INTI JOURNAL | Vol.2025, Issue 1, No.11
eISSN:2600-7320

AI-Driven Automation in Software Testing: Enabling SME Adoption

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

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

[*]**Email:** i24030456@student.newinti.edu.my[1], waiyie.leong@newinti.edu.my[2*]

Abstract

The rapid evolution of the software industry has positioned Artificial Intelligence (AI) as a game-changer in software testing, enabling Quality Assurance (QA) teams to deliver higher-quality software with greater speed and efficiency. Despite these advantages, many small and medium-sized enterprises (SMEs) are hesitant to adopt AI into their software testing due to financial limitations, time constraints, and lack of technical skill resources. The study aims to address these challenges by proposing a framework that enables SMEs to implement AI-based automation in software testing aligned with their operational requirements. The research methodology combines a planned survey and a literature review to identify the commonly used automation tools and assess their impact on product quality. The ultimate goal is to develop a cost-effective, practical process innovation framework tailored to support Malaysian SMEs in adopting AI for software testing.

Keywords

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

Introduction

Automation in software testing is rapidly evolving and transforming with the growing adoption of artificial intelligence (AI). According to (World Quality Report, 2024), Market Guide for AI-Augmented Software-Testing Tools 2024, 80% of companies will have integrated AI-augmented testing tools into their software engineering processes in 2024, compared to only 15% in 2023. According to (Prathyusha Nama, 2024) companies that integrate AI into their testing processes could see up to a 50% reduction in testing costs and a productivity gain of up to 30%. AI has become a significant system that supports software testing in recent years, providing new approaches to automating the creation of test cases, code analysis, anomaly detection, performance testing, and regression testing, reducing test execution time and widening test coverage. According to (Thakur et al., 2023) AI testing tools have capabilities to generate tests, cross-verify

Submission: 31 December 2024; **Acceptance:** 1 April 2025; **Available online:** April 2025

the outcomes with the expected results, and autonomously monitor the software quality throughout the stages of the development process and with lower manual effort as compared to the conventional methods of testing.

Testing is the process of validating and verifying if a particular system satisfies the original specified requirements. It aims to find defects, mistakes, or missing requirements in software. Stakeholders receive accurate information about the product's quality from the process (Kumar, 2023). Software testing is an essential part of the software development life cycle (SDLC), which is an important process to deliver quality products to the customers and the software able to perform as per the client's requirement (Thakur et al., 2023). During agile SDLC, software requirements may change frequently at the request of stakeholders. Finding adequate time for complete, satisfactory testing to ensure the code is developed correctly is challenging. Incomplete and selective testing increases bugs and issues in the released software, according to (Farahanchi & Yassan, 2023).

The generative artificial intelligence tools (GAITs), powered by advanced machine learning algorithms, can generate new and innovative solutions to complex problems. The continuous advancement of AI technology has paved the way for its integration into software development, bringing forth an era of unparalleled innovation and efficiency, according to (Prakash & Rubin, 2024). According to Muhi Majzoub in (World Quality Report, 2024), the use of emerging technologies in software development and delivery operations is no longer a goal but a necessity. As stated in the (World Quality Report, 2024), the introduction of Gen AI into the Quality Engineering space is still in its early stages, and most organizations leveraging Gen AI are experimenting with various use cases to identify which ones deliver the most significant benefits.

Software test automation tools can be categorized into several types, including Unit Testing Tools, Functional Testing Tools, Code Coverage Tools, Test Management Tools, and Performance Testing Tools. Unit testing allows the developer to detect errors in the early stages by testing individual units of code as soon as they are developed. Functional testing is the process of assessing software's functionality without being aware of its internal workings (Charan et al., 2023). The test cases are derived from the specifications of the software components being evaluated. Code coverage tools are used to assess the extent of software code that is tested by automated tests. These tools measure how many lines, statements, or code blocks are covered by the tests within automated test suites. Test management tools automate various aspects of testing, including test cases, plans, test strategies, test results, and test reports. They enhance project management by creating searchable repositories for test activities, enabling easier maintenance. Performance testing assesses a software’s responsiveness and stability under various conditions and workloads. It evaluates quality attributes such as scalability, reliability, latency, response time, load handling, and resource usage, ensuring adherence to specified performance requirements and validating overall system efficiency.

There are four main goals for this research. The first step is to determine the main obstacles that small and medium-sized businesses (SMEs) encounter when implementing AI-based software testing automation technologies. The second step is to identify the most common automation tools that are often utilised in the sector. Third, the study aims to develop a cost-effective AI-driven testing framework tailored to the specific needs of Malaysian SMEs based on the insights gathered.

Finally, a Focus Group Discussion with industry practitioners will be carried out to validate the proposed framework to ensure its practicality and effectiveness in real-world SME environments.

Methodology

A hybrid methodology is used for this study, incorporating both quantitative and qualitative data collection through surveys and interviews. Data are collected from companies with varying levels of experience in using AI in software testing, including those using AI for more than five years, less than two years, and those still relying on manual testing. The reason for collecting data from companies still using manual software testing is to analyze the factors that are contributing to their reluctance to use automation in software testing. Analysis will be done through their success stories of implementing AI in their software testing. The most common automation tools will be identified, and evaluation of tools that are suitable for different platforms. Based on the survey findings and existing literature review, a framework will be developed that will be cost-effective and efficient for Malaysian SMEs. The new framework will be validated via Focus Group Discussion.

Results and Discussion

This research aims to develop an artificial intelligence-based test automation framework, offering Malaysian SMEs an affordable solution to harness the potential of AI in software testing. By addressing the challenges faced by SMEs in Malaysia, the framework enhances scalability, accuracy, and efficiency in their operations, providing a competitive edge in a technology-driven business landscape. The proposed automation framework will evaluate criteria such as repetitive testing across multiple releases, tests requiring extensive datasets, and labor-intensive or time-consuming processes. Figure 1, shows an Artificial Intelligence-based test automation framework.

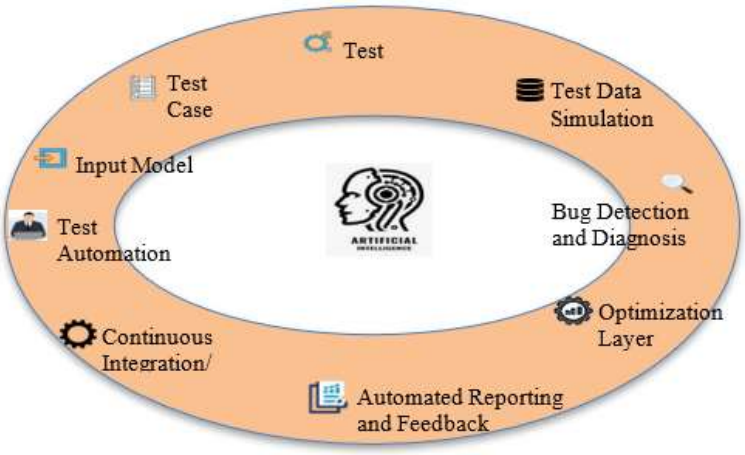

Figure 1. Artificial Intelligence-Based Test Automation Framework

As illustrated in Figure 1, the framework is designed to optimize the software testing lifecycle by leveraging AI technologies. The process begins with test automation, whereby the

Quality Assurance (QA) team gathers user requirements and specifications to create the input module. Using AI-powered software testing tools, comprehensive test cases are automatically generated. The Test Engine executes these test cases, monitors the execution process, and collects real-time feedback. The test execution utilizes test data stored in the test database to ensure accuracy. Reports on the test results are generated for the Quality Assurance team and developer in resolving the issues. Performance testing is conducted to evaluate the software's scalability, reliability, and overall performance. The framework also performs regression testing to ensure compatibility with continuous integration/continuous deployment (CI/CD) pipelines.

Figure 2, illustrates a node-based workflow that visualizes the AI-driven software testing process. Machine learning models enable AI to learn from historical data (Leong, 2024a; 2024b). The iterative process incorporates real-time data back to the models to continuously improve prediction and risk analysis. The workflow features AI-generated test cases, which are automatically created and executed using test automation tools, significantly reducing manual effort. AI assesses potential risks within the software, prioritizing critical areas for testing and predicting defects before they occur (Leong 2024c). Automated test generation converts insights into actionable test cases that can be run through automation tools. Performance testing is conducted to ensure the software performs efficiently as per client requirements. Continuous monitoring and real-time testing ensure that testing adapts dynamically to changes in the system. Test results are inserted back into the machine learning model, refining predictions and improving test coverage over time. Reports and Dashboards will be used to monitor defects for immediate action. These interconnected nodes in the workflow collaborate to deliver high-quality software.

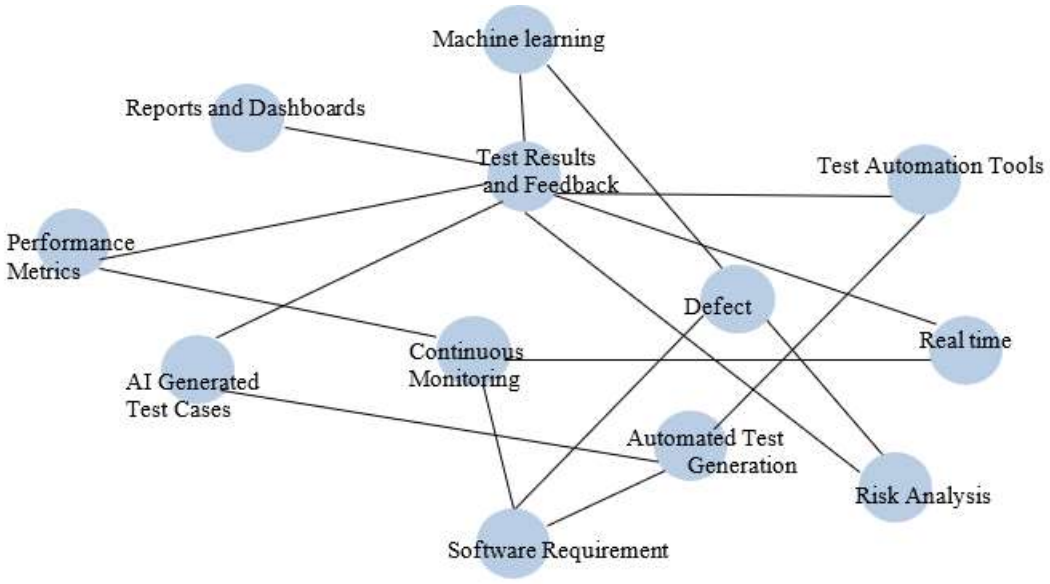

Figure 2. AI-driven software testing process

The study (World Quality Report, 2024), reveals a growing adoption of AI, particularly Generative AI, in the world of Quality Engineering. Figure 3 shows the outcome from a survey that identifies three key cost benefits of test automation in the organization. According to the findings, 61% of Chief Information Officers (CIO) believe improved testing coverage enhances

confidence in IT systems. Furthermore, 60% of senior management reported that higher automation levels have enhanced fast feature delivery, and 59% of CIOs indicated that automation speeds up the route-to-live, thereby expanding their customer base. Over half, 58% of the respondents highlighted that automation reduces manual effort, leading to lower operating costs, and 53% of the respondents mentioned that automation has led to fewer defects, which enhanced user experience. The survey conducted clearly shows that test automation significantly impacts time, cost, and quality, and there is a growing consensus on the need for increased automation adoption.

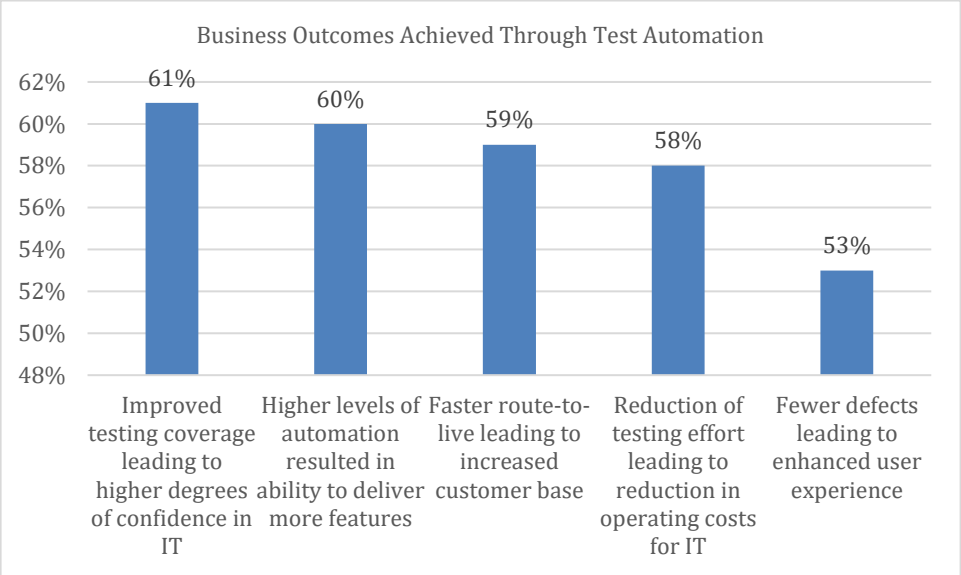

Figure 3. Business Outcomes Achieved Through Test Automation
(Source: World Quality Report 2024-2025)

Conclusion

In conclusion, the proposed AI-based testing framework offers a revolutionary solution to companies' challenges in adopting automated testing tools. By leveraging advanced AI capabilities, the framework streamlines the entire testing process—from test case generation and execution to bug detection, performance analysis, and reporting for the quality assurance team. It ensures accuracy, scalability, and alignment with software specifications, reducing manual effort and improving overall efficiency. Its seamless integration with CI/CD pipelines addresses the demand for rapid and reliable software delivery, making it highly adaptable to modern development practices. The research aims to assist SMEs to streamline their best fit tools for their specific needs in terms of budget, skill sets, development platform and requirement.

Acknowledgement

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

References

1. Charan, K., Karthikeya, N., Narasimharao, P. B., Devi, S. A., Abhilash, V., & Srinivas, P. V. V. S. (2023). Effective Code1 Testing Strategies in DevOps: A Comprehensive Study of Techniques and Tools for Ensuring Code Quality and Reliability. In *2023 4th International Conference on Electronics and Sustainable Communication Systems, ICESC 2023 - Proceedings* (pp. 302–309). https://doi.org/10.1109/ICESC57686.2023.10193275
2. Farahanchi, H., & Yassan, R. (2023). *A natural language processing model to improve the software testing process under an agile methodology praxis directed by Professorial Lecturer of Engineering Management and Systems Engineering*. [Master's thesis, George Washington University]. ScholarSpace. https://scholarspace.library.gwu.edu/etd/z890rv14r
3. Kumar, S. (2023). Reviewing software testing models and optimization techniques: An analysis of efficiency and advancement needs. *Journal of Computers, Mechanical and Management*, *2*(1), 32–46. https://doi.org/10.57159/gadl.jcmm.2.1.23041
4. 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
5. 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
6. 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*. https://www.irejournals.com/paper-details/1705175
7. World Quality Report, 2024. (2024). *World Quality Report*. www.worldqualityreport.com