Automation Testing in Information Technology

Sikender Mohsienuddin Mohammad

Wilmington University

Information Technology

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.

Introduction

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 recent years 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 should not 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-free system 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 the IT 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.

Literature Review

Definition

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 authors claim 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 software under 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 positivities and 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 relevance of 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 automation strategy. In an experiment, Berner et al. claimed that automating the processes of testing relieves experts and testers from the 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 significant drawback 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 quality product and service delivery (Dudekula Rafi, Katam Moses and Kai Petersen, 2012). In IT systems, there is a trio of indispensable aspects, quality, speed, and certainty in running and delivering the product.

Merits and Demerits

Merits	Demerits
Enhances the quick finding encounters of bugs with heightened accuracy	Inherent knowledge of the tool required
Saves time and energy since the process is quite 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 process and software accuracy	If playback approach is the option, test maintenance is quite a burden
Increased coverage owing to the multiple testing tools for parallel testing encounters	High levels of proficiency required to creat the scripts needed for the test.

Frameworks for Test Automation

Strategic integration of test protocols is the way to success automation of software testing. A test framework in automation 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

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 Moses and 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.

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 (Khan, 2010). Some of the test and analysis frameworks used in automation testing are discussed below:

- Modular Frameworks

In this category, the software program under testing is divided into small sections known as modules. Each module is tested and analyzed separately. A test script is then established for each module before a combined and broader set of tests in a hierarchical strategy (Khan, 2010). In the event larger sets are formed, the software testers can now represent a number of test cases.

- Data-Oriented Frameworks

The Data-driven category of frameworks is rife in environments where test data is separated and potentially sorted out from the script logic. Afterward, it is stored externally in external sources and storage such as Excel Spreadsheets, Text Files, SQL Tables, CSV files, and the ODBC repositories (Khan, 2010). In this framework approach, test scripts are inherently connected to the external source protocol to execute the data for popularity.

- Keyboard-Oriented Frameworks

In this framework, a similar approach as the one for data-driven frameworks is adopted. However, in this approach, the script logic and the test data are separated with keywords stored alongside their associated objects hence making them utter independent from the execution tool assigned to the tests. Keywords are unique script components that represent various actions and executions to test the software. They can be labeled as "login" or "click" or 'clicklink" or "verylink."

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 common tool, but in the event of desktop-based software, the tool becomes unhelpful (R. Ramler & K. Wolfmaier, 2006). Proper understanding 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.*

How the Research is going to Impact the United States

For many years, the United States has been an undisputed leader in Information Technology and Software Development. The cloud space and information world become more sophisticated and vulnerable. This research paper is a powerful insight into handling the testing complexities experts in the United States face. In the study, there are discussions on the various test automation tools and categories. The information is vital and informs the decisions of organizations and software developers seeking to make automated testing their priority. The research paper offers a powerful sensitization on the essentiality of moving to automation testing in information technology (M. Polo, P. Reales, M. Piattini, and C. Ebert, 2013). Experts and Cybersecurity specialists have an obligation to meet the research's understanding to improve their ever growing information space integrity and needs.

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.

References

M. Polo, P. Reales, M. Piattini, and C. Ebert. (2013). Test automation. *IEEE Software, Volume 30 Issue 1*, 84-89.

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

Dudekula Rafi, Katam Moses and Kai Petersen. (2012). "Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. *Automation of Software Test, 7th International Workshop*, DOI: 10.1109/IWAST.2012.6228988.

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

Khan, E. (2010). Different Forms of Software Testing Techniques for Finding Errors. *International Journal of Computer Science Issues Volume 7, Issue No. 3*, 11-16.

M. Sharma & R. Angmo. (2014). Web based Automation Testing and Tools . *International Journal of Computer Science and Information Technologies, Vol. 5 (1)*, 908-912.

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

Thummalapenta et al. (2011). IBM Research Report - Automating Test Automation. *Journal of Computer Science*, Retrieved from: https://researcher.watson.ibm.com/researcher/files/in-saurabhsinha/ata-techrep-RI11014.pdf.

Rahul Reddy Nadikattu, " CONTENT ANALYSIS OF AMERICAN & INDIAN COMICS ON INSTAGRAM USING MACHINE LEARNING", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.2, Issue 3, pp.86-103, September 2014, Available at :http://www.ijcrt.org/papers/IJCRT1133460.pdf