when the same tests are to be executed for every version of the products and is highly inefficient to execute repeated tests/regression tests.

CONCEPTS OF RETESTING AND REGRESSION TESTING

a) Retesting

In the early stage of test life cycle, testing professionals execute each test case manually and check whether the actual outcome is matching with the predicted outcome. In case of any discrepancy, the same is logged as a defect. Once the defects are logged, developers fix the defects and a new build is given to testing team for retesting. Testers verify whether the bug is fixed and ensure it is closed. This activity is called retesting and will continue till the software is stable.

b) Regression Testing

Once the initial version of the software is stabilized, new versions comprising of new set of features will be developed. During this stage, testing team will be performing functional testing for the new features by preparing test scenarios, test cases and will perform Regression testing for the features of the previous versions. Regression testing is performed to ensure that the existing feature does not have any impact with addition of the new features.

TEST AUTOMATION: A BRIEF

Test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes. Automation tools help to speed up Unit testing, API Testing and GUI Testing.

Test Automation is highly economical to run regression tests. The clear beneficial aspect of Automation is its ability to run more tests with less time when compared to manual testing. Automation tools are employed to run tests which are very difficult to be run manually. Another major advantage of Automation is to run compatibility testing (i.e., running the same tests in different environment) which thereby increases productivity and ensures quality software is delivered to end users.

Some of the other benefits of automation tools include running more number of tests more number of times, performing tests which are very difficult to be tested manually, improving consistency of test results and also help effective utilization of manual testers in more creative and productive work and reusability of tests.

Though Automation improves test efficiency, some of the limitations of it include inability to find defects and does not help improve test effectiveness. Maintenance of the Automation scripts is another limitation. Typically skill set of personnel doing automation tests is different from Manual Testers, though now industry is equipping manual testers with the required automation skills.

Test Life Cycle

a) Testing by Development Team

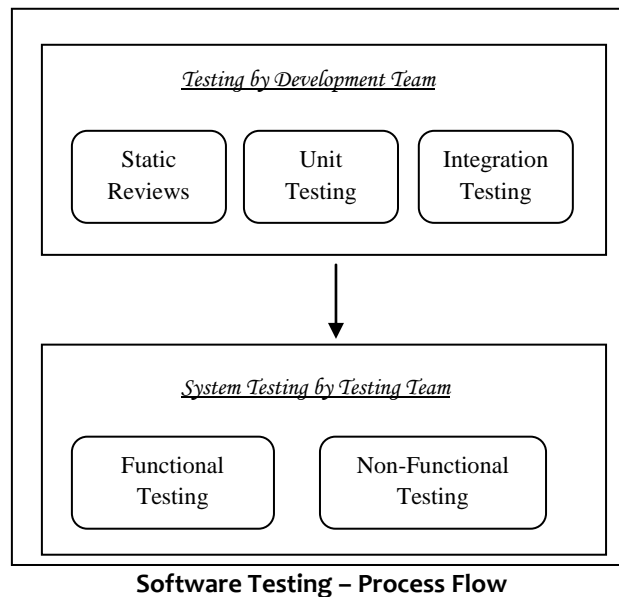
Systems Analysis and Design activities of the development team will deliver System Requirement Specifications (SRS) and Design Documents. Developers will use SRS and detailed design document as input to program their respective modules. Each of the modules will undergo Static reviews and Unit testing. The Static Reviews include Inspection, Peer review and Walkthroughs. This is followed by Unit Testing where the code of each module is dynamically tested.

All the unit tested modules are integrated and Integration Testing is performed to ensure all the interfaces are working properly. Development Team is responsible for performing Static Reviews, Unit Testing and Integration Testing. All the tests are White box in nature

which requires knowledge of internal working of the code/system. Many static code analysis and unit testing tools specific to programming language is available in market.

b) *Testing by Testing Team*

The Integration tested system is handed over to testing team for System Testing. During System testing, both functional testing and non-functional testing are performed. When the initial version of software is developed, Functional testing is performed to ensure that the software is stable and bug free. Once the system is functionally stable, non-functional testing activities like performance testing, Security testing, Availability testing, etc., are carried out to test the system's behaviour is as per expectation.



SOFTWARE TEST AUTOMATION TOOLS

Software Testing Tools are Computer Aided Software Engineering tools which support to speed up the Testing activity. Testing Tools are available to facilitate development team as well as Test Team. The tools which support developers help them in performing code reviews and unit testing. The tools supporting testing team assists in performing regression testing, performance testing, and test management and so on. The tools which support automating regression tests are Software Test Automation tools.

The software testing tool market can be broadly categorised in to two:

a) *Proprietary Software Test Automation Tools*

Proprietary Software Test Automation Tools are software's developed by companies which require licenses to be purchased to work with them. Some of the leaders in developing proprietary software include HP, IBM, Oracle, Microsoft, Borland and so on., The Proprietary Software Test Automation tool market is highly matured and companies provide various features for testing done by both development team as well as testing team. Companies offering proprietary software provide excellent support for the tool and the tools can be easily integrated with other test management, defect management tools using third party plug-in.

Gartner in their recent report published on Test Automation in 2015 has identified 12 vendors in the proprietary market. 4 vendors are positioned as Leaders, Oracle is positioned as Challenger, 4 vendors are positioned as Niche players and 3 vendors are positioned in the magic quadrant of Test Automation market. All the tools identified support testing mobile applications; web based applications and packaged applications. Gartner has also highlighted the strengths and

Cautions of each vendor. Some vendors are dropped as their products were not supporting all mobile, web and packaged applications.

The details of products offered by the vendors, their product name, key supremacy and key weakness of each tool is provided below. Though automation tools have many advantages, two major limitations of the proprietary software are: 1. Source code of the software is a copyright of the company offering these tools and cannot be modified/customized 2. Cost of licenses is very high for SME companies to adopt them.

List of Proprietary Software Test Automation Tools

Vendor	Product Name	Key Supremacy	Key Weakness
HP	Unified Functional Testing	Market Leader with strong presence in UI Automation	Pricing Model
IBM	Rational Functional Tester	Ease of Use and Maintenance	Slow in innovation
Tricentis	TOSCA	Model Based Test Automation Framework	Pricing Model
Worksoft	Certify	Preferred for Packaged Applications testing	Not reliable in Product Stability and Support
Oracle	Oracle Application Testing Suite	Good Customer Service and Support	Challenge to maintain Scripts & does not support mobile applications
SmartBear	Test Complete	Offers support for multiple skill levels	No Support for Packaged Applications
Ranorex	GUI Test Automation	Straight forward license model	Complex UI object recognition
Progress	Telerik Test Studio	Supports Microsoft Technology applications	No Support for Packaged Applications
Automation Anywhere	Testing Anywhere	Strong customer support and ease of use	Complex Licensing Administration
Borland	Micro Focus SilkTest	Role-based testing	Needs improvement to deliver unified test automation solution
TestPlant	eggPlant	Use of intelligent image recognition algorithms	Difficult to maintain scripts
Original Software	TestDrive	Code free approach	Missing Mobile Testing

b) Open Source Test Automation Tools

Open Source Software Test Automation tool market is an emerging market with companies offering testing tools free of cost and the Source code of the tools are available for modification/customization. Support is also available for these tools from Open Source development community. As like proprietary software's, Open Source testing tools also offer tools for both the development team and testing team.

Gartner in their report on Automation tools has also clearly stated that by 2020, the Open source automation tool - Selenium WebDriver will become the de-facto standard for test execution. With more and more agile development happening, the trend is to use open source tools for testing. Many open source testing tools are getting adopted and is gaining momentum.

The drivers for the growth of open source testing tool market include affordability, availability, user friendliness, reliability, smooth integration with other tools & no lock-in period.

The following provides details of some of the most popular open source software test automation tools in the market.

List of some of the Popular Open Source Software Test Automation Tools

Vendor	Product Name	Key Supremacy	Key Weakness
Apache	Selenium	Web Driver supporting multiple browsers & multiple programming languages	Supports only web based applications
Apache	Geb	Using powerful Selenium WebDriver APIs with simple Groovy language	Specific to 'Java based applications'
Apache	Windmill	Recorder tool that allows writing tests with learning a programming language	Supports only Web environment
Tyto Software	Sahi	Simple JavaScript based scripting tool with complete API which can run parallel scripts	User Interface is confusing
BSD	Watir (Web Application Testing In Ruby)	Supports Multiple OS & Supports playback of multiple scripts	Supports only ruby language
GitHub	Protractor	End to End framework for AngularJS Applications	Based on JavaScript
Cucumber Ltd.,	Cucumber	Based on Behavior Driven Development, easy to write scripts	Supports only Web environment
GitHub	SpecFlow	Employs consistent Domain-specific language and is user friendly	Specific to .Net based applications
Grant Street Group	Tellurium	Works adopting 'UI Module' approach	Supports only Web environment

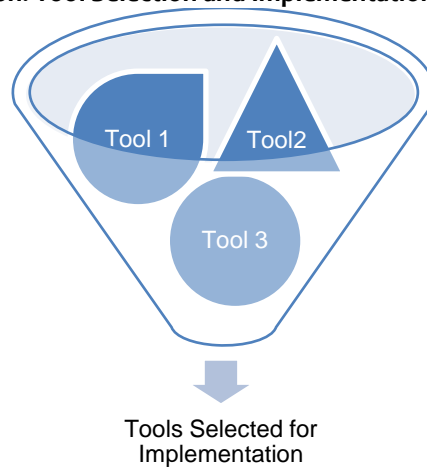
Adoption of Testing Tools

IT Companies designing and developing systems have started adopting testing tools extensively. One of the major reasons for adoption is that the testing tools facilitates them to improve productivity, provide accurate results, deliver timely reports, platform independent, reduces testing staffs such that they can be used in other productive activities and so on., Typically open source testing tools are adopted by companies who are moving from manual testing to automation or by companies who would like to supplement their existing commercial tools with open source tools.

In general, the tool selection and evaluation process starts with a business case providing justification on the need for test automation. This is followed by analysis of tools in the tool market, identifying potential tools, evaluating the tools based on specific parameters, ranking the tools and selecting the best tools for the company. Once the tools are selected, Implementation of the tool for a pilot study is done followed by companywide implementation. The Implementation of tool requires support of management commitment, support from internal users, training and buy-in from the users.

Earlier days, this process will happen once in 2-3 years. But now, with wide number of open source test automation tools entering the market from time to time & industry moving to use more than one tool, the tool selection, evaluation and implementation is repeated quite often.

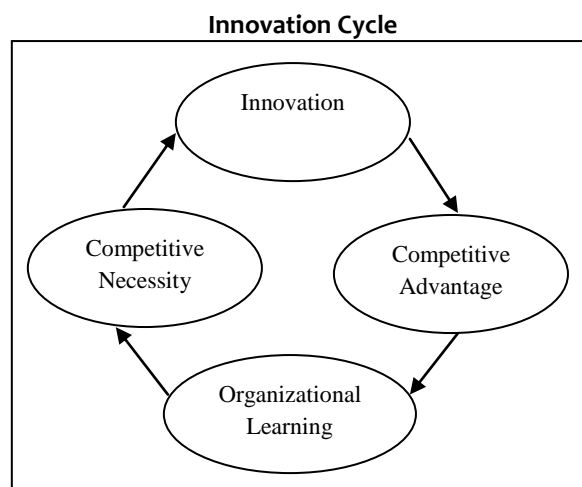
Tool Adoption: Tool Selection and Implementation Process



Companies who are relatively new to business, performing manual testing are moving towards adoption of open source testing tools. These companies are new to test tool adoption and require more careful planning before implementation. There are also companies who are already having some proprietary testing tools being adopted moving for open source test tool adoption. In such companies, open source tools are viewed as supplement or complement to their existing tool infrastructure. As already one implementation cycle is done, these companies will have learning experience in tool implementation and have probability of high success rate. Nevertheless, both categories of companies are adopting open source software testing tools.

a) *Competitive Necessity: A conceptual Framework*

Companies innovate continuously to deliver competitive products/services. Generally, companies who are market leaders in a particular market will invest more on R&D, create an environment which nurtures innovation and adaptation. The innovative products/services of these companies become one of the competitive advantages for them. Once innovative products/services are introduced into the market, the competitors in the market will learn the innovation and will start adopting them. This will make the competitive advantage as competitive necessity for the market leaders to start innovating again. This cycle continues which is really very much positive from growth prospects of the Industry.



b) Open Source Testing tools – A Competitive Necessity

In the past decade, Open Source Testing tool was looked as a competitive advantage by companies. In the earlier days, companies were adopting these tools to perform testing of in-house developed systems, using them to train testing professionals who are relatively novice test tool learners, testing prototypes and so on.,

Slowly, Open Source Testing tools started gaining acceptance after recognition of Open Source Initiative (OSI) community. This made many IT companies start training their testing professionals on open source testing tools. They initiated discussion with their clients on open source tools and some of them were successful in adopting them to test the software developed for their clients.

The scenario of few companies adopting Open Source testing tool has now shifted to Competitive Necessity. With high reliability and support from Open Source testing tools, clients started demanding companies to adopt open source testing tools which consecutively lower their total cost. Open Source testing tools have become mainstream tools in supporting testing activities and has become a competitive necessity in IT Industry

Gartner has mentioned that Selenium WebDriver, the open source testing tool from Apache will become the standard for functional test execution by 2020. It has also acknowledged that an ecosystem around Selenium has made Open source tools to have strong effect on the test automation market.

It also states that IT Companies designing and developing systems have started adopting more than one testing tool and Gartner predicts this trend will continue for next 10 years. Testing tools – both proprietary and open source are being adopted relatively in equal proposition.

Today, Open Source Test Automation Tools is laying the road to become Industry standards. It is setting the business environment for proprietary tools and there is huge impact of these tools on different facets of managing testing organization like recruitment, training, integration of existing commercial tools and so on., It is imperative for every organization to have open source testing tools implemented and is a competitive necessity.

CONCLUSION

Open Source testing tools have become competitive necessity to survive in the market and IT companies are gaining maximum advantage by adopting them. The adoption has become a necessity to such an extent that testing professionals recruited are expected to have expertise on open source testing tools. Even commercial tools vendors developing open source tools are providing integration features to accommodate open source testing tools. Gartner in their 2016 report on software test automation predicts Selenium, an open source testing tool will become the bench mark for 2020. It also forecasts that more than one testing tool will be adopted by companies. So, Open source testing tools have become a competitive necessity for today's survival and tomorrow's excellence.

References

1. The Art of Software Testing – Glenford J.Myers – John Wiley & Sons. Extracted from <http://www.computing.dcu.ie/~ray/teaching/CA358/TheArtofSoftwareTesting.pdf> (2015)
2. Software Test Automation: Mark Fewster and Dorothy Graham – Addison Wesley
3. Management Information Systems – James A O'Brien, George M Marakas, Ramesh Behl – Tata McGraw Hill
4. Management Information Systems – Kenneth C.Laudon and Jane P.Laudon - Pearson
5. Gartner Report on Software Test Automation Tools, 2016.

6. Dan Cornell, “Comparing proprietary tools and open source tools”, Tech Target Extracted from <http://searchsoftwarequality.techtarget.com/answer/Comparing-proprietary-tools-and-open-source-tools> (Aug 2013)
7. “Open source initiative” Extracted from <https://opensource.org/history> (Sep 2012)
8. Kolawa, Adam; Huizinga, Dorota (2007). Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press. p. 74.
9. Learning Test-Driven Development by Counting Lines; Bas Vodde & Lasse Koskela; IEEE Software Vol. 24, Issue 3, 2007
10. https://en.wikipedia.org/wiki/Open-source_software#Advantages_and_disadvantages

Authors' Biography

Mr. K. Saravanan has 13.5 years of overall experience with 3 years of experience in teaching MBA Students and 10 years of IT Corporate experience. He has worked for reputed companies like Infosys, Accenture, and Cordys R&D and has experience providing Software Testing services for companies belonging to Manufacturing, Banking, Telecom and Healthcare domain. Currently he is working as an Assistant Professor at Sree Vidyanikethan Institute of Management, Tirupati.

Mr. E. Poorna Chandra Prasad is specialized in Marketing and Systems Management with 9 years of teaching and 1 year of corporate experience. He is interested in customer studies and he has published 12 research papers in many reputed journals, presented papers in several management conferences and attended 12 FDPs. He has published a book on “CRM Practices in Indian Telecommunications.” He has recently submitted his PhD thesis at JNTUA, Anantapuramu.