INTI JOURNAL | Vol.2025, Issue 1, No.10
eISSN:2600-7320

Modernizing Testing: A Comparative Review of Test Automation Frameworks and AI Tools

Sothy Sundara Raju[1], Wai Yie Leong[2*]

[1,2]INTI International University, Persiaran Perdana BBN Putra Nilai, 71800 Nilai, Negeri Sembilan, Malaysia

[*]**Email:** waiyie.leong@newinti.edu.my

Abstract

Artificial Intelligence has emerged as a revolution in software testing due to the software industry's rapid expansion, allowing Quality Assurance (QA) teams to produce higher-quality software more quickly and effectively. The comparative assessment of test automation frameworks and Artificial Intelligence (AI) powered tools presented in this journal emphasises the revolutionary potential of incorporating advanced AI capabilities into software testing procedures. The objective of this study is to create a framework that will enable organisations to implement AI-driven automation in software testing that is compatible with their requirements. The expected results from this research are to come up with a framework that improves accuracy, scalability, and adherence to software standards while minimizing manual effort and increasing overall testing efficiency. The methodology combines questionnaires and a literature review to discover the organisation's automation technologies and their influence on increasing product quality. A hybrid methodology will be used for this study that will have both quantitative and qualitative data via surveys and interviews review to discover the organisation's automation technologies and their influence on increasing product quality.

Keywords

Artificial Intelligence (AI), Quality Assurance (QA), Small and Medium-Sized Enterprise (SMEs), Software Testing, process innovation

Introduction

The development of software systems has become increasingly complex and sophisticated in recent years. Software solutions need to be dependable and of a high calibre to satisfy the needs and expectations of clients. To increase the overall quality of software, it is necessary to address the several sub-attributes that make up software reliability. In recent years, Artificial Intelligence (AI) has become a significant system that assists in software testing, providing new strategies for automating test cases, code analysis, anomaly detection, performance testing, regression testing, reducing test execution time, and wider test coverage. According to (Abdulwareth & Al-Shargabi,

Submission: 8 January 2025; **Acceptance:** 4 April 2025; **Available online:** April 2025

2021), in the software industry, testing comprises around 40% of the System Development Life Cycle (SDLC), regardless of the software development strategy that is used. Hence, testing should be handled and accomplished using the best practices to produce high-quality software with cost-effective methods. This paper reviews five AI testing tools: Testim, Mabl, Functionize, Rainforest QA, and Katalon Studio- based on platform support, language compatibility, setup ease, usability, scripting capabilities, and testing types. It also outlines six automation frameworks and their relationships: linear, modular, library architecture, data-driven, keyword-driven, and hybrid.

The research would assess the effectiveness of AI in software testing for SMEs by conducting empirical assessment, collecting and analysing real-world data from SMEs that have adopted AI tools. In this research, quantitative data will be collected in terms of time savings, cost reduction, and defect detection rates. Qualitative feedback will be collected from developers and testers on the usability and impact of the AI tools in the software testing. The research will focus on real-world case studies from SMEs that have successfully implemented AI in test automation, providing practical insights into what worked and what didn’t. This would help bridge the gap between theoretical challenges and actual application. The planned research will provide both organisations and educational institutions with useful information that will help them choose and use AI-powered software testing technologies that best suit their unique requirements. This study advances knowledge of AI's function in software testing and promotes the adoption of these technologies by businesses in order to improve quality assurance, expedite development cycles, and satisfy customer requirements.

According to (Bajjouk et al., 2021), testing is one of the most crucial stages in any methodology since businesses use a variety of development approaches, and it can find flaws and problems if they are present in the program. Testing is not as simple as it may seem, though, as there are numerous testing types, and each can be used in a certain scenario. In order to deliver high-quality products to clients and ensure that the software functions in accordance with their needs, software testing is a crucial component of the software development life cycle (SDLC), according to (Thakur et al., 2023). In contrast to traditional testing methods, there are a number of AI-powered advanced testing tools that can create test cases, cross-check results with expected results, and closely and independently monitor software quality throughout the development process with less manual labour, according to (Thakur et al., 2023).

In the world quality report, as stated by Muhi Majzoub (World Quality Report, 2023), over 50% of companies believe that using AI would boost productivity and accelerate development, and they want to keep investing in the implementation of scaling the quality engineering process in line with these expectations. According to (Prakash & Rubin, 2024), the development of generative artificial intelligence tools (GAITs) has attracted a lot of attention lately. These technologies can produce fresh and creative answers to challenging issues because they are driven by sophisticated machine learning algorithms. In order to increase efficiency and effectiveness, businesses in a variety of industries are increasingly looking to minimize human participation and heavily rely on AI solutions.

As stated by (Sani & Jan, 2024), automation testing tools must improve testing efficiency, reduce manual effort, and ensure comprehensive test coverage. However, the challenges primarily hinge on tool selection, cost, ability to function across various platforms, and integration with

Continuous Integration and Continuous Development (CI/CD) pipelines. Moreover, the complexity of managing evolving technologies and the limitations of current automation frameworks were significant concerns. This comparative analysis highlights the significance of the suitable tool capabilities with project requirements while tackling the intrinsic challenges of scalability, accuracy, and tool compatibility in various environments.

Methodology

Analysis

Figure 1 shows the targeted keywords by the number of publications, and the targeted keyword is analysed in ≥0.50% of publications, or ≥250 publications. This level was determined by taking into account the trade-off between providing a comprehensive impression of the testing field and clarity, as well as by looking at the drop-off in significance over the top 10 most prevalent keywords. Additionally, we looked at the keywords that, on average, have gotten the most citations per publication. Here, we look at every keyword that has an average of at least 20 citations. This threshold, which produces 23 keywords, was selected to allow for easier comparison because it produces a quantity that is comparable to the number of most popular keywords. The Scopus database was used to collect publications in order to obtain a comprehensive picture of software testing. Numerous conference and journal publication venues are covered by the extensive meta-database Scopus. The search was on all publications returned for the search keyword ‘‘software testing’’ on September 26, 2020, stated by (Salahirad et al., 2023)

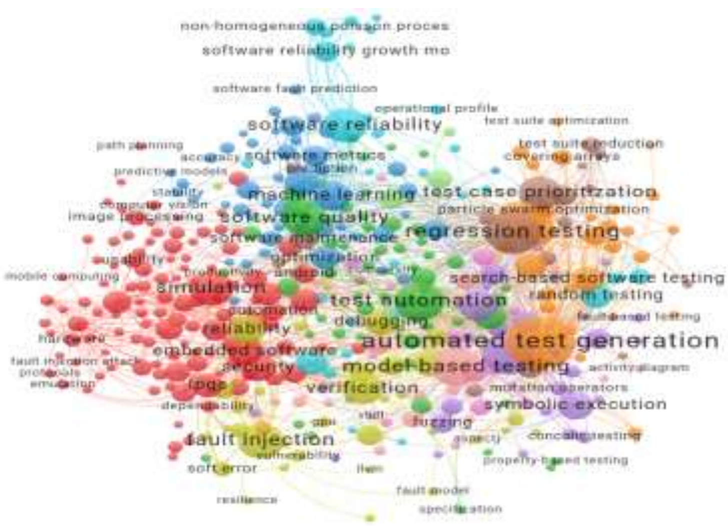

Figure 1. A visualization of the connections between publication keywords
Source: Mapping the structure and evolution of software testing research over the past three decades

A hybrid methodology will be used for this study that will have both quantitative and qualitative data via surveys and interviews. SME in Malaysia that still use manual testing as well as SME that use AI for their software testing will be receiving questionnaires. Data that will be collected are from companies that have used AI in software testing for a range of lengths of time, including those that have been using it for more than five years, less than two years, and those that continue to use manual testing. Data will also be gathered from companies that continue to do software testing manually in order to examine the reasons behind their resistance to automation. The data that will be collected includes current software testing procedures, the difficulties SMEs encounter while testing software, and their willingness to employ AI tools for software testing. The percentage of SMEs that are currently adopting AI will be determined by analysing the data that will be gathered from surveys and interviews. Their success stories of integrating AI into their software testing will be used for analysis. The most popular automation tools will be determined, and tools that work well on various platforms will be assessed. A framework that is both economical and effective for Malaysian SMEs will be created based on the results of the survey and the current literature review. Focus groups will be used to validate the new framework. Table 1, shows a comparison of the literature review research method, research techniques and the gaps from the research which was conducted.

Table 1. Comparison of Literature Review Research Method and Research Techniques

Literature Review	**Research Model**	**Research Techniques**	**Gap (Lacking)**
Hands-on Use and Adaptation of AI in Dev. and Test. Soft. Applications (Haeggström, 2024)	Theoretical Model using Literature Review	Quantitative Techniques. Interview questions are distributed to one company	There is a need for empirical assessment by collecting and analysing real-world data from SMEs that have adopted AI tools and developing a testing framework that will be validated by a focus group.
Empirical Analysis of Widely Used Website Automated Testing Tools (Sani & Jan, 2024)	Literature Review and Empirical Analysis	Multivocal Literature Review	Several gaps were identified, such as prioritising test cases, integrating scalable cloud solutions, automating test result analysis and feedback, and leveraging AI and machine learning in test automation to enhance prediction.
Soft. Test. Techniques and Tools (Izzat & Saleem, 2023)	Theoretical Model using Literature Review	Multivocal Literature Review	The absence of experience-based testing methodologies in the literature review is the highlighted gap.
Security Testing of Web Application (Aydos et al., 2022)	Systematic literature mapping (SLM) and systematic literature review (SLR) principles		More research involving industry and academics is required. The evidence supporting web application security testing and empirical assessments of efficacy and efficiency needs to be reviewed and improved.

AI-powered test automation tools: A systematic review and empirical evaluation (Garousi et al., 2024)	Empirical assessment on two automation tools Parasoft Selenic and SmartBear.	Multivocal Literature Review	Conduct surveys with practitioners on the benefits and the limitations they faced from using these tools. Research on more automation tools is needed by conducting a wider scope of case studies.
AI In Test Auto.: Overcoming Challenges, Embracing Imperatives (Khankhoje, 2024)	Empirical Model on case studies	Qualitative Techniques	Need to research real-world case studies from SMEs that have successfully implemented AI in test automation, providing practical insights.
Integrating AI in test. automation: Enh. test coverage and predictive analy. for improved soft. quality (Prathyusha Nama, 2024)	Empirical Model on case studies	Qualitative interviews with quantitative surveys	This research lacks challenges faced by skilled personnel to manage AI tools and further enhance testing practices.
Pioneering Test. Technologies: Advancing Soft. Quality Through Innovation Method. and Framework (Syafiq Rahman & Farah Nadia, 2024)	Theoretical Model using Literature Review	Quantitative assessment of the benefits and limitations of different testing approaches	This research needs to further analyse the real-life case study on the successful implementation of AI in software testing for SME so that a comprehensive framework can be developed.
A survey on Factors Preventing the Adoption of Auto. Soft. Test. (George Murazvu, Simon Parkinson, 2024)	Empirical Model	Qualitative Techniques	The questionnaire focused on negative aspects of why automation is not used in software testing. Positive aspects of why automation is used in software testing are not identified.
A systematic review of AI based soft. test case optimization (Padmanabhan, 2024)	Theoretical Model using Literature Review	Multivocal Literature Review	Need to further investigate the merging of different AI, their advantages in handling fuzzing problems and adopting fuzzing technology to maximise test cases.

Types of Test Automation Frameworks

Test automation frameworks come in various types, each with its own architecture and unique features. To select the appropriate framework, it is crucial for QA teams to thoroughly understand the features that ensure high-quality testing for their organisation. The first framework is linear automation framework, where the tester records each step, including user inputs, transaction flow, and then plays the script back to conduct the test in a sequential order. The second framework is modular based framework, the software is divided into separate functions or modules, each of which will be tested separately. A test script is created for each part and then combined to build larger tests in a hierarchical approach. The third framework is the library architecture framework, it builds on the modular approach by grouping similar tasks or functions in the software for testing, rather than testing individual modules in isolation. These functions are stored in a library, allowing test scripts to call them as needed. The fourth framework is a data-driven framework, in which test data are separated from script logic and stored externally to an external data source, such as Text Files, CSV files, SQL Tables, or ODBC repositories. The fifth framework is a keyword-driven framework, builds on the data-driven approach by separating test data and script logic while also storing keywords and their associated objects in an external data source. The sixth framework is the hybrid testing framework, which combines multiple frameworks to leverage their strengths and address their weakness, offering a flexible approach to meet automation needs. Figure 2 shows the relationship of different types of automation testing frameworks.

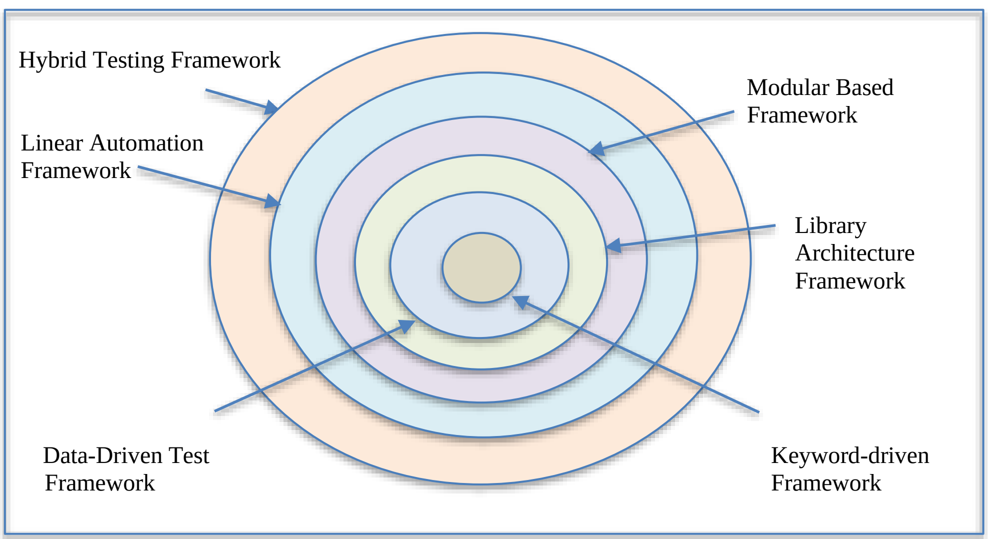

Figure 2. Relationship of Different Types of Automation Testing Framework

Results and Discussion

By creating an artificial intelligence-based test automation framework for assessing software quality traits, this research hopes to provide Malaysian SMEs with an inexpensive way to take advantage of AI's promise in software testing. The framework gives SMEs in Malaysia a competitive edge in a technologically advanced business environment by tackling the issues they encounter and improving scalability, accuracy, and efficiency in their operations. Requirementbased testing, usability testing, development testing, performance testing, and regression testing are among the testing categories covered in the study. Criteria including labor-intensive or time-consuming procedures, tests needing large datasets, and repetitive testing over several releases will all be assessed by the suggested automation framework. To facilitate automated software testing, the AI-based test automation framework combines procedures, methods, and resources. Table 2, shows a comparison of some of the commonly used artificial intelligence tools that are available for software testing.

Table 2. Artificial Intelligence Tools for Software Testing

Feature	**Testim**	**Mabl**	**Functionize**	**Rainforest QA**	**Katalon Studio**
Platform Supported	Web	Web	Web	Web	Web, Mobile, Desktop
Languages Supported	JavaScript, TypeScript	No Coding required (supports JS for custom)	Python, Java, JavaScript, Ruby	No coding required	Java, JavaScript, C#, HTML/CSS
Ease of Setup	Easy	Very Easy	Moderate	Easy	Easy
Ease of Use	Very Easy	Very Easy	Easy	Very Easy	Easy
Writing Test Scripts	Use AI to auto-generate or enhance test scripts	Require little or no coding for creating tests.	Use AI to auto-generate or enhance test scripts	Require little or no coding for creating tests.	Write test scripts, offering both no-code/low-code options and advanced scripting for developers
Testing Categories	Functional testing	Functional testing	Functional testing	Functional testing	Functional Testing, Test Management (Limited)

Conclusion

This paper presents a comparative analysis of different types of test automation frameworks and AI-powered tools. The framework will be designed specifically for SMEs, it will address common challenges related to cost, limited technical resources and simplify crucial processes, including test case creation, bug detection, performance analysis, and reporting. To maintain affordability and accessibility, the framework will rely solely on open-source tools. It is important to note that security testing is beyond the scope of this framework. Nevertheless, by leveraging AI-driven automation and cost-effective solutions, the framework will have the potential to help SMEs improve software quality, reduce testing costs, and overcome traditional barriers to automation adoption, paving the way for more efficient and scalable quality assurance practices.

Acknowledgement

The researcher did not receive any funding for this study, and the results have not been published in any other sources.

References

Abdulwareth, A. J., & Al-Shargabi, A. A. (2021). Toward a multi-criteria framework for selecting software testing tools. *IEEE Access, 9,* 158872–158891. https://doi.org/10.1109/ACCESS.2021.3128071

Aydos, M., Aldan, Ç., Coşkun, E., & Soydan, A. (2022). Security testing of web applications: A systematic mapping of the literature. *Journal of King Saud University - Computer and Information Sciences, 34*(9), 6775–6775. https://doi.org/10.1016/j.jksuci.2021.09.018

Bajjouk, M., Rana, M. E., Ramachandiran, C. R., & Chelliah, S. (2021). Software testing for reliability and quality improvement. *Journal of Applied Technology and Innovation, 5*(2), 40–46. https://jati.sites.apiit.edu.my/files/2021/03/Volume5_Issue2_Paper7_2021.pdf

Garousi, V., Joy, N., & Keleş, A. B. (2024). AI-powered test automation tools: A systematic review and empirical evaluation. *arXiv*. https://arxiv.org/abs/2409.00411

Murazvu, G., Parkinson, S., & Kandjani, S. (2024). A survey on factors preventing the adoption of automated software testing: A principal component analysis approach. *Information*, *15*(2), Article 81. https://doi.org/10.3390/info15020081

Haeggström, M. (2024). *Hands-on use and adaptation of AI in developing and testing software applications* (Doctoral dissertation, Uppsala University). https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-538326

Izzat, S., & Saleem, N. N. (2023). Software testing techniques and tools: A review. *Journal of Education and Science*, *32*(2), 31–40. https://doi.org/10.33899/edusj.2023.137480.1305

Khankhoje, R. (2024). AI in test automation: Overcoming challenges, embracing imperatives. *International Journal on Soft Computing, Artificial Intelligence and Applications, 13*(1), 1–10. https://doi.org/10.5121/ijscai.2024.13101

Padmanabhan, M. (2024). A systematic review of AI-based software test case optimization. *International Research Journal of Multidisciplinary Scope, 5*(4), 847–859. https://doi.org/10.47857/irjms.2024.v05i04.01451

Prakash, M., & Rubin, J. (2024). *Role of generative AI tools (GAITs) in software development life cycle (SDLC)–Waterfall model* [Master's thesis, Visvesvaraya Institute of Technology]. DSpace@MIT. https://dspace.mit.edu/handle/1721.1/154009

Nama, P. (2024). Integrating AI in testing automation: Enhancing test coverage and predictive analysis for improved software quality. *World Journal of Advanced Engineering Technology and Sciences, 13*(1), 769–782. https://doi.org/10.30574/wjaets.2024.13.1.0486

Salahirad, A., Gay, G., & Mohammadi, E. (2023). Mapping the structure and evolution of software testing research over the past three decades. *Journal of Systems and Software, 195,* 111518. https://doi.org/10.1016/j.jss.2022.111518

Sani, B., & Jan, S. (2024). Empirical analysis of widely used website automated testing tools. *EAI Endorsed Transactions on AI and Robotics, 3,* 1–11. https://doi.org/10.4108/airo.7285

Rahman, S., & Farah Nadia. (2024). Pioneering testing technologies: Advancing software quality through innovative methodologies and frameworks. *Journal of Artificial Intelligence and Machine Learning in Management*, *8*(2), 44–70. https://journals.sagescience.org/index.php/jamm/article/view/188

Thakur, D., Mehra, A., Choudhary, R., & Sarker, M. (2023). Generative AI in software engineering: Revolutionizing test case generation and validation techniques. *Iconic Research and Engineering Journals*, *7*(5), 281–293. https://www.irejournals.com/formatedpaper/17051751.pdf

World Quality Report 2023. (2023). *The future up close*. Capgemini, Sogeti, Micro Focus. https://www.opentext.com/assets/documents/en-US/pdf/the-future-up-close-world-quality-report-2023-24-en.pdf