



ISSN 2047-3338

An Agent Based Assisting System for Choosing the Right Testing Tool and Environment

Khadim Hussain¹ and Irum Mazhar²

^{1,2}CSE Department, University of Engineering and Technology, Lahore

Abstract— Software automation is a financial contribution having a high preliminary financial or monetary effect on software improvement. Usage of software automation may additionally undoubtedly affect the value, cost (e.g., by way of rushing up development iterations with the help of offering repeatable test contents and regression testing) and also improve the quality of software program or system, in big scale. Methods to test automation won't continually be suitable or successful. The exchange-off between manual testing and automated testing and the tools for use have to be recognized and justified. The venture to decide which tool to apply to maximize the benefits and improve the quality of a project is not a trivial one. Multiple software testing or software program test automation tools are available in the market, both business and open supply and particular and these tools have various goals in each development framework. The precise number of equipment is unknown and chances or resources to try out different tools are very restrained. Selecting the appropriate testing tool that can match the environment, control the cost and increase the chances of a project is necessary. Selecting and making use of the simplest and most effective tool for specific purposes in a particular context is vital for the achievement of enterprise. The goal of the research is to define a systematic, empirically validated agent based assisting system for selecting a tool for software test automation.

Index Terms— Software Testing, Testing Tools, Manual Testing and Automated Testing Tools

I. INTRODUCTION

FOR the development of programming framework testing and quality assurance account up to half of the financial plans [1]. Those activities are many times complex, take time, require experience and skilled personnel. Therefore, testing and quality assurance have been described as bottlenecks for the software development.

The Two main types of Software testing Black Box Testing and White Box [2]. Black Box Testing concerned with the specification of the System component under test which not require intensive knowledge about the internal structure of the system. White box strategy otherwise requires high experience of the internal system code for developing test suits suited the test cases.

Software testing is a procedure utilized for assessing characteristics or capabilities of the program and ensures that it meets the necessities [3]. Now-a-days testing becomes a very important activity in terms of exposure as well in terms of functional and non-functional testing. Performance, security and usability testing are very important types of testing.

If we consider hardware and software licenses, the testing is the too much expensive task for the user. Testing is very vast and hot topic in the field of research.

Studies claim that very few companies have realized the value of software test automation. However, recent commercial surveys display extended interest of organizations and companies in software automation. As an example, preferred adoption of taking a look at automation is alleged to be seventy- two percent in organizations or companies and the common percentage of test case automation is claimed to have extended within a year (considering that 2014 to 2015) from 28% to 45%.

Investments for take a look at automation apply to development environments as an entire, no longer simply to processes, practices, and equipment but also to the skills of employees. Lack of right tool for testing activities and absence of skilled sources were cited as limitations to test automation inside the industrial surveys. accordingly, it's miles now not a wonder that the findings of ISTQB file of 2015-2016 ranked test automation and test device and automation consultation as the main place of development possibilities (in trying out activities) and the service most wanted from external experts, respectively.

In the previous decades, a considerable measure of research exertion has been put into the improvement and investigation of programmed experiment age, programmed test prophets, and other (semi-) robotized testing methods as the theory and routine concerning programming testing end up being more built up, a more significant and more noteworthy automation of the testing technique is possible. Thusly, the computerization of different testing exercises is presently turning into an indispensable piece of mechanical practice. In light of and in the help of these energizing advancements, the fifth Workshop on the Automation of Software Test gives a distribution gathering that crosses over any barrier between the hypothesis and routine with regards to mechanized testing.

II. LITERATURE SURVEY

Anderson and Arilo [4] clarified significance and effect of basic variables of achievement in software test automation lifecycle. That clarifies the testing instruments and their choice is important to the point that it can prompt the disappointment of the venture it isn't chosen legitimately.

A. Fathima Musfira, MFM.Shakir and A.Fathima Munsifa [5] reason that over 52% of associations who have testing staffs and furthermore give preparing to analyzers are probably going to convey programming items with fewer deformities. It unmistakably exhibits that the product testing in programming improvement needs to end up more expert in Sri Lanka by naming testing staffs to the every last creating group and need to wind up more organized by giving basic testing preparing intermittently to the analyzers. Through the investigation of research result it is prescribed to give the accompanying aptitudes to the testing staffs, for example, consistent reasoning, tender loving care, thoroughly considering of the case, persistence, readiness to learn, flexibility, capacity to comprehend about creating style, fast comprehension about the client necessities and expectation energy of future upgrade of the framework and different prerequisites of client, space information of the item that they are trying, and better correspondence.

Neha Bhateja [6] displayed a relative investigation of different robotized testing instruments that utilized on various stages. Computerization testing devices help the analyzer to effortlessly robotize the entire testing process. Automation testing enhances the exactness and furthermore spare the season of the analyzer when contrasted with the manual testing. Mechanization testing additionally enhances the exactness and recovers the season of the analyzer and association's cash. It is best suitable in the earth where the necessities are over and over changing and a tremendous measure of relapse testing is required to be performed.

Päivi [7] clarifies the imperative choice factors that include in picking the correct testing instruments and condition for a particular situation. He depicts that test automation is a speculation having a high starting monetary effect on programming advancement. Use of test computerization may emphatically influence the expenses (e.g. by accelerating advancement cycles by giving repeatable tests and relapse testing) and the nature of programming or framework, in extensive scale. Ways to deal with test automation may not generally be proper or effective so it is essential to pick the correct apparatuses as indicated by the prerequisites and requirements of an undertaking.

It is accounted for in Security testing in the best IT system need [8] that testing is imperative in IT industry and it upgrades the life of the product. It gives a short outline of what number of associations are performing trying and what measures ought to be taken to advance its significance in the IT business as it is the basic piece of programming improvement lifecycle.

Raulamo-Jurvanen, Päivi, Kari Kakkonen, and Mika Mäntylä [9] broke down discoveries from information gathered using overviews and Web-scratching, to help Knowit Oy, a product testing meeting organization, during the time spent choosing the correct instruments for software testing

and test computerization. They directed two reviews (2013 and 2016) among (for the most part Finnish) software experts to secure criteria and a rundown of devices utilized for programming testing in the business. Considering all information sources Selenium was the most mainstream unadulterated apparatuses, while Robot Tool was the most referenced instruments (last review). As per the reviews, Jenkins and Sikuli have the most astounding increment in fame (or nature). Top alluded criteria for determination were ease of use, usefulness, practicality and accessible help for an apparatus. While Knowit thinks of it as best to use customary reviews, Web-scratching is viewed as practical help for such instruments.

Jan, Syed Roohullah et al [10] led a preparatory study on programming testing rehearses, The review concentrated on five noteworthy parts of programming testing, specifically testing strategies and procedures, robotized testing apparatuses, programming testing measurements, testing benchmarks, and programming testing preparing and instruction. In light of the study comes about, current practices in programming testing are accounted for, and also a few perceptions and suggestions for the eventual fate of programming testing for industry and the scholarly community.

Esparcia-Alcázar, Anna I., [11] assess the principal way to deal with hereditary programming (GP) for activity determination that includes developing IF-THEN-ELSE rules; they do investigations and contrast the outcomes and those got by arbitrary choice and furthermore by Q-taking in, a supportive learning system. Three applications are utilized as Software Under Test (SUT) in the analyses, two of which are restrictive work area applications and the other one an open source online application. The factual investigation is utilized to think about the three activity determination methods on the three SUTs; for this, various measurements are utilized that are legitimate even under the supposition that entrance to the source code isn't accessible and testing is just conceivable by means of the GUI. Indeed, even at this preparatory stage, the investigation demonstrates the capability of GP to develop activity determination instruments.

III. MOTIVATION

Testing plays a very important role in a project success. There is always a contrast between choosing the manual testing or automation testing. But automation testing excels in every situation. To choose automation testing tool it is very important to decide whether this tool is enough for a specific organization or not. More than 60% of organizations in IT industry is using automated testing tools [12] and invest millions on it but their end products are not as successful as they desire because of choosing wrong test tools in the wrong environment which make their project a complete disaster.

Testing is important in all phases whether it is development phase, security check or different iteration [13] but how to choose a perfect testing tool according to the scenario and requirements of an organization. Because choosing the wrong tool and investing in wrong resources can collapse the whole project and end product will not be too good for survival. Nobody likes an application with plenty of bugs. So for

designing a system that will suggest the perfect tool for each step according to the maturity, budget, and needs of an organization, following points should be taken into account.

- Assessment of the organization's maturity (e.g., readiness for change)
- Identification of the areas within the organization where tool support will help to improve testing processes
- Evaluation of tools against clear requirements and objective criteria.
- Proof-of-idea to see whether the item works as wanted and meets the prerequisites and purpose characterized by it.
- Evaluation of the vendor (training, support and other commercial aspects) or open-source network of support
- Identifying and arranging internal execution (including instructing and tutoring for those new to the utilization of the apparatus).

IV. PROBLEM STATEMENT

Problems arise in an automation system while testing when we are not using better or appropriate tools, and whenever we are using right tools but not using good environment. So 50 % of the project success relies on the right selection of the testing tools [14] and environment as wrong selection can lead to project failure. So there should be some kind of mechanism which can eliminate the manual hit and trial selection criteria and give exact testing tools to make a project successful.

These are the basic issues arises while running the automation testing in the agile environment:

- If the Automation tools we have selected is wrong
- If the time, at which we are starting the test, is wrong
- Not selecting the right tool for development
- The right test case selection
- When manual testing is not synchronized with automated testing

Test automation exercises ought to be viewed as a necessary piece of project quality affirmation, so if we want to improve productivity as well then these mistakes are highly avoided by the quality assurance teams. We are trying to develop some methodologies to implement the following goals with the help of this research

- To enhance the importance of testing in software development life cycle.
- To select the testing tools according to the need and maturity of a project or organization neither extra efficient nor under the requirement.
- To minimize the failure rate in software projects.

V. METHODOLOGY

In this research correlation of the proposed apparatus with comparable works examined. This correlation is performed in light of these components:

Supported Test Types: The more test writes are upheld by a test instrument, the all the more intense is that device.

Test Strategy: Generally there are three test strategies. Black-box testing forces the minimum requirements for the test to be performed. It doesn't require the source code or inward data about the SUT. The white-box procedure is on the opposite end. It requires that the source code of the systems be accessible. Grey box procedure lives in the center. It requires some data about the inward structure of the systems or its points of interest, for example, the database structure, yet not the source code. A casing work that is constrained to white-box methodology has less immaterialness than one that utilizes black-box technique, since it may not be conceivable [15] to solicit the suppliers from a system to influence the source to code of the systems open with a specific end goal to test the usefulness of open interface of the systems.

Information Sources: This thing demonstrates the sorts of data sources that are used by the structure to robotize the test procedure [16]. An apparatus that can utilize distinctive sources (e.g., UML models, session data, source code...) is obviously more compelling than an instrument that works just within the sight of a solitary source.

Human Manual Intervention: The less human mediation is required in the execution of a test procedure, the more successful is the fundamental apparatus.

Test Applicability Time: In which periods of SDLC the apparatus can be utilized? Is the apparatus relevant just when the systems are sent or it can be utilized amid the entire advancement cycle?

Tool Architecture: As said sometime recently, a circulating device is all the more intense and adaptable in the testing mechanized systems, since it adapts better to the qualities of these systems.

Target Type: What kind of systems can be tested utilizing the apparatus? Does it bolster web administrations or just customary robotized systems? Here, we focus on talking about the proposed structure concerning these variables.

Below is the system overview: The Fig. 1 shows the abstract system overview. First, we select the framework in which testing is to be applied then we select the tool and select the feasible environment. And then we get the list of tools recommendations.

A model of the proposed device was executed in Java. In this segment, a few issues about the execution of this model are quickly examined, since a far-reaching talk of the usage points of interest is beyond the extent of this paper. JADE2 is utilized as the underlying foundation of the device. It gives the fundamental administrations to building up multi-agent systems and covers up many low-level complexities and execution points of interest. Test Run Executer, Test Executer, Dashboard, Result Analyzer, Test Script Generator, Test Case Generator and Modeler agents have been created.

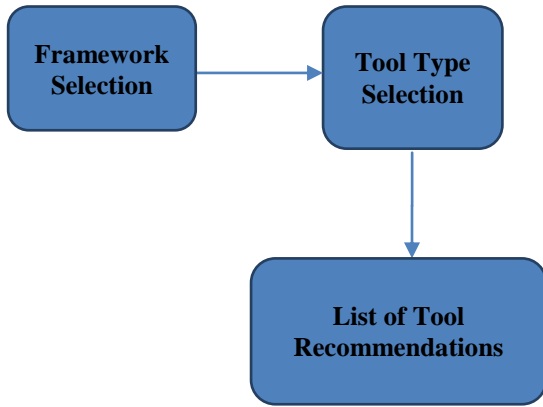


Fig. 1: Proposed System Overview

VI. EVALUATION AND RESULTS

We developed an agent-based assisting system for selection of automation tool. By using this we are able to find automation tool regarding our requirement, As you know multiple automation tools are available we analyzed them on basis of a framework, supported platform, licensed, open source, scripting language, ease of use/installation and cost. For example, if our system required much security then we have to find licensed automation tool for testing by entering some requirement in our assisting system it gives us the list of tools with description.

The Fig. 2 describes the User Interface of the assisting system user needs to log in through proper channel by signing up with an ID and password. If not registered then he must be signed up for the system.

The Fig. 3 describes the interface for user to select framework to find the existing tools.

The Fig. 4 shows the list of the entire tools name and their description of selected framework, e.g., we have selected framework of web applications and open source tool, as a result, the list of all assisting tools are listed.

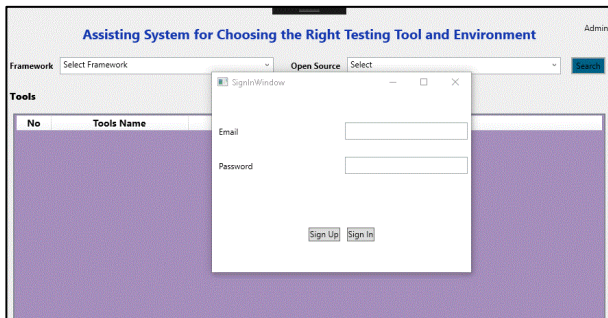


Fig. 2: User Log In for testing tool of framework

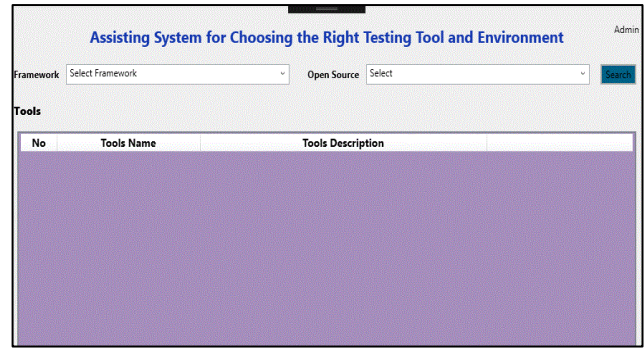


Fig. 3: User Log In for testing tool of framework

No	Tools Name	Tools Description
1	Selenium	Testing Framework for Web aApplications
2	Watn	Web Applications Testing in .NET
3	Microsoft Coded UI	UI testing(Web,WPF)
4	Sahi	Web Automation and Test Tool
5	Capybara	Web based Functional Testing Automation Framework
6	Pytest	Performance and Scalability Festing for Web Services
7	Windmill	Web Testing Tool
8	Katalon Studio	Web/API and Mobile(Android and IOS)
9	CasperJS	Web Applications
10	Jasmine	Unit testing framework for Javascript and Angular JS

Fig. 4: List and Description of all frameworks

Table I: Comparison of the proposed system with existing systems and also shows the deficiencies of proposed system

Sr. No.	Comparison with Present Systems	Deficiencies in the proposed work
1-	ROI is better than the previous tools	Test reusability may reduce efficiency
2-	It also reduces the risk of manual testing	The cost of manual testing incorporates the time, cost of manual hours and the endeavors of the testers
3-	Enhances Accuracy	As in the pressure testing, test mechanization has ended up being an aid here
4-	These tools will be adaptable to coordinate the future testing details	You will require less manual endeavors
5-	Highly robust	
6-	A stretch execute your test contents on more than one PC on a common system or server all	

We should talk about the financial benefits of our proposed methodology:

With automated testing, the outcomes are exact, as well as spare time. Numerous a periods, there are test cases with more than thousand lines of code and to compose it and test it would be extremely troublesome with manual testing. This can be effortlessly finished with automated testing. Likewise, these tools will ensure them all through the application like the databases, UI, web administrations, and so forth work as indicated by the necessities and enhancing the general test scope.

The cost of manual testing incorporates the time, cost of manual hours and the endeavors of the testers, QA directors, and so forth. Furthermore, on the off chance that you have automated testing tools, you will test quicker, effectively, productively, precisely and would convey applications without bugs.

VII. CONCLUSIONS

Testing automated systems diverse agents are planned with particular parts and they work together with each other to play out the test. The principle outline objectives have been to build up a successful and adaptable structure that backings diverse sorts of tests and use distinctive wellsprings of data about the systems under test to automate the test procedure. One of the oddities of this work is the utilization of test code which depends on the possibility of portable code. It gives benefits to expanding the execution and diminishing the multifaceted nature of test executer agents. Another curiosity of the work is the modeler agents that utilization diverse in-development hotspots for programmed test content age. A model of the proposed structure has been actualized and is utilized to play out a few examinations. The outcomes are promising and confirm the general plan of the structure.

REFERENCES

- [1] V. Garousi and D. Pfahl, "When to automate software testing? A decision-support approach based on process simulation," *Journal of Software: Evolution and Process*, vol. 28, no. 4, pp. 272–285, 2016.
- [2] M. Monier and M. El-mahdy, "Evaluation of automated web testing tools," *International Journal of Computer Applications Technology and Research*, vol. 4, no. 5, pp. 405-408, 2015.
- [3] V. Garousi and M. Mäntylä. "When and what to automate in software testing? A multi-vocal literature review," *Information and Software Technology*, vol. 76, pp. 92-117, 2016.
- [4] Rodrigues, Anderson and A. D. Neto. "Relevance and Impact of Critical Factors of Success in Software Test Automation lifecycle: A Survey," in *Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing*, ACM, 2016.
- [5] A. F. Musfira, M. F. Shakir and A. F. Munsifa. "Analysis of software testing challenges in Sri Lankan context," *South Eastern University of Sri Lanka*, 2016.
- [6] N. Bhateja, "A study on various software automation testing tools," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 6, Jun 2015.
- [7] P. Raulamo-Jurvanen, "Decision Support for Selecting Tools for Software Test Automation," *ACM SIGSOFT Software Engineering Notes*, vol. 41, no. 6, pp. 1-5, 2017.
- [8] Y. L. Floch, "Security testing in top IT strategy priority, world quality report" 2015-2016.
- [9] K. Kakkonen, et al. "Using Surveys and Web-Scraping to Select Tools for Software Testing Consultancy," in *Proceedings of Product Focused Software Process Improvement 17th International Conference*, Norway, 2016 pp.22-24.
- [10] S. Roohullah, et al. "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 2, 2016.
- [11] I. Anna, et al. "Evolving Rules for Action Selection in Automated Testing via Genetic Programming-A First Approach." at the Conf. of *Applications of Evolutionary Computation*, Europe, March 25, 2017.
- [12] E. F. Collins, "Software Test Automation Practices in Agile Development Environment :An Industry Experience Report," pp. 57–63, 2012.
- [13] T. Sheth, S. K. Singh, "Software Test Automation- Approach on evaluating test automation tools," *International Journal of Scientific and Research Publications*, vol. 5, no. 8, Aug 2015.
- [14] M. Imran, D. Hebaishy, and A. S. Alotaibi. "A Comparative Study of QTP and Load Runner Automated Testing Tools and their Contributions to Software Project Scenario," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 1, Jan 2016.
- [15] H. Mittal, "Comparative Analysis of Automated Functional Testing Tools," *Journal of Network Communications and Emerging Technologies*, vol. 6, no. 6, 2016.
- [16] R. Chauhan and I. Singh "Latest Research and Development on Software Testing Techniques and Tools," *International Journal of Current Engineering and Technology*, vol. 4, Aug 2014.