

Critical Analysis of software testing techniques and automation testing tools

Hafsa Riasat , Zaeem Nazir ,Saleem Zubair

Abstract

In the software testing process software or application is examined to get deficiency-free software. Software testing helps to understand the quality of the available software. In the testing process, automated testing has great significance. The use of automation tools while software testing reduces the number of workers and there are fewer chances of faults to overlook by the testers eye. So the automation tools in software testing are crucial to making software fruitful or acceptable. This paper aims to analyze software\program testing and its role in software quality. Then finding out various sorts of software tools and techniques and focus on the significance of automation testing.

Keywords: Software testing, testing, automated testing, testing tools, software quality

1. Introduction.

Testing is primarily a method surrounding validation and evaluation method that if the designed framework encounters the specifications specified by the consumer [1]. The test cycle consists mainly of many stages, from Test Preparation to Test Outcomes Analysis [2]. Testing done on a consumer application is called testing for software. Finding flaws in the software is the primary goal of debugging or software testing. The bug is an inconsistency or mistake caused by the software program or software application's actions. Software testing should be conducted to ensure that the software satisfies all the specifications stated in the design process, produces the right performance for different inputs, will achieve the job within an appropriate period limit, and can run in different environments [3]. In directive to assess applications for trustworthiness/reliability, performance, efficacy, robustness, etc., software testing is also carried out. Testing is expensive, but it appears to be costlier to stop program testing [4]. In software testing, the key goal is to check and confirm the completeness of the product. The product should conform to the product's technological specifications. Technical glitches should be accurately identified by the tester, and the tester should also guarantee that the program is glitch-free before it is launched on the market. High-quality test cases must be produced by testers and the correct problem report must be released [5]. The ANSI/IEEE 1059 normative description of testing is the means of reviewing a software item to detect the discrepancies between

existing and obligatory conditions (i.e. defects/errors/bugs) and assessing the software item's features [6]. It is typically distributed into a variety of discrete phases: a review of conditions, preparation of tests, production of tests, execution of tests, assessment of exit parameters, and closure of tests [7]. Both parties who are actively or indirectly interested with software creation operations need to be considered by software testers. Testing practices should be combined into a separate regulated testing process to be successful [8]. Software testing is a major V&V operation consisting of complex V&V of software actions on a limited range of test cases, opposed to ten predicted behaviors [9]. Verification and validation tasks are carried out during the whole period of software development to assess and increase product consistency. Verification and certification are common concepts that involve software verification, a major step in the life cycle of software development [10]. The risk identification of the consistency characteristics of complex software and hardware systems will be maintained over the whole life cycle [11]. Testing-related costs are on the rise; the tech industry has identified a need to reduce the rising costs of maintaining the research environment [12]. Test cases are considered the series of conditions and inputs used throughout testing, and the array of test cases is called a test suite [13]. Any software application testing automation has a series of operations, procedures and resources that will be processed to execute the software test. Automated processing speeds up the testing process. It saves time and resources and gives full coverage of the code [14]. Numerous automation testing tools such as Selenium Web driver, RFT, Coded UI, and

Appvance are accessible on the market, both commercial and open source [15]. Automated training solves the problems that manual testing poses. Automated training helps testers to build repeatable and interchangeable conditions for testing [16]. Manual testers manually discover the imperfections that expect them to work as end users and use program highlights to ensure their accurate and correct working [17]. In this article we review about software testing, different software testing tactics and automation testing.

- Hafsa Riasat is currently pursuing master's degree program in

software designers and produce a top-quality product, learning about numerous software development methods has been an important activity for the testing team. [3]. The article states that testing is the greatest important aspect of the life cycle of software products since it is everything on which the actual delivery of the application rests. It is time-consuming and an expensive operation, so it desires improved methods and creative methodologies. This enables the application of predictive tests and other different test metrics before and during the testing process [1]. The research states that monitoring is performed to guarantee and maximize the probability of the program or a product's success rate. An individual or a tester simply does not require good practical information for a good quality test, just some good first-hand experience to make the testing process quick and easy. As of now, the current popular methodologies are White Box, Black Box, and Grey Box; and none other than these are still effectively applied, but work is sure to proceed in this area. The currently established test case generation techniques are effective, but Genetic Algorithm is one of the new techniques and has a lot of potential in the coming period if it could be combined with and make it more efficient with any other already existing techniques [4] Automated testing methods help to build test cases and test scripts for manual and automation testing, respectively. The method for evaluating a small part of the system and also for testing the whole system is given by these techniques [5]. Ad hoc or exploratory testing methods are substantially

*software engineering in Superior University, Lahore Pakistan
E-mail: msse-f20-009@superior.edu.pk*

- Zaeem Nazir is currently pursuing PhD degree program in Computer science in Superior University, Lahore Pakistan, E-mail: phcs-Ss20-001 @superior.edu.pk

- Saleem Zubair in Superior University, Lahore Pakistan,

2. Background and literature review

The primary aim of any project is to provide any software product with the highest standard. That ensures that testing plays a very significant role in any digital product development. Several research tools have been developed and are currently working on the market. To compete with other

different from conventional manual program testing since the pre-designed test plans or test cases do not require this. Therefore, brainstorming is based on a more efficient methodology than limiting the tester to predefined tests [18]. Moreover, as none of the conceptualized ontologies are explicitly related to the components of NFRs and FRs, they did not find sufficient ontology for our intent [19]. Webb Miller and David Spooner implemented 'search-based program testing' in 1976 to produce test results from a version of the testing software (SUT). This enables the application of predictive tests and other different test metrics before and during the testing process. It will strengthen the current methods of testing, both for time savings and for an efficient and consistent finished product that not only satisfies the defined specifications but also provides optimum operating efficiency [7].

The EMVO algorithm provided can have very competitive results and best efficiency relative to most other tested algorithms and show EMVO's probability of understanding problems of unknown search spaces. Results from the EMVO algorithm are more effective and reliable. In comparison with other optimization algorithms, the results used by EMVO to produce test data for software testing are successful [20]. Finally, on average, seven experts' views will offer credible proof with a modest degree of precision. Practical and efficient ways are required to carry out tool evaluations that provide program practitioners with credible scientific evidence [21]. To classify program deviations, the paper by Rogstad and

Briand uses clustering algorithms such that each category includes deviations unique to a particular shift or regression fault. Shi et al. assess the potential of a test collection by black-box-dependent distance entropy to detect faults [22]. Marik et al addressed good practices in the production and testing of applications. In any step of software development, checking should be undertaken. Testing should begin at the level of design. Testing can be carried out in the Prototyping, Requirement Analysis, Formal Analysis, Architecture, Formal and Self-Testing phases [23]. Researchers find in this essay that experience of mastering software testing is a basic skill of practitioners in the software industry, and it is also the basic prerequisite for colleges and universities to educate software professionals [24].

3. Methodology

In this survey contextual analysis is done. we have explored diverse studies, surveys, and experimentation to examine the findings of software testing techniques and automatous testing tools. According to the perspective of our concern, this study is presenting the comparative analysis of explored testing and tools in literature with different perspectives. In this research we select the tools and techniques with respect to the different internal and external factors.

Internal factor:

- Models utilized in system development.
- Testing objectives
- Knowledge level of tester

External factor:

- Risk assessment.
- Cost and Time constraints
- Type of system.

3.1 Selection

For selection, we have investigated the basic points which are to be considered are given below.

- To find out the significance of software testing
- To find out the role of different testing techniques in software quality assurance.
- To find out how automated testing vs manual testing diverse.

- To find how automated testing upturns software quality.

After the selection procedure software testing was observed that how software testing plays a major role to achieve quality software and satisfying customers' requirements. The automated and manual testing was compared and observed their importance in software testing and their objective in quality assurance. The basic reason for choosing software testing and its tools was that it plays a crucial role in software quality which are striving for. There is a need to review software testing to achieve defect-free and quality software.

3.2 Software Testing Strategies:

A sketch that portrays the testing approach in Software Development Life Cycle (SDLC) is known as a software testing strategy. Several different methodologies for the design of test cases for the software are constituted in it. Later, a proper series of steps are developed from them which enable the successful testing of the software. Software engineers, testing teams, and project managers often develop these testing strategies. These testing strategies are of great importance. We define some of the testing strategies below.

A. Unit Testing:

The individual testing of the very basic units of software is known as unit testing. Unit testing is the lowermost level of testing. A unit is a very basic module or in other words, the smallest chunk of code that can be tested. Usually, unit tests are written by the developers and the refactoring is done till all the unit tests are passed successfully.

B. Integration Testing:

Once the unit testing is done, the individual components are combined or integrated. It's when the integration testing is used, that is when at least two or more modules are integrated and are required to be tested as an entire. Integration testing is performed on large interfaces. Verifying that all the software modules work correctly after integration is the prime motive of integration testing.

C. System Testing:

After the integration testing, the next phase is system testing. System testing is performed once the whole system is integrated as a unit, to test the performance of the whole system. All sorts of requirements, that is, both functional and non-functional requirements are validated and the system is given a green signal if and only if it successfully passes all the tests otherwise it is looped back into the Software Development Life Cycle.

D. Acceptance Testing:

After the system testing is done and the system successfully passes all the tests, then the system is delivered to the customers or users of the system. This is where acceptance testing takes place. Verification of the user requirements is the prime objective of the acceptance testing.

3.3 Software Testing Methodologies:

We describe some of the software testing methodologies below.

A. Black Box Testing:

Black box testing is the most popular software testing/inspection technique. As apparent by the name, this technique is based on the concept of the black box, which shrouds any internal details from the tester. The tester of the software only knows about the expected outputs and required specifications and has no idea about the code. This technique finds any incorrect functionalities and checks the behavior of the system for diverse types of input.

There are numerous types of black-box testing, some of which are listed below:

- Use Case testing
- Error Guessing
- All Pairs testing
- State Transition testing
- Error Guessing
- Cause-effect graph

B. White Box Testing:

White Box testing also recognized as clear box or glass box testing is a software testing procedure in which test cases are developed based on the code. In contrast to black-box testing, where the tester

has the complete knowledge about the code and programming talents are required to develop test cases. This technique is effective as it tests the interior structure of the software as well as its functionality. This way any bugs or errors and the issues caused by them can be detected and resolved. But despite that effectiveness, this technique is rarely used in practice. There are several types of white box e.g., Prime Path testing, Branch testing, Path testing, Control Flow testing etc.

C. Grey Box testing:

Grey box testing is a testing procedure that is a mix of both black box and white box testing techniques. It is a new technique that has evolved recently. This technique is utilized for penetration testing. It takes the benefit of both techniques. The tester has some knowledge about the code but is not completely familiar with the code. The tester also develops use cases for the software system like black-box testing.

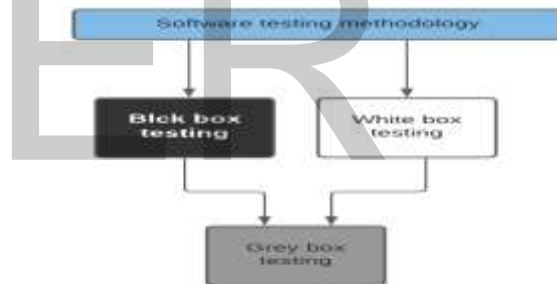


Figure 1 Software testing methodology

3.4 Software Testing tools:

Software testing apparatuses can be classified as Test management tools, Load testing tools, Functional testing tools.

3.4.1 Test management tools:

Test management tools are used to stock material about the testing plan, schedules, and activities. These tools allow access to data analysis and help make the testing process more efficient. These tools have different features and approaches to deal with testing. Arrangement of manual tests, accumulation of execution information from mechanized tests, handling various sorts of

circumstances, and input of data about discoveries is done using these tools.

Some of the test management tools are given below.

- *Micro Focus Quality Centre*: It is a test management instrument used for quality assurance purposes.
- *QA Complete*: It is a test management instrument that helps to effectively be able to the complete testing process.
- *Test Environment Toolkit (TET)*: It is an open-source, unsubstantiated command-line instrument that is used in numerous test applications.
- *Requirements and Testing Hub (RTH)*: It is an open-source test-management instrument, which also has requirements-management and bug-tracking competencies.

3.4.2 Functional Testing tools:

Functional testing tools are used to validate the software for the required functional specifications.

Some of the functional testing tools are given below.

- *Selenium*: It is an open-source and portable/transferable testing tool to test web applications that maintains a different browser, platforms, and operating system. It provides support for multiple languages including Python, C#, Java, and Ruby.
- *Junit*: It is an open-source framework intended in Java for writing and performing tests. It also has the capability of regression testing which helps the developers run the tests repeatedly during the development of the system.
- *QTP/UTF*: It is a graphical interface record playback automation apparatus/tool.

3.4.3 Load Testing tools:

Load testing is used to ascertain the behavior of an application under certain loads to check how it would behave when accessed by multiple users. The application is tested up to its limits.

There are various load testing tools, some of which are described below:

- *JMeter*: The Apache JMeter is open-source software, that is purely a Java application intended to load test functional behavior and measure the performance of an application. Test scripts can be written to verify and validate the expected results of an application using JMeter. It can be utilized to test the performance of static and dynamic resources which can be FTP servers, Java objects, CGI scripts, and databases, etc. Massive loads on a network or server can also be stimulated using these tools to test the performance and strength under various types of loads. It is a very flexible tool as it lets the testers take advantage of regular expressions to make assertions.
- *WebLoad*: WebLoad is an open-source software used for load testing. Performance and stress testing of web applications can be performed using this tool. But it is limited in features as there's a ceiling to the number of concurrent users it can generate. It can be run on both Windows and Linux. The prime power of this tool is its ease of use with features like automatic correlation, Javascript as its scripting language, and record/playback. This tool enables the users to perform load testing using HTML5, Ajax, etc on web applications.

3.5 Web Automation Testing Tools [38]:

Web testing is the term given to the testing of software which fully focuses on web apps [39]. There are numerous test automation tools for a web application. We discuss a few below:

Selenium:

Selenium is an open-source test automation suite that is designed for web applications supporting various browsers and applications. It was developed in 2004 by Jason Huggins. Lots of improvement has been made so far since its development and it is one of the strongest tools for web automation testing.

WinRunner:

HP WinRunner software is an automation functional graphical user interface GUI testing tool/instrument which enables the consumer to capture, replay and verify UI interaction as a test script. It is developed by Mercury Interactive.

3.6 Automation Testing Tools of Mobile Applications:

Robotium:

Robotium is a user interface automation outline/framework for android. It is a free-to-use instrument utilize by enterprises and individuals. Java is used as the programming language in Robotium. It works with android devices natively and can run tests in parallel mode in contradiction of various devices.

Ranorex:

Ranorex is a testing framework and testing tool that supports the script-less way of working and coding capabilities. It uses standard programming languages like C#, VB.NET as a base.

Appium:

Appium is an open-source, free-to-use cross-platform test automation framework that allows writing tests against multiple platforms, that is, iOS and Android, using the identical API and permits the code reuse between iOS and Android test suites. Appium can be tested on both real devices and simulators. Ruby, C#, Java, and other languages used in Web Driver Library, are used in Appium.

MoneyTalk:

The MoneyTalk is an open-source, power functional yet elementary to use testing tool which automates tests for mobile, hybrid iOS, native, and android applications. The MoneyTalk Agents are libraries used for android and iOS that are linked to the apps to be tested. It is installed in android applications, iOS applications, or both for cross platform testing. The MoneyTalk agents allow applications to record and play MoneyTalk commands where each command performs a verification step or user interface action.

UIAutomator:

UIAutomator, provided by Google's Android, is a testing framework. The testing performed by UIAutomator makes sure that the application meets its functional specifications and achieves a high standard quality which is to be effectively adopted by users. It is informal as it handles

asynchronous events such as dialogs, alerts, and toasts.

4. Results and discussion

We investigate that software testing plays a major role in quality assurance because if our software is defect-free then it is of high quality. There must be some kind of testing strategies to have quality testing. Quality software enhances the system's trust and use and customer satisfaction level is also involved in it as well. For the sake of testing, we have manual and automated testing. The testing process includes automated testing and test matrices this help in the effective, efficient and reliable product which meet all the requirement and provide functioning productivity. Test automation is the major progression is testing. Automated testing compares the results it is not time-consuming and laborious like manual testing in which users perform testing itself and where there is a chance of defects. Automation testing has taken place over manual testing as it brings out more defects and gaps which is not possible to get in manual testing. So automation testing enhances the software quality as well.

5. Conclusion and future work

The primary aim of any project is to produce any digital product/software of the highest quality. That ensures that testing plays a very significant role in any digital product development. Several research tools have been developed and are currently working on the market. To compete with other software developers and produce a top-quality product, learning about various software testing tools has been an important activity for the testing team. This paper explores some relevant and often used high demand applications and also introduces a report on various automated testing tools used on various platforms and highlights the importance of automation testing over manual testing and concludes that the unseen needs of the testing need to be understood and software built verifies the secret research criteria and develops tools that verify all concealed needs.

REFERENCES:

- [1] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, "Software Testing Techniques: A Literature Review," p. 6.
- [2] A. K. Arumugam, "Software Testing Techniques & New Trends," *International Journal of Engineering Research*, vol. 8, no. 12, p. 6.
- [3] K. Sneha, "Research on software testing techniques and software automation testing tools," p. 5.
- [4] J. Gaur, A. Goya, T. Choudhury, and S. Sabitha, "A Walk Through of Software Testing Techniques," *Information Technology*, p. 6, 2016.
- [5] D. S. Taley and D. B. Pathak, "Comprehensive Study of Software Testing Techniques and Strategies: A Review," *International Journal of Engineering Research*, vol. 9, no. 08, p. 6.
- [6] V. R. Basili, "Comparing the Effectiveness of Software Testing Strategies," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, no. 12, p. 19, 1987.
- [7] H. Hourani, A. Hammad, and M. Lafi, "The Impact of Artificial Intelligence on Software Testing," p. 6.
- [8] S. Sharma, L. Raja, and D. P. Bhatt, "Role of ontology in software testing," p. 10.
- [9] M. Dadkhah, S. Araban, and S. Paydar, "A systematic literature review on semantic web enabled software testing," *Journal of Systems and Software*, vol. 162, p. 110485, Apr. 2020, doi: 10.1016/j.jss.2019.110485.
- [10] I. Santos and S. M. Melo, "Towards a unified catalog of attributes to guide industry in software testing technique selection," p. 10.
- [11] E. Jharko, "Verification and Software Quality Assurance for Nuclear Power Engineering," p. 4.
- [12] T. Hynninen, J. Kasurinen, A. Knutas, and O. Taipale, "Software testing: Survey of the industry practices," p. 6.
- [13] M. Khari, "An extensive evaluation of search-based software testing: a review," p. 14.
- [14] M. Khari, "Empirical Evaluation of Automated Test Suite Generation and Optimization," *Arabian Journal for Science and Engineering*, p. 17.
- [15] A. S. Gadwal and Lalji Prasad, "Comparative review of the literature of automated testing tools," 2020, doi: 10.13140/RG.2.2.36836.19848.
- [16] H. V. Gamido and M. V. Gamido, "Comparative review of the features of automated software testing tools," vol. 9, no. 5, p. 7, 2019.
- [17] V. Jain and D. K. Rajnish, "Comparative Study of Software Automation Testing Tools: OpenScript and Selenium," vol. 8, no. 2, p. 5, 2018.

[18] I. Bhatti, J. A. Siddiqi, A. Moiz, and Z. A. Memon, "Towards Ad hoc Testing Technique Effectiveness in Software Testing Life Cycle," p. 6.

[19] G. Tebes, "Analyzing and documenting the systematic review results of software testing ontologies," *Information and Software Technology*, p. 23, 2020.

[20] S. N. Fakhouri, "Enhanced Optimizer Algorithm and its Application to Software Testing," p. 25.

[21] P. Raulamo-Jurvanen, S. Hosio, and M. V. Mäntylä, "Practitioner Evaluations on Software Testing Tools," p. 10, 2019.

[23] A. Inam, "Test Automation Assessment part 1," 2019, doi: 10.13140/RG.2.2.11161.98405.

[24] H. Chen, X. Wang, and L. Pan, "Research On Teaching Methods And Tools Of Software Testing," p. 4.

IJSER