e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

AUTOMATION TESTING

Adarsh Malik*1, Ashima Mehta*2

*1Department Of Computer Science AndEngineering, Dronacharya College Of Engineering, Farruknagar, Gurgaon, India.

*2Faculty Of Computer Science AndEngineering, Dronacharya College Of Engineering, Farruknagar, Gurgaon, India.

ABSTRACT

Despite the proliferation of quality assurance tools and experts, IT software testing has been critical in the development and security of software systems in the internet space. The complexities and challenges of testing have pushed experts to the technology of test automation, which comes in several phases and with ultimate calls for heightened levels of management and control. In this paper, a clear demonstration of the automation approaches and the development approaches is eminent. The results of the research clearly show that automating software testing realms is the new strategy that is poised to change the usability, security, efficiency, and reliability of the current and future software programs. Testing is a very important activity in Software Development Process. It is to examine & modify source code. Effective Testing produces high quality software. This Paper deals with a significant and vital issue of Software Testing. Testing can be conducted manually as well as Automated. These Techniques have their own advantages & disadvantages. The Objective of this paper is to perform Automation Testing using Software Testing Tool "Selenium". With this web testing tool, test cases are automatically recorded in background while tester is entering the data in a web application screen.

I. INTRODUCTION

Test automation is the practice of running tests automatically, managing test data, and utilizing results to improve software quality. It's primarily a quality assurance measure, but its activities involve the commitment of the entire software production team. From business analysts to developers and DevOps engineers, getting the most out of test automation takes the inclusion of everyone. Information technology and software developments have become an indispensable part of life. Potentially touching millions of individuals and businesses, safety, and reliability of IT softwares and systems is paramount. Human interventions in software developments are never short of errors and bugs. IT software errors have proliferated in recentyears and can adversely affect service delivery and operational efficiency. Treating errors and a failure to debug in the preliminary stages of the software program development phase is the onset of a manageable IT system environment. Around the globe, numerous organizations have fallen victims of unscrupulous and inefficient software developments and integration. The National Institute of Standards and Technology (NIST) has released reports indicating the abysmal depths of adversities organizations have endured due to software errors and network testing bugs.

IT software testing refers to analyzing and interrogating a software program to unearth errors and secret faults. Software testing is projected to ensure that an integrated software program performs to meet satisfaction thresholds, access, and quality preservation. For an Information Technology system to be complete, the software testing realm shouldnot miss in the Software Development Life Cycle (SDLC). Under the requirements of SDLC, software development is not absolute under it is subjected to the testing process. Inherently, testing is not performed to demonstrate an error-freesystem but to establish a confidence wall that supports the installation and performance of the entire system.

Determinant factors for an efficient and successful software testing project include; choice and utilization of the appropriate automation tool, the right testing method, and the best framework at disposal. Such strategies ensure that theIT software and system under testing behaves and appears as expected (M. Sharma & R. Angmo, 2014). All unit and system testing protocols should not be compromised if the software's consumer is to harvest optimally from its installation. In this paper, the main focus is on test automation, especially IT software testing, the various tools, literature review on the recent past studies and findings, and the implication of the survey of the globalized landscape of IT and software testing, especially in the United States.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

II. LITERATURE REVIEW

Definition

Automated testing is also called test automation or automated QA testing. When executed well, it relieves much of the manual requirements of the testing lifecycle. In this review segment, the paper defines test automation. As the name preempts, it is elementary to assume that an automated approach of testing replaces manual testing. In simple terms, the software testing process is primarily free of human input and intervention. According to Thummalapenta et al. (2011), test automation is the function of establishing a systematic test case that is dominated by mechanical representations of interpretative paradigms. The representations may entail a program of scripting language and general-purpose programs. In their argument, the authorsclaim that test automation is all about attempts to free up testers and create automated testing protocols.

Automated testing is fundamentally the process of utilizing a software program away from the primary softwareunder analysis to manage and control test executions and comparison contexts at the end of the testing process(M. Polo et al. 2013). Once employed, automated testing is a powerful time-saver, and it can efficiently run large quantities of tests within a short period. Away from saving resources and manpower, automated testing improves the performance and quality of testing operations.

Impediments and Benefits for Automated Software Tests

Rafi et al. conducted a systematic literature analysis to seek the knowledge and understanding of the positivitiesand limitations of automated testing programs. The study also sought to compare the findings with the existing industrial survey records. In their research, they also integrated the critical meta-information paradigms about the reach focus. After intensive research, only 25 papers were eventually considered and proven relevant (Fitzgerald, B. & Stol, K.-J., 2014). Also, the research depicted a publication bias where the positive results are given publication rights. The relevanceof automation has been questioned by several experts in the field of IT and automated systems.

Industrial softwares are usually designed and released in a variety of versions. The series of products is a hidden impediment for those who don't understand the local build-ups in the designated developer's site. According to R. Ramler& K. Wolfmaier (2006), automated testing comes with many benefits. The processes are designed as a single solution that inhibits and plummets the recurring costs of testing with a proposed cost model to determine the ultimate automationstrategy. In an experiment, Berner et al. claimed that automating the processes of testing relieves experts and testers fromthe boredom and challenge of monotonous regression suite from time to time. The experts also stated that well planned and timely maintenance of the digital test suits is a necessity. In the eventuality, the maintenance process is a significantdrawback of the whole automated testing realms.

Why Automation

The proliferation of technology space has created sophisticated IT systems for organizations. Against every aspect of change and enhancement, testing software and programs play an integral role in shaping the way for a qualityproduct and service delivery (Dudekula Rafi, Katam Moses and Kai Petersen, 2012). In IT systems, there is a trio ofindispensable aspects, quality, speed, and certainty in running and delivering the product.

- **Improved operational efficiency**

Automation reduces time, effort and cost, whilst reducing manual errors, giving your business more time to focus on your primary objectives.

- **Time saving**

Repetitive tasks can be completed faster.

- **Improved quality and consistency**

Automating processes ensures high quality results as each task is performed identically, without human error.

- **Increased employee satisfaction**

Manual tasks are boring and laborious. Automation allows your employees to work on more engaging activities, and thus increases satisfaction.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

- **Increased customer satisfaction**

Happier employees, faster processing and time savings enable your teams to concentrate on providing better customer service, which all helps boosts customer satisfaction.

III. ADAVANTAGES AND DISADVANTAGES

Advantages	Disadvantages
Enhances the quick finding encounters ofbugs with heightened accuracy	Inherent knowledge of the tool required
Saves time and energy since the process isquite efficient	Considerable time is lost in choosing the right approach and tool
The test script can be comfortably repeated	Initial cost of acquiring a test tool is high
Improves the quality of the testing processand software accuracy	If playback approach is the option, testmaintenance is quite a burden
Increased coverage owing to the multiple testing tools for parallel testing encounters	High levels of proficiency required to creatthe scripts needed for the test.

IV. FRAMEWORKS FOR TEST AUTOMATION

Automation frameworks enable automation testers by simplifying the test development and execution activities. A typical automation framework provides an environment for executing test plans and generating repeatable output. They are specialized tools that assist you in your everyday test automation tasks. Whether it is a test runner, an action recording tool, or a web testing tool, it is there to remove all the hard work from building test scripts and leave you with more time to do quality checks. Test Automation is a proven, cost-effective approach to improving software development. Therefore, choosing the best test automation framework can prove crucial to your test results and QA timeframes. Strategic integration of test protocols is the way to success automation of software testing. A test framework inautomation is the extensible and aligned support structure under which the suite for test automation is designed, fabricated, and implemented. The framework comprises numerous tools and practices designed to assist and shape an efficient performance and reputation of software execution. It comprises physical structures used to test the creation and logical interaction between the fundamental components such as the coding standards, object repositories, test results storage, and information of accessing external resources. A framework ensures a standard way of adding, modifying, and exempting test functions and scripts. In the process of execution, the framework ensures scalability and ultimate reliability with minimized energy consumption.

The Significance of Test Automation Framework

An automation testing framework is a platform developed by integrating various hardware, software resources along with using various tools for automation testing and web service automation framework, based on a qualified set of assumptions. This framework enables efficient design and development of automated test scripts and ensures reliable analysis of issues or bugs for the system or application under test (AUT). The test automation framework has a critical role in the automation procedures of information technology and relevant software. Testers can instill an integrated script and record tests using the automation protocols. A complete automation framework helps the testing team achieve higher reusability and functionality of test components, script development that ensures easy maintainability, and access to high-end automation scripts (Dudekula Rafi, Katam Mosesand Kai Petersen, 2012). The integration of a framework in the testing process increases efficiency and speed; it also increases test accuracy and lowers the disturbing maintenance costs and risks. These frameworks are reliable foundation elements to a variety of software tests such as Functional Testing, Unit testing, and Performance testing. The important functions of software testing automation frameworks are broadly defined as they are effectively used to identify objects and arrange them to be reused in test scripts, perform some action on these identified objects and further also used to evaluate these objects to get the expected results.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

V. TYPES OF FRAMEWORKS FOR SOFTWARE TESTING

There are numerous categories of test automation frameworks with each entitled to its unique infrastructure. They usually differ in terms of their automation, key characteristics such as reusability, and simplicity in maintenance and repair. Always, it is paramount for the software expert and testing team to make the right choice of automating framework. Some of the test and analysis frameworks used in automation testing are discussed below:

1. Linear Scripting framework

The creation and execution of test scripts are done individually for each test case. Testers capture each test step such as browsing, navigation, user inputs, enforcing checkpoints, and then play the scripts to carry out the tests.

This type of framework is used in applications of small sizes.

2. Data-Driven Automation

In contrary to modular framework data-driven can easily pass and make the test scripts work properly for different sets of data.

Using this framework, the number of test scripts is significantly reduced and at the same time, it gives test coverage with reusable tests. Designing this framework requires coding skills and experience in test automation.

3. Keyword Driven Automation

Otherwise known as action word-based test automation, it performs automation test scripts based only on the keywords specified in the excel sheet.

The tester is able to work with keywords to develop any test automation script. This framework has shares similarities with data-driven testing.

Even though it can take some time to design keyword-driven automation, it isn't needed to be an expert in order to be able to write test scripts. Yet, employees with good test automation skills are needed.

4. Modular Automation

The complete application is broken down into smaller independent modules based on which testers create individual test scripts modules. These test scripts can be modified and combined to make larger test scripts to achieve the required scenario.

5. Hybrid Automation

As the name suggests, hybrid test automation is the combination of two or more frameworks. May teams are using this framework in the current market attempting to leverage the strengths of other frameworks.

6. Behaviour Driven Development

The aim of this framework is to develop a multi-functional platform that allows business analysts, developers, testers, etc. to continuously participate.

Selecting an Automation Tool

Technically, there are no right or lousy automation tools. The option and choice of an automation tool are significantly informed by the nature of the IT tool and software to be tasted. For instance, Selenium is the most commontool, but in the event of desktop-based software, the tool becomes unhelpful. Properunderstanding of the requirements attached to each testing software leads to an appropriately selected tool to match the predetermined requirements and needs. The following factors inform the choice of an automation tool:

a) The software and technology stack to be tested
b) Comprehensive testing requirements
c) The cost of the tool license
d) The depth of skill sets at the organization's disposal

VI. CONCLUSION

Software testing has grown to a prime necessity in the field of IT. As time passes, the technology soars I demand, and its impact is largely felt in software validations and verifications. Software testing is essential for two main reasons.First, it is the surety of software quality, and secondly, it is an integral segment of software costs and

management. Despite the high maintenance and implementation costs, test automation and the relevant test cases have a remarkably helpful return in the long runs of security and efficiency in the design, development, and launch of new software programs. Automated unit testing is a necessity if developers are to meet the fast-paced development deadlines of today's software markets. More and more development organizations are following Microsoft Corporation's lead and implementing the build-a-day approach. In the past, build dates were set and builds occurred at specific intervals that were measured in days, weeks, or months. New development processes such as those associated with object-oriented development and languages like Java require more frequent software builds.

VII. REFERENCES

[1] Khan, E. (2010). Different Forms of Software Testing Techniques for Finding Errors. International Journal of ComputerScience Issues Volume 7, Issue No. 3, 11-16.

[2] M. Sharma & R. Angmo. (2014). Web based Automation Testing and Tools . International Journal of ComputerScience and Information Technologies, Vol. 5 (1), 908-912.

[3] R. Ramler & K. Wolfmaier. (2006). Economic perspectives in test automation: balancing automated and manual testingwith opportunity cost. Proceedings of the 2006 international workshop on Automation of software test, 85-91.

[4] Berner, S., Weber, R. & Keller, R. K. . (2005). Observations and lessons learned from automated testing. . Proceedingsof the 27th international conference on Software engineering , 571-579.

[5] Fitzgerald, B. & Stol, K.-J. . (2014). Continuous software engineering and beyond: trends and challenges. Proceedingsof the 1st International Workshop on Rapid Continuous Software Engineering, 1-9.