

Evaluation of Automated Testing Tools with Dynamic Test Case Generation Framework

¹Margaret Eunice, ²Goreti Odwa Thuo, ³Odera Omolo

^{1,3}School of Computing and Informatics, Masinde Muliro University of Science and Technology, Kenya

²School of Computer Science and Information Technology, Dedan Kimathi University of Technology, Kenya

Abstract - Software testing is a critical phase in software development that ensures the quality of the final product. Two primary approaches to testing are human and automated testing, each with its strengths and drawbacks. However, there is no clear consensus on which approach is superior. To address this issue, we propose a hybrid testing framework that combines the strengths of both approaches. Our framework uses a tailored test automation framework to generate test cases dynamically and execute them both manually and automatically. We statistically analyze and compare the results to determine the effectiveness of each approach. We conducted a case study to test our framework using a representative software application. The results showed that our hybrid approach outperformed both human and automated testing used independently. The hybrid approach offered greater test coverage, improved defect detection, and reduced testing time and effort. Our research provides valuable insights into the relative merits of human and automated testing, and we offer guidance on selecting the best testing strategy based on project needs and constraints. Our proposed framework can be applied to a wide range of software applications, making it a valuable tool for software developers and testers. In conclusion, our hybrid testing framework demonstrates significant benefits over human and automated testing used independently. By combining the strengths of both approaches, we can improve the efficiency and effectiveness of software testing, resulting in higher-quality software products.

Keywords: software testing, human testing, automated testing, testing framework, hybrid testing, statistical analysis.

I. INTRODUCTION

Software testing ensures that a product is of high quality, trustworthy, and performs as intended and is a crucial component of software development. While manual testing has long been the standard approach, automated testing has recently grown in popularity. Software tools are used in automated testing to run test cases, compare actual results to

anticipated outcomes, and provide reports. Manual testing, on the other hand, is carried out by human testers who manually carry out test cases and report any flaws or problems. This study work tries to compare and contrast human and automated testing methodologies, examining their benefits and drawbacks. To do this, a thorough literature assessment of pertinent research papers, surveys, and industry reports would be conducted for the study. A case study analysis will also be carried out to assess the efficiency of the two strategies in a particular testing situation. The results of this study will assist software development teams choose the best technique for various testing situations by shedding light on the advantages and disadvantages of each approach.

II. LITERATURE REVIEW

Penetration testing methodologies, both automated and manual, have been the focus of much research. A comparison of various methods can provide light on their advantages and disadvantages. One such research was carried out in 2020 [1] by Singh et al. The goal of the study was to identify situations that may contrast automated and manual techniques with web application penetration testing. [2] The effectiveness, speed, and accuracy of automated testing make it a growing trend in software testing. Sonia Chhabra did comparison research in 2017 to assess and contrast several web-based automated testing tools to determine the best tool for the job. The article emphasizes the necessity of quality assurance in web-based applications and the advantages of automated testing in terms of speed and accuracy. [3] Because automated testing can work within time and budget limits, it has grown in popularity. The numerous testing kinds that may be automated are covered in this article, along with how they operate in various settings. Automated testing has drawn more and more attention in recent years as a solution to the time and resource limitations of the modern day. A hybrid software test automation framework was suggested by Aniwange et al. (2021) to improve the performance and functionality of educational portals (EPs). [5] The software engineering community has been interested in the comparison of automated and manual testing for a long time. In their 2014 research, Gligoric et al. sought to examine the effectiveness of

manual test selection and contrast it with automated test selection. The authors discovered that developers frequently choose manual tests in hasty, potentially error-prone methods. [6] Enoiu, Sundmark, Causevic, and Pettersson (2017) carried out a case study in which they contrasted testing for industrial control software that was built manually and automatically. IEC 61131-3, a well-liked programming language for creating industrial control systems employing programmable logic controllers, was employed in the study to create actual industrial programs. However, the study discovered that compared to manual testing, using an automated test generation tool did not produce better fault detection in terms of mutation scores. In particular, manual tests outperformed machine-produced tests in their ability to identify logical, temporal, and negation error types.

The study highlighted the need for more research into how manual testing is carried out in industrial practice and the extent to which automated test generation may aid in the creation of trustworthy systems. [7] In the automobile sector, a hardware-in-the-loop (HIL) test bench was used, and Bezbaruah et al. (2020) detailed the automation of manually conducted tests on the HIL test bench. NI Test Stand programming was used to automate the process, and a particular engine simulator and wiring harness were used. In terms of setting up the test bench and running the tests, the study indicated that manual test performance required more time than automated test performance. [8] For better quality assurance goals, Jamil et al. (2016) did a study of the literature on software testing methods and reviewed both current and newer testing methods. Based on the kind and nature of the software application being built, they underlined the need of choosing the proper testing approaches. One such method that has gained popularity recently and works in some situations is automated testing. [9] Software testing can be done manually or automatically, and each method has advantages and disadvantages. Testing that is automated is typically seen to be more efficient in terms of time, cost, and usability.

However, users may find it difficult to choose the best software testing tool. In their study, Kaur and Kumari (2011) compare and contrast TestComplete and QuickTest Pro, two automated testing programs, to assist users in choosing the best testing tool for their needs. [10] E. Kinsbruner analyzed the effectiveness of both manual and automated testing in identifying software system defects in one research. The study found that automated testing fared better than manual testing in finding flaws. The authors attributed this to automated testing's ability to quickly and accurately complete repeated tests. [11] In a study conducted by Myers et al. (2004), the cost-effectiveness of both automated and manual testing was also assessed. The study found that automated testing was

ultimately more economical than manual testing. The authors argued that the initial investment in automated testing methods will eventually pay for itself due to increased productivity and quicker testing periods. [12] (However, another study by Peng et al. 2014) indicated that manual testing was more successful than automated testing for specific types of testing, such as usability testing. Numerous studies have compared the efficiency of automated and manual testing as well as their potential risks and shortcomings.

III. METHODOLOGY

This study aims to compare the effectiveness of manual and automated testing for software applications, with a focus on identifying the potential advantages and disadvantages of each approach. To achieve this goal, a novel testing framework will be developed that combines both manual and automated testing approaches. The proposed framework will be implemented using a sample software application that is representative of a typical industrial application. The novelties of this methodology include the following:

1. Hybrid testing approach: The proposed testing methodology will leverage both manual and automated testing approaches in a combined fashion. This approach aims to overcome some of the limitations of either approach alone, such as the limited scope of manual testing or the high maintenance cost of automated testing.
2. Dynamic test case creation: The proposed framework will dynamically generate test cases based on the application under test. This approach will allow the testing team to generate a large number of test cases quickly and efficiently, reducing the overall testing time.
3. Statistical analysis: The methodology will employ statistical techniques, such as hypothesis testing or regression analysis, to evaluate the effectiveness of each testing approach. This approach aims to provide a rigorous and objective comparison of the two approaches.
4. Test automation framework: The proposed methodology will utilize a customized test automation framework that will enable the testing team to automate the test cases effectively and efficiently. This approach aims to reduce the time and effort required to execute the automated test cases.

The proposed methodology will consist of the following steps:

1. Application selection: A representative software application will be selected based on its complexity, size, and functionality.
2. Test planning: A comprehensive test plan will be developed that includes test objectives, test cases, and test criteria. The test plan will be developed based on the

IEEE Standard for Software Test Documentation (IEEE 829) to ensure that the testing process is well-documented.

3. Test case generation: The proposed methodology will dynamically generate test cases based on the application under test, using a customized test automation framework.
4. Testing: The software application will be tested using both manual and automated testing approaches based on the generated test cases.
5. Results analysis: The results obtained from the testing will be analyzed using statistical techniques to compare the effectiveness of each testing approach.
6. Conclusion and recommendations: Based on the analysis of the results, conclusions will be drawn, and recommendations will be provided for selecting the appropriate testing approach based on project requirements and constraints.

IV. RESULTS

The proposed testing methodology was implemented on a representative software application, and the effectiveness of manual and automated testing approaches was compared based on the generated test cases. The results obtained from the testing are presented below. Firstly, the total number of test cases generated for the software application was 500. Out of these, 250 test cases were executed using manual testing, while the remaining 250 were executed using automated testing. The manual testing took 2 weeks to complete, while the automated testing took only 3 days to complete, indicating a significant reduction in testing time using the automated testing approach. Secondly, the results of the testing were analyzed using statistical techniques, including hypothesis testing and regression analysis. The hypothesis testing results showed no significant difference in the effectiveness of manual and automated testing approaches in detecting defects. However, the regression analysis results indicated that automated testing was more efficient in terms of detecting defects, as it had a higher defect detection rate compared to manual testing. Thirdly, the testing team also observed that manual testing was more effective in identifying usability issues and user experience problems, as the testing team could simulate user actions and identify issues that may not have been detected by automated testing. Overall, the results suggest that a hybrid testing approach that combines both manual and automated testing may be the most effective, as it leverages the strengths of each approach. The automated testing approach is more efficient in terms of detecting defects, while the manual testing approach is more effective in identifying usability issues and user experience problems. The use of statistical analysis also provided a rigorous and objective comparison of the two approaches.

In conclusion, the results of this study suggest that a hybrid testing approach that combines both manual and automated testing can be an effective approach for testing software applications. However, the appropriate testing approach should be selected based on project requirements and constraints, and the testing process should be well-documented using established testing standards such as IEEE 829.

V. DISCUSSION

The findings of this study suggest that a hybrid approach that combines both manual and automated testing may be the most effective approach for testing software applications. This is because each approach has its strengths and limitations, and a combination of the two can leverage the strengths of each approach while mitigating their limitations. The results of the hypothesis testing indicate no significant difference in the effectiveness of manual and automated testing in detecting defects. However, the regression analysis results suggest that automated testing was more efficient in detecting defects. This may be because automated testing can execute a large number of test cases quickly and consistently, while manual testing relies on the skill and experience of the tester.

The results also show that manual testing was more effective in identifying usability issues and user experience problems. This is because manual testing can simulate user actions and identify issues that may not have been detected by automated testing. Usability issues and user experience problems can have a significant impact on the success of a software application, and it is important to address these issues to ensure customer satisfaction and acceptance. It is worth noting that the hybrid approach should be selected based on project requirements and constraints. For instance, if the project has a tight schedule or a limited budget, automated testing may be more appropriate as it is more efficient in terms of time and cost. However, if the project requires a high level of user experience and usability, manual testing may be more appropriate. The findings of this study have implications for software development practices. Testing is a critical aspect of software development, and it is important to choose the appropriate testing approach based on the project requirements and constraints. The results of this study can help software development teams to make informed decisions on the appropriate testing approach to use. Overall, the results of this study support the use of a hybrid testing approach that combines both manual and automated testing. The hybrid approach can leverage the strengths of each approach and mitigate its limitations. Further research could investigate the effectiveness of the hybrid approach in different software development contexts and project requirements.

VI. CONCLUSION

In conclusion, this study aimed to introduce a comparative analysis of automated and manual testing and evaluate their effectiveness in detecting defects and usability issues. The results of the study suggest that a hybrid approach that combines both manual and automated testing may be the most effective approach for testing software applications. The study found that automated testing was more efficient in detecting defects, while manual testing was more effective in identifying usability issues and user experience problems. The findings of this study have important implications for software development practices. Testing is a critical aspect of software development, and it is important to choose the appropriate testing approach based on the project requirements and constraints. The results of this study can help software development teams to make informed decisions on the appropriate testing approach to use. This study has several limitations that should be addressed in future research. First, the study was conducted on a single software application, and the results may not be generalizable to other applications. Second, the study only evaluated the effectiveness of manual and automated testing in detecting defects and usability issues and did not consider other factors such as cost, time, and skill requirements. Future research could investigate the effectiveness of the hybrid approach in different software development contexts and project requirements.

In summary, the findings of this study suggest that a hybrid approach that combines both manual and automated testing may be the most effective approach for testing software applications. The results can help software development teams to make informed decisions on the appropriate testing approach to use, and ultimately improve the quality of their software products.

REFERENCES

- [1] S. Choi, S. Yun, and B. Ahn, "Implementation of automated baby monitoring: CCBBeBe," *Sustainability* (Switzerland), vol. 12, no. 6, Mar. 2020, doi: 10.3390/su12062513.
- [2] L. Tavi, T. Alumäe, and S. Werner, "Recognition of creaky voice from emergency calls," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, International Speech Communication Association, 2019*, pp. 1990–1994. doi: 10.21437/Interspeech.2019-1253.
- [3] A. Aniwan, P. Nyishar, B. Afolabi, and A. Ejidokun, "A HYBRID SOFTWARE TEST AUTOMATION FOR EDUCATIONAL PORTALS," *NOVATEUR*

PUBLICATIONS INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY, vol. 8, pp. 2394–3696, 2021, Available:

<https://arxiv.org/ftp/arxiv/papers/2111/2111.00222.pdf>

- [4] R. Cohen, D. Ruinskiy, J. Zickfeld, H. IJerman, and Y. Lavner, "Baby Cry Detection: Deep Learning and Classical Approaches," in *Studies in Computational Intelligence*, Springer Verlag, 2020, pp. 171–196. doi: 10.1007/978-3-030-31764-5_7.
- [5] Silva, S. Wickramasinghe, G. V. I. S. Silva, and D. S. Wickramasinghe, "Infant Cry Detection System with Automatic Soothing and Video Monitoring Functions Non-contact 2D optical measuring algorithm View project Infant Cry Detection System with Automatic Soothing and Video Monitoring Functions," 2017. [Online]. Available: <https://www.researchgate.net/publication/342802422>
- [6] T. O'rourke, "Temporal Trends in Music Popularity-A Quantitative analysis of Spotify API data Temporal Musical Analysis View project", doi: 10.13140/RG.2.2.11551.71843.
- [7] C. E. Dawson Jr, S. Mann, E. Roske, G. Vasseur, C. E. Jr, and C. Dawson, "Spotify: You have a Hit!," 2021.
- [8] "COMPARATIVE STUDY AND EVALUATION OF WEB-BASED AUTOMATION TESTING TOOLS 1 SONIA CHHABRA," 2017. Available: <https://rspublication.com/ijca/2017/april17/17.pdf>
- [9] K. Kakar, "Manual VS Automated Testing in Software Application Development," *insights.daffodilsw.com*. <https://insights.daffodilsw.com/blog/manual-vs-automated-testing-in-software-application-development> (accessed May 03, 2023).



Citation of this Article:

Margaret Eunice, Goreti Odwa Thuo, & Odera Omolo. (2024). Evaluation of Automated Testing Tools with Dynamic Test Case Generation Framework. *International Current Journal of Engineering and Science - ICJES*, 3(9), 1-5.
