

OPEN SOURCE AUTOMATED TESTING: AN INSIGHT INTO CURRENT TRENDS AND SCOPE FOR FURTHER RESEARCH

Mr.K.Saravanan¹, Mrs. E.Sreedevi²

¹Department of MBA,

Sree Vidyanikethan Institute of Management, Tirupati,
Andhra Pradesh

²Department of MCA,

Sree Vidyanikethan Institute of Management, Tirupati,
Andhra Pradesh

Abstract

Software Testing is one of the critical activities to deliver high quality software. Functional testing is a Software testing type focusing on testing the functions and features of a Software Application. Automated Testing is the usage of specialized software to perform functional testing to increase the speed of execution, improve repeatability of tests, enhance the reusability of Test scripts and generate accurate test reports which thereby increases test efficiency. Test Automation Systems are built employing proprietary testing tools and also Open Source testing tools. Organizations are adopting more than one tool to build their Test Automation systems. According to the recent reports from Gartner, Forrester and IDC on Software Test Automation, Open Source test automation tools are well-adopted by companies practicing Agile development and Continuous delivery.

With Industry moving towards Agile and DevOps development practices, there is an increasing trend in the adoption of Open Source Automation tools. The reasons for high adoption availability of these tools for lower cost, greater flexibility in customizing these tools, and the tools having features as like any commercial tool. Irrespective of the nature and number of testing tools adopted, IT companies adopt a systematic process to build the Test Automation System and the typical phases include Decision to Automate Test, Evaluate the Test tool market, Select the testing tool(s), Implement, Train testers, Utilize and perform maintenance. Though there are numerous researches carried out on each phase of building the Test Automation System, there is only little research available with regard to the utilization of the Test Automation Systems and the methods employed to evaluate utilization. This paper proposes the Scope for further research in analyzing the various impediments to utilization, the reasons for impediments and the measures that can be taken to minimize the impediments which in-turn helps improve the utilization of Test Automation Systems.

Keywords: Open Source Automation, Agile, Test Automation System

1. Introduction

Glenford Myers in his book titled “The Art of Software Testing” defines ‘Testing is the process of executing a program or system with the intent of finding errors’ [1]. Testing can be performed either manually or by automating tests. Test Automation refers to the use of computer programs to control the test execution and compare actual outcomes to predicted outcomes. It automates not only test case execution, but also test case generation and test result verification. [2]. Test Automation is preferred by companies to increase speed of test execution, increase test coverage and to reduce errors done by manual testing [6]. IT Companies planning to build their Software Test Automation Systems (STAS) will be adopting a systematic process having various stages similar to the process as shown in Fig 1.

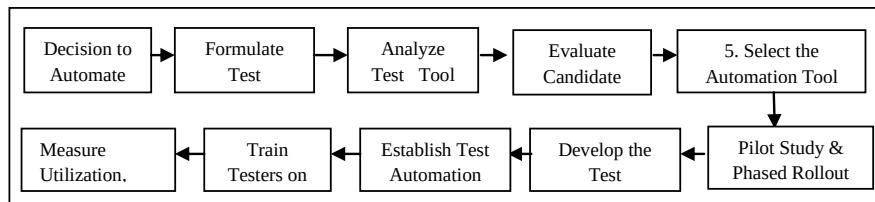


Fig 1: Process to build Software Test Automation System

Initially, the decision to automate is taken by Management by analyzing the potential benefits of automation and this is followed by formulation of Test Automation Team. The Automation Team include experts in test automation process, test tool market experts and leaders who have good people management skills. The next steps include the analysis of the test tool market which eventually ends in the evaluation of various tools and selecting the best tool which is suitable for the organizational requirement. Using the selected testing tool, a pilot project is carried out and once it is successful, the test automation system is built adopting the test framework appropriate for the organization. This is followed by training the testers on the test development and execution using the selected tool and the automation system is put to use. After considerable duration, the utilization of the Automation System is measured and the improvement in test efficiency and effectiveness is computed.

2. Review of Literature

The literature on Software Test Automation Systems is a highly explored area and many experts have contributed to the improvement of its knowledge base. It can be observed that most of the literature contributions are focused on exploring one of the stages of building the Test automation system and only very few contributions are done on the overall holistic process of building STAS. Particularly, maximum contributions are focused towards evaluating candidate tools and also discussing the Test Framework/Approach.

Contributions of Elfriede Dustin et al. [3]:

In their work, the authors propose a framework named “Automation Test Life-Cycle Methodology (ATLM)” for

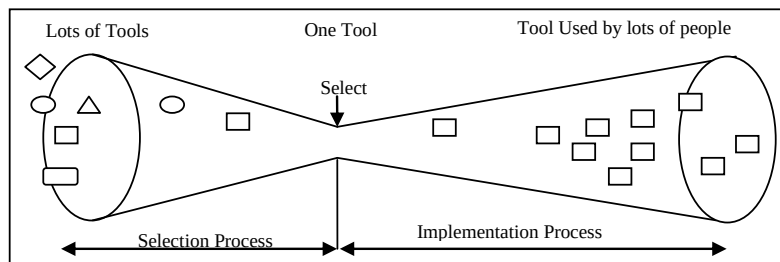
building a STAS. ATLM is a structured approach for the implementation of STAS and consists of six steps. The crucial step in the first stage – “Decision to automate test” is getting management support. In the next stage - “Test tool acquisition”, the parameters that are to be used to evaluate the test tools are to be decided and the tool is acquired. In the third stage – “Automated Testing Introduction Process”, the test automation team should rollout the tool and train the testers. During the fourth stage – “Test Planning, Design and Development”, the steps that are needed to setup test environment, techniques available to develop test cases and the procedure to build automation infrastructure are undertaken. Method to be adopted for evaluating test phases is given in Stage 5 – “Test Execution and Management”. Finally, in the last stage - “Test Program Assessment”, the methods used to calculate Return on Investment and take corrective actions are to be done.

Contributions of Linda G.Hayes [4]:

In this work, the author emphasizes the importance of Test framework in developing a Test Automation System, narrates the characteristics of a testing framework and gives the steps to be followed for establishing it. The author mentions a “Test framework” should have a test library management system having common functions, templates to develop test scripts and should have change management implemented for tracking purposes.

Contributions of Mark Fewster & Dorothy Graham [5]:

In their work, the authors have viewed building test automation system as a 2 step process - Tool Selection as the First step and Implementation as the Second step as shown in Fig 2. They suggest that the tool selection should start from business requirement and not from tool market. They suggest the methods to be used for evaluating tools and discuss the various Key roles involved in Implementation. Tool Implementation is viewed as a process of getting acceptance among the users and treated as a ‘selling’ task. Guidelines to be adopted for successful implementation are also provided



Contributions of ISTQB [6][7]:

International Software Testing Qualification Board, a non-profit organization involved in certifying Software Testers worldwide provide details on the various issues that a Test Manager should consider before building the automation system as part of the “Advance Test Manager – Certification Syllabus”. One of the major issues to be managed is “Tool Implementation and Management”. ISTQB cites that tool selection is a long term investment and suggests a process for tool selection. A four phase life cycle of the tool is proposed as Tool Acquisition, Support and Maintenance, Evolution and Retirement

Contributions on specific stage of building Test Automation System:

There are several literatures available on specific stage of building Test Automation System. Literature emphasizing the benefits of Automation [8], test team formation and the various team structures are available in plenty. There are number of literary work in the area of test tool evaluation and selection [9][10][11][12]. The significance of training on test automation tool and the need to establish a test infrastructure [17][21] is discussed by several laurels successful in this area of work.

3. Current Trends in Software Test Automation

Agile Development and DevOps is the current practice of Software Development. Manual Testing is accounted 20% of the overall testing effort and 80% of the testing activity is accounted for Automation. In the earlier decade, proprietary tools were dominating the test automation market and one solution was employed for functional test automation. Today, there is a shift seen in the market towards Open Source tools and most of the organizations are employing more than one solution for automating their testing effort.

The “2016 - Gartner Report on Test Automation” [26] emphasizes the strong demand for automation solutions for the next five years. In its Magic Quadrant for Test Automation, 9 commercial vendors including HPE, IBM, Tricentis, Worksoft, Microsoft, Ranorex, SmartBear, MicroFocus, and TestPlant are positioned as key proprietary vendors in this market. The report also cites that Open Source testing tools are adopted in combination with commercial tools and companies are moving away from the concept of “Monolithic tool” adoption and usage. It also cites that Selenium, Appium, Cucumber, Geb, Sahi are some of the most widely adopted Open Source tools.

“Forrester Wave Report on Functional Test Automation tool, 2016” [27] also gives a similar picture and highlights that continuous Delivery, continuous Testing and continuous Integration is the current trend adopted in Software development Industry. It also acknowledges that Open Source testing tools are expanding in functional test automation market and points out Selenium and Cucumber as the widely adopted Open Source testing tools. It identifies 11 commercial vendors having HPE, IBM, Tricentis and Parasoft in its “Leader” Wave and the others as Strong Performers and Contenders.

“IDC’s - Worldwide Automated Software Quality Forecast, 2016–2020 Report” [28] emphasis the impact of Open Source testing tools on Automation tool market and highlights ‘Selenium and Cucumber’ as key open source solutions. It also highlights the integration of vendors and cites several consolidations happening in this market. It expects more consolidation and reports that smaller vendors will continue to be pivotal in bringing more innovations in this space. In Nutshell, there is a strong demand for automation technologies and open source testing tools will complement the proprietary tools in building effective test automation systems.

4. Proposed Work

Though there is lot of literature on how test automation systems are to be built, there is only a little work done on ascertaining the utilization of test automation systems and exploring the impediments caused for effective utilization of them.

Kasurinen, J et al [29] and Martin et al [30] performed research in this area and their findings are: Complicated tools causes problems, Tool usability and Configuration of testing systems. Their studies were further explored by Kristian Wiklund et al [31][32]. In their study, the impediments for utilization of testing systems are categorically investigated

under two headings. 1. Problems in using the automation system, 2. Help requirements when users seek for help to know how to do or use the system. The major 'problem related impediments' are related to centrally managed IT Environment & configuring test environment. With respect to 'help requests', the major impediments are related to test script development. They further report that few of the problems are unknown and conclude saying that 'same mistakes are repeated; the same problems are encountered over and over again'.

Their recommendation is to perform a research on preventing the commonly occurring impediments by implementing preventive approaches. They correlate prevention of commonly occurring impediments in Test Automation System to "Software Development". In "Software development projects", during the initial phases, preventive procedures to control and reduce defects are implemented [33]. Similar to the defect prevention approaches, during the initial phases of an automation project; preventive mechanisms are to be put in place to control the recurring impediments and improve the utilization of test automation systems.

Our Research Proposal is twofold: First, we propose to conduct a research to analyse the various methods used in quantifying the utilization of Test Automation Systems. A Comparative analysis of these methods highlighting their pros and cons will help employ these methods effectively. The Second proposal is to conduct research and arrive at the average utilization rate of Test Automation Systems by approaching companies who have already established TAS. Once the average utilization rate is quantified, the major reasons for non-utilization of the Automation System have to be identified. Solutions to remove the recurring impediments should be identified. By proactively implementing these solutions during the initial phase of building test automation system will help improve Utilization of the Test Automation System. Effective Utilization in turn will improve Managerial Performance.

5. CONCLUSION

Software Test Automation is the only solution for companies focusing to deliver high quality products, improve faster time to market and innovate continuously. Agile Development and DevOps practices demand more sophisticated automation tools and there are number of commercial vendors competing in this market. Open Source testing tools are also grabbing a major share in this market and most of the organizations are expected to have more than one testing tool for building their automation systems in the near future.

Once the Automation System is built, the Test Automation Management team should focus on the methods to measure the utilization of these systems, identify impediments for utilization and take measures to minimize these impediments. A Research on this aspect is proposed whose outcome is to identify the root cause of the recurring impediments which results in identifying solutions to minimize impediments. By formulating a proactive mechanism to prevent such impediments during the early stage of building test automation system will help to improve the utilization which in-turn results in greater efficiency.

References

- [1] Glenford J. Myers, Tom Badgett, Corey Sandler, "The Art of Software Testing", *John Wiley & Sons, Inc.*, 1979
- [2] Shilpa Mahajan, "Review of automation software testing (ast) and implementation of rft automation testing", *International Journal of Computational Intelligence & Telecommunication Systems*, 2011
- [3] Elfriede Dustin et al, "Automated Software Testing – Introduction, Performance and Management", *Addison Wesley*, 2008
- [4] Linda G. Hayes, "The Automated Testing Handbook", *Software Testing Institute*, 1995
- [5] Mark Fewster and Dorothy Graham, Software Test Automation: Effective use of test execution tools. *Harlow: Addison-Wesley*, 1999.
- [6] Certified Tester Advanced Level Syllabus Test Manager Version 2012, *ISTQB*, 2012
- [7] ISTQB® Worldwide Software Testing Practices Report, 2015 - 2016
- [8] Dudekula Mohammad Rafi, Katam Reddy Kiran Moses, Kai Petersen, "Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey
- [9] Monika Sharma, Rigzin Angmo, "Web based Automation Testing and Tools", *International Journal of Computer Science and Information Technologies*, 2014
- [10] Neha Bhateja, "A Study on various Software Test Automation Tools", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 5, Issue 6, June 2015
- [11] Vishawjyoti, Sachin Sharma, "Study and Analysis of Automation Testing Tools", *Journal of Global Research in Computer Science*, Volume 3, No. 12, December 2012.

- [12] Neha Dubey, Mrs. Savita Shiwani, "Studying and Comparing Automated Testing Tools; Ranorex and TestComplete", *International Journal Of Engineering And Computer Science*, 2014
- [13] Sergey Uspenskiy, "A survey and classification of software testing tools", *lappeenranta university of technology*, 2010
- [14] S.Rajeevan, B.Sathiyan, "Comparitive Study of Automated Testing Tools: Selenium and Quick Test Professional," *International Journal Of Engineering And Computer Science*,. vol. 20, pp. 569–571, Nov. 1999.
- [15] Inderjeet Singh and Bindia Tarika, "Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir", *International Journal of Information & Computation Technology*, 2014
- [16] Manjit Kaur, Raj Kumari, "Comparative Study of Automated Testing Tools: TestComplete and QuickTest Pro", *International Journal of Computer Applications*, 2011
- [17] Harsh Bajaj, "Choosing the right automation tool and framework is critical to project success", *Infosys Limited*, 2015
- [18] Tarik Sheth, Dr.Santosh Kumar Singh., "Software Test Automation – Approach on evaluating test automation tools", *International Journal of Scientific and Research Publications*, Volume 5, Issue 8, August 2015
- [19] Olanrewaju m. Abiodun, "Open Source Software Testing tools Evaluation Model", *Kemi-tornio university of applied sciences Technology*, 2011
- [20] Vishal Sangave, Vaishali Nandedkar, "A review on Automating Test Automation", *International Journal of Advanced Research in Computer Science and Manaegment Studies*, December 2014
- [21] Douglas Hoffman, "Test Automation Architectures: Planning for Test Automation", *Software Quality Methods*, 1999
- [22] Milad Hanna, Nahla El-Haggar, Mostafa Sami, "A Review of Scripting Techniques Used in Automated Software Testing", *International Journal of Advanced Computer Science and Applications*, 2014
- [23] P.Suresh Kumar, Dr.N.Samba Siva Reddy, "Automation of Software Testing in Agile Development", *Global Journal for Research Analysis*, Volume-3, Issue-7, July-2014
- [24] Lars-Ola Dammlet al, "Introducing Software Test Automation and Test-Driven Development:: An Experience Report"
- [25] Valery Safronau and Vitalina Turlo, "Dealing with Challenges of Automating Test Execution", *The Third International Conference on Advances in System Testing and Validation Lifecycle*, 2011.

- [26] Magic Quadrant for Software Test Automation, 15 November 2016
- [27] The Forrester Wave™: Modern Application Functional Test Automation Tools, Q4 2016
- [28] Worldwide Automated Software Quality Forecast, 2016–2020
- [29] Kasurinen, J et al, “Analysis of Problems in Testing Practices”, *Software Engineering*
- [30] D. Martin, J. Rooksby, M. Rouncefield, and I. Sommerville, “‘Good’ Organisational Reasons for ‘Bad’ Software Testing: An Ethnographic Study of Testing in a Small Software Company”, *29th International Conference on Software Engineering*, 2007
- [31] Kristian Wiklund, Daniel Sundmark, Sigrid Eldh, Kristina Lundqvist, “Impediments for Automated Testing - An Empirical Analysis of a User Support Discussion Board”, *Seventh IEEE International Conference on Software Testing, Verification and Validation (ICST)*, 2014
- [32] Kristian Wiklund, “Impediments for Automated Software Test Execution”, *Mälardalen University*, 2015
- [33] Nasir Jamal, 2M Zulqarnain, 3M Waqas Boota, 4Saima Khan, 5S M Abrar Akber, “Role of Defect Prevention Techniques Vs Defect Detection to Improve Software Quality: Critical Analysis Summary of Defect Preventive Approaches”, *International Journal of Novel Research in Computer Science and Software Engineering*, 2015

