WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Test Automation Framework to Produce Reliable and High-Quality Test Automated Systems in a Business Scenario

Rashmi Popli

Associate Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, India, rashmimukhija@jcboseust.ac.in

ABSTRACT

Test Automation has become the need of the hour for the software development industry and the reasons behind, are the benefits it provides - confidence on build quality, lower turnaround time, quick releases, remove repetitive job from tester's scope are few of them. However, the test automation solution should be enough robust to provide adequate confidence to the project team about product quality and prone to failure which causes high maintenance. Some approaches are being devised by the automation experts and implemented in the automation solutions so that it not only diagnoses the failure but starts resolving the issue without any manual intervention and reduces the maintenance cost. These approaches are a sneak peek for testers into what next, they can expect in test automation industry as these not only confront the common failure causes but act as insurance for automation tests when they fail. In this research work an adaptive autonomous Test Automation Framework for a Business Scenario is proposed that reduces manual maintenance effort of automation framework.

KEYWORDS: Test Automation Framework, Self-healing, Return on Investment (ROI), Application Under Test (AUT), Object Repository (OR), Test Scripts.

1. Introduction

Testing is the process of checking if the developed product fulfils the user requirements, has minimum or zero defects and performs its goal [1]. This is done by comparing the actual behavior of product with the expected behavior. Testing is needed to ensure that the final product meets its purpose, has all the functionalities working and has no critical defects [2]. Before releasing the final product, it is required to perform a thorough testing of all the features to avoid bad customer experience and protect the reputation of the software product [3].

One of the major testing type, functional testing is performed by identifying test scenarios of each feature of the product and then converting these scenarios into test case generation which contain pre-requisite, post-requisite, test data, action steps, expected results and actual results[19]. These actions are then executed manually by functional testers on AUT and actual results are compared with the expected behavior of the product. Based on the results, test cases are marked passed or failed and bugs are reported in the defect management system [4,15].

Test automation is the use of other software/tool/frameworks to create the automated test scripts corresponding to each functional manual test case and manage &control the execution of these [38,10]. These automated test scripts when executed, compare the actual outcomes with predicted outcomes and present the outcome in a test execution report [5, 6]. Test automation follows the approach "Write Once, Run Anywhere". Once written, these test scripts can be executed infinite number of times on multiple platforms, with numerous sets of data and on various environments even in web applications [21]. Test automation scripts program all repetitive and necessary tasks in a formalized testing process already in place or perform additional testing that would be difficult to do manually [22]. Test automation is critical when software is produced in short cycles and immediate feedback (Agile projects) on the business risks associated with a software release candidate is required [8,9].

Also, it increases the overall software efficiency and ensures best quality product. There are specific tools that not only support in designing the automated test scripts effectively but executing them and helping in comparing actual and expected results[23,37]. One of the biggest business perks of automated testing is that, once implemented it requires minimal effort to maintain and provide maximum accuracy in test results.

Test automation framework, if not implemented appropriately itself can be a big challenge for automation. The test automation solution must not only focus on today's application context but instead should also see application context 2-3 years down the line in terms of tools, technology etc[11]. All software applications projects also include test automation scripts (either unit testing scripts or UI testing scripts) which are written to test different feature and business processes. If these test scripts fail to execute properly then it could trigger test automation to fail, making it a big challenge in automation testing [20].

It has also been observed that there is one big bang effort spent on test automation and then as development team keeps on adding new features, QA fails to keep track of these changes, scripts failures and to maintain/adapt the scripts [13, 14]. This automation backlog keeps on mounting over time, which leads to test automation failure [12]. Test automation suite requires regular maintenance, testers write new methods but forget to remove the old ones which results in dead code lying in the framework which is never used. Choosing the right automation tool is also a critical in project success [18].

The goal of this research paper is to analyze the importance of test automation and the failure reasons of a test automation framework and prioritize them on the basis of their occurrence.

2. Literature Review

Test automation will be very difficult to successfully implement if the testing is not planned in a structured way or is considered optional, or if while doing testing there is a general unconstructiveness and lack of enthusiasm to implement processes. Rigid deadlines and a high schedule pressure also increase the risk for deviations from processes and automation failure that may cause problems later. If the automated test is perceived as an impediment by its users, there is a risk that the testing is skipped or that the automation is abandoned in favor of manual work. Table 1 presents the critical analysis of literature work. Automated Testing is carried out to scale back testing time drastically - A typical automated test suite will run in much less than 24 hours. For a sophisticated product, manual testing may require dozens of workforce months to perform the similar testing.

Table 1: Critical Analysis of Literature Work

S.No	Year	Author	Title	Premises	Critical Analysis
1.	2008	B.Haugset ,G.K Hanssen	Automated Acceptance Testing: a Literature Review and an Industrial Case Study[26]	The paper aims to do literature Survey on automated software testing and use the same in an industrial case study.	Automation testing needs the high-level of technical skills. Maintaining automated test scripts is still a cumbersome task if it contains hard coded data for testing also.
2.	2010	Xiaojun Wu, Jinhua Sun	The Study on an Intelligent General -Purpose Automated Software Testing Suite[27]	The work presents an intelligent General purpose software test automation suite which can improve the testing efficiency by decreasing the labour cost and effort.	The proposed general-purpose test automation Tool Namely Harness and Log scanner does not provide a complete solution to all testing needs. The solution has to be implemented in IT projects to analyze its practical aspects.
3.	2013	Zara Shaheen, Abdul Rauf, Bushra Hameed	Automated Software Testing: An Insight into Local Industry[28]	The paper aims to depict the existence of Automated software testing in the local Industry through a survey.	The survey was conducted on a very small sample size. There is a large gap between the literature and practical aspects of Automated software testing in the Industry. More factors need to be considered while analyzing the applicability of automation like large sample size, experience, industry scale and product size.
4.	2013	David Barrett	Automating Testing - Saving Time and Money[29]	This paper explores an experimental approach to upgrade testing and task automation via web browser automation using Selenium.	Selenium has been quite popular these days for web automation testing. Maintenance of test scripts for large software is still a challenge.
5.	2014	Mariani,Leonardo &Pezzè,Mauro&RiganelliOlivieo	Automatic testing of GUI-based applications [30]	The paper presents a technique for automatic test case generation known as Auto BlackTest for web applications. The system level testing can detect several current as well as previous unknown errors by interacting with only GUI.	Auto BlackTest, a black box testing technique can generate test cases more effectively by providing coverage and detecting more bugs. The work can be further extended in other domains like mobile devices and Artificial Intelligence.

S.No	Year	Author	Title	Premises	Critical Analysis
6.	2015	Gordon Fraser, Matt Staats, McMinn, Andrea Arcuri, and Frank Padberg	Does Automated Unit Test Generation Really Help Software Testers? A Controlled Empirical Study[31]	The study tries to investigate the Significance of automated software testing in a practical environment. Despite the fact that test automation improves the productivity and efficiency, its limited adoption in organizations raises the question on its practical value.	The result shows that only by increasing test Coverage will not improve the efficiency and quality of the product. The study marks the need of highly empirical studies in the area of automated test generation like complexity of test cases, maintenance of automated test generation, partial test oracles.
7.	2016	Divya Kumar, K. K. Mishra	The Impacts of Test Automation on Software's Cost, Quality and Time to Market[7]	The work tries to ascertain some of the critical factors related to test automation and cost return of from automation.	A few case studies prove that the automation costs are recovered in long runs. But various aspects of software still can't be automated like user interface testing. Not all the bugs are found.
8.	2017	Raulamo-Jurvanen, Päivi&Mäntylä, Mika &Garousi, Vahid	Choosing the Right Test Automation Tool: a Grey Literature Review of Practitioner Sources[32]	The paper shows a grey literature review on how right test automation tool is chosen to best suite the business.	The paper highlights the practical implications of choosing a right test automation tool.
9.	2017	Claus Klammer and Rudolf Ramler	A Journey from Manual Testing to Automated Test Generation in an Industry Project[24]	The article discusses the transition from manual to automation in GUI test case generation in Industries.	Automated test generation gives a new research dimension in terms of measuring the efforts required to analysis and maintain these automated test scripts and test oracles.
10.	2017	Rashad Khalid	Towards an Automated Tool for Software Testing and Analysis[33]	The author proposes a all in one Test automation tool which can perform of different types of testing unit testing, exception-based testing, static analysis, assertion based testing and dead code testing, based on C programming language.	Since the tool tentatively performs most of the Testing alone, the complexity, effort and time factors are need to be analyzed.

3. Proposed Working of Test Automation Framework while Testing a Business Scenario

A "Test Automation Framework" is scaffolding that is laid to provide an execution environment for the automation test scripts. The framework provides the user with various benefits that help them to develop, execute and report the automation test scripts efficiently in Agile Software Environment [16,17]. It is more like a system that has created specifically to automate the tests. In other words, it is a set of guidelines like coding standards, test-data handling, object repository treatment etc. The advantages of test automation framework can be in different forms like the ease of scripting, scalability, modularity, understandability, process definition, re-usability, maintenance etc. Moreover, the need of a single and standard Test Automation Framework arises when a bunch of developers are working on the different modules of the same application and when they want to avoid situations where each of the developers implements his/her own approach towards automation.

3.1 Steps Followed by Test Automation Framework while Testing a Business Scenario

To explain the working of a test automation framework, consider the following example of a login page of the application (Figure 1) which has three objects i.e. a textbox to enter the username, another textbox for password and OK button to click to continue.

```
<html>
<body>
<form id="login">
<input name="username" type="text"/>
<input name="password" type="password"/>
<input name="OK" type="ok" value="Continue!"/>
</form>
</body>
```

Figure 1: Example Login Page of Application

3.2 Steps to Run Test Automation Framework in a seamless way

Step 1: Automation framework invokes the driver which takes the command for a particular test method that need to be tested.

Step 2: Driver opens the browser and enters the URL of the application which it takes from the configuration file.

Step 3: Once application is opened in the browser, driver reads the test method and looks for the first object to interact with. In this example, this first is object “Username” text box.

Step 4: It reads the property of this object from OR, and try to find the object in the application while parsing all elements of HTML DOM and once, it finds the object with its locator type which is “name” and locator value with “username”, it enters the parameters(test data) username in the text box. This test data is being read from a test data file or from a database.

Step 5: Then it follows the same procedure for finding the “password” text box and “Ok” button.

Step 6: After clicking the OK button on the application, application navigates to application landing page and on this page, tester has to verify that user is successfully able to login into the application. So, driver looks for a unique object on this page and validate a selective property on this page.

Step 7: Once validated, this business scenario is successfully passed, and driver closes the browser after signing out the application.

Step 8: While following this procedure, all test methods are executed and at the end a report is generated to verify all results and if few test methods get failed, its results are analysed to find the failure reasons.

3.2 Object Repository in a Business Scenario

Every UI test automation tool/framework identifies the elements on the UI application with the help of element locators (along with their values) and performs action on it. Each of these object/elements can have multiple locators like id, name, CSS, class, xpath, link text which helps the framework in identification. Test automation frameworks use one of these locators and its value to identify an element and perform action as per business scenarios [31]. As soon as the object is identified, test scripts can perform any action on it.

Typical examples of these locators are: Identifier, Id, Name, Link, DOM, XPath, CSS,UI-element

Valid locators for the objects in the example considered in previous sections are:

- For username textbox, locator is name and its value is “username”
- For password textbox, locator is name and its value is “password”
- For Ok button, locator is again name and its value is “OK”

All these locators and their values are saved at a common place (within page object class file or in XML, json file) which is normally called as object repository (OR) in test automation. This OR canbe a separate component of a test automation framework and have the most important role to play in overall scenario. Without OR, no automation framework can work. A typical automation framework saves one locator and its value of an object in the OR. Figure 2 shows an Object Repository of the example of Business Scenario. A typical OR maintained in a Json file looks like this where application

objects are maintained module wise. Every module has its own separate OR file to maintain. “NameOfElement” is the unique name which is being used in the framework to identify this object, and “ElementPropoerty” contains its locator type as “Type” and value as “Value”.

```
{
    "Modules": {
        "Element": [
            {
                "NameOfElement": "txtusername",
                "ElementProperty": {
                    "Type": "name",
                    "Value": "username"
                }
            },
            {
                "NameOfElement": "txtpassword",
                "ElementProperty": {
                    "Type": "name",
                    "Value": "password"
                }
            },
            {
                "NameOfElement": "btnOK",
                "ElementProperty": {
                    "Type": "name",
                    "Value": "OK"
                }
            },
        ],
        "_xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
        "_xsi:type": "Modules"
    }
}
```


Figure 2: Snapshot of an Object Repository (OR)

4. Conclusion

The proposed adaptive and autonomous Test Automation Framework demonstrates a significant step forward in reducing manual intervention and maintenance efforts in test automation. By intelligently diagnosing failures and initiating corrective actions, the framework enhances confidence in build quality, accelerates release cycles, and minimizes repetitive tester effort. This research lays the foundation for future advancements in self-healing and self-maintaining automation systems, ultimately enabling more resilient, scalable, and cost-efficient testing solutions for the software development industry.

REFERENCES

[1] G.J. Myers, C. Sandler, T. Badgett, The Art of Software Testing, 3rd Edition, 2015.

[2] P. Jalote, An Integrated Approach to Software Engineering, Springer Science & Business Media,2012.

[3]P. McMinn, Search-based Software Test Data Generation: A Survey, Software Testing, Verification and 1256 Reliability, 14(2):105–156, 2004.

[4] E. Kit, S. Finzi, Software Testing in the Real World: Improving the Process, ACM Press/Addison- Wesley Publishing Co., New York, NY, USA, 1995.

[5] Stobie, K. Too Much Automation or Not Enough? When to Automate Testing? Pacific Northwest Software Quality Conference, 2009.

[6] Kasurinen, J., Taipale, O. & Smolander, K. Software Test Automation in Practice: Empirical Observations,Advances in Software Engineering 2010.

[7] Divya Kumar, K. K. Mishra , The Impacts of Test Automation on Software’s Cost, Quality and Time to Market,7th International Conference on Communication, Computing and Virtualization 2016, Procedia Computer Science 79, 2016,pg 8 – 15.

[8]R. M. Sharma, Quantitative Analysis of Automation and Manual Testing, International Journal of Engineering and Innovative Technology (IJEIT) Volume 4, Issue 1, July 2014.

[9] Katam Reddy, Kiran Moses, Kai Peterson, Mika V.Mantyla, Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey, Proceedings of the 7th International Workshop on Automation of Software Test, June 2012 Pages 36–42.

[10] E. Alegroth, R. Feldt, and H. H. Olsson, Transitioning Manual System Test Suites to Automated Testing: An Industrial Case Study, IEEE Sixth International Conference on Software Testing,2013.

[11]S. Eldh, K. Andersson, A. Ermedahl, and K. Wiklund, Towards a Test Automation Improvement Model(taim). In 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops, pages 337–342, March 2014.

[12] Alex Groce,Iftekhar Ahmed, Carlos Jensen, Paul E. McKenney, and Josie Holmes. How verified (or tested) is my code? falsification driven verification and testing, Automated Software Engineering, 25(4):917–960,Dec 2018.

[13] https://blog.thedigitalgroup.com/importance-of-automation-in-software-testing,2019.

[14] https://bitbar.com/blog/increase-efficiency-and-productivity-with-test-automation,2019.

[15] K. Karhu, T. Repo, O. Taipale, and K. Smolander, Empirical Observations on Software Testing Automation, International Conference on Software testing, Verification and Validation, IEEE, Denver, Colorado,USA,2009,pp.201–209.

[16] E.Collins, A.Dias, Neto,VFD Jr.Lucena, Strategies for Agile Software Testing Automation: An Industrial Experience,2012 IEEE 36thAnnual Computer Software and Applications Conference Workshops,Izmir,Turkey,2012,pp.440–445.

[17] K.Wiklund, D.Sundmark, S.Eldh, and K.Lundqvist, Impediments in Agile Software Development:An Empirical Investigation, Product Focused Software Process Improvement (PROFES),Springer Verlag, Paphos,Cyprus,2013,pp.35–49.

[18]Harsh Bajaj, Infosys, Choosing the Right Automation Tool and Framework is critical in project success-White paper, 2018.

[19] T. Kanij, R. Merkel, and J. Grundy, A Preliminary Survey of Factors Affecting Software Testers, 23rd Australian Software Engineering Conference, Sydney,Australia,2014 ,pp.180–189.

[20] E. Alegroth, R. Feldt, and H. H. Olsson, Transitioning Manual System Test Suites to Automated Testing: An Industrial Case Study, IEEE Sixth International Conference on Software Testing, Verification and Validation IEEE;Luxembourg,Luxembourg,2013,pp.56–65.

[21] Shantha Jayalal, Thesis for: Masters in Information Systems Management, A Test Automation Framework For Web Applications Testing Using a Model Based Approach: A Study Based on Sri Lankan IT Industry, June 2015.

[22] V. Sangave and V. Nandedkar, A Review on Automating Test Automation, International Journal of Advance Research in Computer Science and Management Studies, vol. 2, no. 12, 2014.

[23]V. Sangave and V. Nandedkar, Generic Test Automation, International Journal of Science and Research (IJSR), vol. 4, no. 7, 2015.

[24]Klammer Claus, Ramler R, A Journey from Manual testing to Automated Test Generation in an Industry project, IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 25-29 July 2017.

[25]Kim Kunka et.al, Developing A Test Automation Framework for Agile Development and Testing, International Conference on Agile Processes and Extreme Programming in Software Engineering,2017.

[26]B. Haugset and G. K. Hanssen, Automated Acceptance Testing: A Literature Review and an Industrial Case Study,Agile 2008 Conference, Toronto, ON, pp. 27-38, 2008.

[27]XiaojunWu ,Jinhua Sun ,The Study on an Intelligent General-Purpose Automated Software Testing Suite, International Conference on Intelligent Computation Technology and Automation,2010.

[28] Zara Shaheen, Abdul Rauf, Bushra Hameed, Automated Software Testing: An insight into local industry, 2013

[29]David Barrett , Automating Testing - Saving Time and Money,2013

[30]Mariani, Leonardo & Pezzè, Mauro & Riganelli, Oliviero & Santoro, Mauro, Automatic Testing of GUI-based applications, Software Testing, Verification and Reliability, 2014 ,24. 10.1002/stvr.1538.

[31]Gordon Fraser, Matt Staats, Phil McMinn, Andrea Arcuri, and Frank Padberg, Does Automated Unit Test Generation Really Help Software Testers? A Controlled Empirical Study, ACM Transactions Software. Engineering Methodology, Article 23, August 2015

[32]Raulamo-Jurvanen, Päivi & Mäntylä, Mika &Garousi, Vahid, Choosing the Right Test Automation Tool: a Grey Literature Review of Practitioner Sources, 21-30, 2017

[33]R. Khalid, Towards an Automated Tool for Software Testing and Analysis, 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, pp. 461-465, 2017