Barriers and Enablers of AI Adoption in Software Testing: A Secondary Study

Katja Karhu and Jussi Kasurinen
Department of Software Engineering
LUT University
Lahti, Finland
e-mail: {katja.karhu | jussi.kasurinen}@lut.fi

***Abstract*—It seems that AI adoption in software testing is not as straightforward as promoted by the hype surrounding AI: there are a lot of expectations, but the reported practical implementations are still relatively rare. In this survey paper, we investigated the reasons behind the slow AI adoption in software testing by qualitatively analyzing recent empirical studies with industry context. In our work, we classified the barriers and enablers from the earlier studies into six categories: management, processes, human resources, technology, data and external. The main approach to AI adoption in software testing in the industry still seems to be investigation and experimentation in individual organizations without industry-wide reference implementations or standards. A major barrier for AI adoption in software testing is the lack of perceived usefulness or produced value. More research and empirical evidence of successful AI adoption in software testing is needed.**

Keywords-software testing; artificial intelligence; technology adoption; qualitative research; reflexive thematic analysis.

I. Introduction

In the industry, the interest in AI in software testing is high and growing. It is seen as a potential competitive advantage: companies that do not successfully utilize AI in software testing will lose in the competition [1]. However, the practical implementations and the AI adoption rate seem to be lagging behind [2][3].

In Perforce's [2] 2024 industry survey, 48 percent of respondents indicated they were interested in AI but have not yet started any initiatives, and only 11 percent were already implementing AI techniques in software testing. Interest in AI is still growing a year later, in Perforce's newest 2025 industry survey [3], over 75 percent of survey respondents identified AI-driven testing as a pivotal component in their strategy for 2025. But actual adoption rate is still behind, with only 16 percent of respondents reporting on having adopted AI in testing [3].

On the side of academia, the research on AI in software testing has been quite extensive [4][5]. However, Nguyen at al [6][7] found in a study conducted in 2023, that most of the existing studies on AI in software quality assurance are "experimental studies and thus do not take into consideration the industrial context". They further state that "how GenAI models deal with real-world software quality issues remains a mystery" [6][7]. King et al [8] had similar findings in 2019: "only a few of these works are backed by real-world case studies, or result in industrial tools and methods". Since some time has passed when the studies mentioned were performed, and advances in especially generative AI and large language models have been made, we wanted to explore the current state of empirical studies on AI in software testing with a strictly industry context. To our knowledge, literature surveys with this specific scope have not been conducted before.

Our first paper on AI adoption in software testing, based on the same dataset of literature, was about how AI is utilized on software testing and what are the expectations related to it [9]. There, we focused on the actual and potential use cases for AI in software testing, and their actual and expected benefits [9].

In this survey paper, we continue the reflexive thematic analysis of the literature from the point of view of barriers and enablers of AI adoption in software testing. Overall, our goal in this study is to explore, why AI adoption rate in software testing is still quite low, and what could be done about it via identifying the barriers and enablers. Our research questions are:

- RQ1: What are the issues that prevent or hinder AI adoption in software testing?
- RQ2: What are the enablers behind successful AI adoption in software testing?

The paper is structured as follows. In Section 2, we describe the data collection and research methods and process. In Section 3, we present the results of the qualitative analysis. Section 4 contains the discussion, where we further reflect on the findings. Finally, the conclusion and future work are presented in Section 5.

II. Methods

We utilized systematic mapping study, as described by Petersen, Vakkalanka and Kuzniarz [10] to identify earlier studies on AI adoption in software testing. Then, we qualitatively analyzed the papers found via the systematic mapping study. Our primary data analysis approach in this study is reflexive thematic analysis. Braun and Clarke [11][12] define thematic analysis as a flexible qualitative analysis method, or more appropriately, a family of methods, for observing themes within data. The overall research process is documented in higher detail in the "sister paper" of this study [9].

A. Data Collection

We performed a systematic mapping study, and found 17 papers that fit our inclusion and exclusion criteria [9]. Our goal was to find recent (year 2020 or later) original empirical papers on AI in software testing where the data had been collected from testers or other QA experts. The databases we

 ISBN: 978-1-68558-296-8

TABLE I. NUMBER OF PAPERS PER YEAR

Year	Papers
2020	2 [13][14]
2021	0
2022	2 [1][15]
2023	4 [16]–[19]
2024	10 [20]–[28]
Total:	17

used were Scopus and Google Scholar, because they are known to include papers from a wide selection of different fields.

Over half of the papers (10) we found had been published in 2024 (see Table I). Out of the 17 papers, nine were peer-reviewed, six were theses and two were other grey literature. The reason, why we included theses and other grey literature in our study, was that they contained rich and detailed data collected from experts, making them well suited for qualitative analysis.

B. Reflexive Thematic Analysis

In this study, we used reflexive thematic analysis, a non-positivist approach [12], as our data analysis method. A theme is a concept that captures important patterned information and insights about the data, related to the research question [11]

We followed the phases of thematic analysis defined by Braun and Clarke [11]:

- Phase 1: familiarizing yourself with your data
- Phase 2: generating initial codes
- Phase 3: searching for themes
- Phase 4: reviewing themes
- Phase 5: defining and naming themes
- Phase 6: producing the report

In reality, the process was more iterative, where especially the phases from three to six were repeated several times. We followed an inductive approach in our analysis, and our goal was identify interesting themes in the papers. Eventually, the large number of identified themes resulted in splitting the reporting into different papers, as too many themes in one paper resulted in a very incoherent and long report. In our earlier paper, we focused on how AI is utilized in software testing, analyzing the actual and expected use cases and benefits, and the discrepancy between the expectations and reality of AI adoption in software testing [9].

In this study, we wanted to further investigate the reasons why AI adoption in software testing seems to be quite low. Therefore, we selected the barriers and enablers as our primary theme. We also tried to identify the differences between actual barriers and enablers, and expected barriers and enablers, but that proved too complicated, especially with the earlier literature, where a lot of the context was missing.

III. RESULTS

We grouped the barriers and enablers identified from the studies into six categories (see Table II): management, processes, human resources, tools, data and external. The category here describes the "source" of the barrier, or the level the barrier could be resolved. Barriers and enablers in different categories can also affect each other. For example, outsourcing can be one way of resolving the barrier of AI skill gap. It is worth noting, that the barriers of adoption are not inherently "bad" and enablers "good". For example, strict IT policies or data privacy and security issues as barriers are essential in cases where the developed software is safety-critical. Enabling the AI adoption by loosening the IT policies in this will most likely result in unwanted side-effects.

The management category describes the barriers and enablers related, for example, to the organization's finances, priorities, strategy and personnel management. The process category contains the barriers and enablers related to the daily operations within the organizations, such as policies, communication, software development processes, etc. The enablers and barriers in the human resources -category include employee-level topics, such as skills and feelings. In the tools category, we have technological barriers and enablers. The data category contains barriers and enablers related to data. And finally, in the external category, we have items that impact AI adoption in testing, but come from outside the organization, for example, societal, business ecosystem, or industry level barriers and enablers.

The obvious elephant in the room is the perceived *lack of usefulness or produced value* of AI in software testing. In Purovesi's [26] investigation of AI adoption in the test automation context, interviewees had observed that the value produced by AI is still minimal. Also, Hossain et al [27] found that in companies there was uncertainty about the usefulness of AI testing. Some felt that there was a lack of concrete estimates about the time-saving of AI assisted test automation, as the evidence was limited [26]. In the study by Amalfitano, Coppola, Distante and Ricca [20], the respondents "pointed out that while there are numerous general-purpose tools available in the domain of Large Language Models (LLM), their potential for GUI-based testing tasks remains unproven". The earlier studies reported benefits from AI adoption in software testing, but in some cases, it was difficult to quantify them [26], or respondents felt that the speed or efficiency improved only a little [24][26]. In addition, AI adoption may cause additional work that may undo the benefits: it takes more time than it saves [26]. For example, creating test cases with AI is easy, but maintenance is not: making changes to, or finetuning, AI-generated test cases takes a lot of time and effort [24].

Significant investments are required in AI adoption: especially *investments in technology and skill development* were seen as important enablers. Hossain et al [27] found that especially for software development organization that have no previous experience with AI systems, implementing AI can be both costly and time-consuming. Costs include, for example, hiring more staff or consultants, training of personnel, and infrastructure and computational resources [27]. The development of trustworthy AI systems can take a long time, months or even years, because of the experimental and iterative approach to development [27].

TABLE II. BARRIERS AND ENABLERS PER CATEGORY FROM EARLIER LITERATURE

Category	Barriers	Enablers
Management	Lack of usefulness/produced value [1][20][24][26][27] Requires significant investments [1][27] Risk aversion [1][21][27] Lack of time and resources [1][21][24][25][27]	Marketing AI benefits [1][26] Leadership support [14][16][17][25] Investments in technology [14][16][17][26] Investments in skill development [26] Outsourcing [27] Hiring new employees [27]
Processes	Incompatibility with current processes [1][13][21] Strict IT policies [25] Poor internal communication [25]	Evaluation of current processes [1][28] Change management [25] AI roadmap [27]
Human resources	AI skill gap [1][20][21][24][26][27] Lack of trust in AI [13][21][23]–[25] Resistance to change [1][25]	Personnel training [16][24]–[27] Internal communication [21][25] Collaborative experimentation and research [1] Guidelines for working with AI [20]–[22][25]
Tools	Difficulties in finding and selecting tools [20][24] Lack of transparency [13][21][28] Incompatibility with legacy systems [1][21][26] Poor usability of tools [1] Unreliability (e.g., hallucination and bias) [21][23] Tool pricing [20] Lack of domain knowledge [20]	Explainable AI (XAI) [13][21][28] Monitoring and reviewing [21] Building test automation first [1] AI tool documentation [4] Company's internal AI tools [25] Open-source AI tools [20] Formal screening process for AI tools [25]
Data	Lack of training data [1][13][20][21][23][25]–[28] Data privacy and security issues [24][27]	Purposefully collecting data for training [1][26] Creating training datasets [20][26] Tools for data cleaning and pre-processing [27] Reliable data sources [27] Proper training of AI with high quality data [20][27][28]
External	Lack of reference implementations or standards [1][21]	Education system (e.g., university level) [27] Collaboration with other organizations [27] Certifications [16][21][26]

Because of the experimental nature, someone has to make the initial commitment and investments to the development of AI-based testing solutions, but without reference implementations, it can be difficult to get organizations to commit to, or even try, AI adoption in testing [1]. Ahven [1] investigated AI adoption in testing in an IT consulting company, where their customers were not willing to commit because of the *lack of reference implementations*. And on the other hand, the consulting company was not willing to develop AI solutions internally and take the financial risk of trying something new that might result in a failure [1]. To get around this problem, the interviewee's in Ahven's [1] mentioned, that the company had started *marketing* communications (videos, webinars) to create excitement about AI-assisted testing within customers. However, on the other side of the marketing coin, aggressive marketing, unrealistic promises and a "hype peak" related AI in testing were observed [26].

Khan et al [21] found that especially small companies doing software development "may be *risk-averse* towards adopting new technologies, including AI-based software testing techniques, due to the fear of potential failures or increased costs". In addition to the financial risks, AI technologies include risks, such as risks related to security and privacy. Purovesi [26] found that, even though the interest in using AI in test automation, customers wanted to carefully consider the security and privacy before making the commitment. Uncertainty about risks was one of the reasons Hossain et al [27] also found, that reduced the willingness in organizations to commit to AI adoption in testing. In safety-critical scenarios, the uncertainty of results can be unacceptable [20]. In addition, people unaware of AI risks are a risk. Even though, in most of the studies, the interviewees were knowledgeable about the risks related to AI, a complete lack of risk awareness was also observed: some employees did not see any risks in AI utilization in software testing [25].

Time and resources are needed in many aspects of AI adoption in software testing, such as designing, building and training AI models [27], implementing AI systems for testing [1], skill development [25]. In addition, new computational resources are needed for the AI infrastructure [27]. AI adoption means time away from daily work, which may be difficult to organize and cause delays in testing work [27]. Skill development may be hindered if testers do not have working time allocated to learning about new topics that are not directly related to their current work [25].

AI adoption in software testing also requires changes in *current processes* and ways of working. Evaluating current ways of working and how things could be done differently were seen as important enablers, but it is also often blocked by lack of time [1]. Also, *resistance to change* can hinder adoption: if existing processes are working quite well already, it might be difficult to convince people to think things differently

[1]. Laine [25] suggests that *change management* efforts are therefore needed to enable successful AI adoption. Hossain et al [27] suggest creating an *AI roadmap*, and evaluating, where AI would fit in, as well as monitoring the performance indicators and milestones over time [27].

AI skill gap as a barrier was highlighted in several studies. AI as a term can contain a variety of technologies requiring specialized skills that is not usually present in a testing organization [26][27]. Verifying and validating AI model's accuracy after training requires skills, such as statistical analysis and data visualization [27]. Prompting, while seeming deceptively simple, can require significant effort and a trial-and-error approach, "which requires a fine-tuning of the prompts used and implies the possibility of wrong results of the testing activities" [20]. In order to bridge the AI skill gap, the organization must invest in skill development via, for example, personnel training, hiring new employees, developing guidelines (e.g., for prompting and creating training datasets) [20][21][27]. In one company, a study group ("future testing research unit") had been created, where testing specialists *experimented and researched* new tools and new ways of working, and presented the results to the organization [1]. Another option to resolve the AI skill gap is *hiring new personnel* [27].

The *lack of trust in AI* seems to be closely related to the *unreliability of AI tools*. In a study by Adu [23], and interviewee summarized the problem of the underlying reliability issue in LLMs: *"My main concern is that LLMs are not capable of thinking and do not really "understand" the prompts nor the content they are producing. They are rather content generators which output the statistically like response given some input. As such, the problem with using them for testing related tasks is that there is no guarantee that they are doing what you asked them to do."* [23].

Due to their inherent nature, LLMs are not therefore ideal for testing tasks that require reliability or determinism. It comes down to selecting the right AI tools for the right tasks. In addition, human supervision of AI via *monitoring and reviewing* was seen as crucial [21]. Khan et al summarized that "rule-compliant processes, monitoring systems, and external oversight contribute to reliability" [21].

Earlier it was mentioned, that marketing, and communicating the benefits of AI to customers are important. Same seems to go for the *internal communication*. Laine [25] states that "when introducing a new AI tool, the communication should emphasize the benefits, especially to the employees themselves". Poor internal communication can manifest as lack of knowledge about, e.g., the internal training materials and AI tools that are available in the company [25]. Khan et al [21] highlight also the transparent and ethical *organization culture* and communication in order to build long-term trust to AI, as well as self-regulation and self-imposed AI standards.

Clear communication is also important if, for example, utilizing public AI tools is forbidden by the IT policy due to privacy and security reasons. In addition to communication, the overall *leadership support* was seen as an important factor in engaging employees [14][16][17] and was seen as contributor to a successful adoption [25].

From the technical side, *incompatibility with legacy systems* and code was raised as a potential problem in AI adoption [21][26]. On the other hand, from the data point of view, old and complex systems were more suitable for leveraging AI, because of the large amounts of data available for AI model training [26].

The *lack of transparency* in AI models and tools can cause a lack of trust in AI, since it is difficult to understand and trust their decisions [13][20][21]. With commercial AI testing tools the black-box nature can hinder also the ability to fine-tune the tools effectively [20]. In addition, commercial tools were *difficult for testers to evaluate*, as they require subscriptions, which may lead to testers giving up on the tool evaluation completely [20].

Amalfitano, Coppola, Distante and Ricca [20] suggest *open-source AI tools* as way of increasing the transparency and explainability of AI systems. Khan et al [21] state that "*explainable AI (XAI)* is crucial for building trust, but challenges exist in achieving transparency". Solutions suggested for enabling and increasing explainability were: implementing parallel algorithms [21], knowledge distillation, saliency mapping, and symbolic reasoning [13], as well as visualization [28]

Lack of training data, and especially the *lack of high quality training* data was also a major barrier. The quality of training data directly impacts the reliability of the AI system [27]. However, potential training data was not collected systematically or it was discarded as useless [1]. Training data collection also raises *privacy and security concerns* [27]. Overall, careless handling of sensitive data in AI tools may result in compromised security [23]. Another issue was the bias in AI systems, caused by the training data. Khan et al [21] suggested using benchmark datasets in addressing the bias [21]. Overall, collecting, labeling and *cleaning* training data was seen as a challenging and high effort activity, which also requires time and investments [13][21][27].

IV. DISCUSSION

Our research questions were: "what are the issues that prevent or hinder AI adoption in software testing" and "what are the enablers behind successful AI adoption in software testing". We went through earlier empirical studies on AI in software testing with industry context, and looked for the barriers and enablers of AI adoption. We found that AI adoption in software testing is not only a technological issue, as we classified the barriers and enablers reported in the earlier studies into six categories: management, processes, human resources, technology, data and external.

A very significant barrier for AI adoption in software testing was a lack of usefulness or produced value, combined with the lack of reference implementations. The potential reasons we identified based on the analysis of the earlier studies were:

- The early phase of adoption: the approach to AI adoption is still exploratory and investigative, as there are yet no industry-wide practices or standards

- The nature of some AI tools: for example, LLMs are not suitable for testing tasks that require reliability and determinism.

In our earlier study of the same literature dataset [9] we found that there were potential use cases for AI in software testing, such as test case generation, code analysis, and intelligent test automation, but the reported actual implementations and observed benefits were limited. Many implementations were still on the proof-of-concept level [9].

The lack of usefulness or produced value, and the lack of reference implementations, seem to be also reflected in grey literature on AI in software testing. Ricca, Marchetto and Stocco [29] have performed a multi-year analysis of grey literature in test automation, and concluded that "the nature of these sources limits to suggestions on how to use AI for TA, without presenting concrete evidence of its practical application or usage details".

For the lack of usefulness or produced value we did not identify many direct enablers. Collaborative experimentation and research, and collaboration with other organizations, could be ways for identifying useful reference AI solutions for software testing. Most of the enablers were focused on resourcing, building trust in AI/tackling resistance, skill development, and documentation.

Simmler and Frischknecht [30] have proposed a taxonomy for evaluating human-machine collaboration, that could be utilized also in software testing context to evaluate suitable AI solutions for different testing tasks, as well as the resources needed for monitoring these solutions. They define two dimensions: level of automation and level of autonomy. In short, the higher the level of automation, the less humans have control over the system, and the higher the level of autonomy is, the more the transparency decreases and it becomes more difficult to trace the machine's actions [30]. The more autonomy the AI-system has, the more important human monitoring becomes, and the higher the level of automation is, the more reliable the system needs to be [30].

It is worth noting, that the interest in AI seems to have increased significantly [2][3], even despite the lack of perceived value or usefulness. What makes companies want to invest in new technologies that do not yet have a proven track record? Gulzar and Smolander [31] provide an explanation to this: they have established that there are various motivations for new technology adoption, which can have both positive and negative effects. These motivations include [31]:

- Market dynamics - staying one step ahead, technology as a competitive edge
- Internal imperatives - goal setting by management, strategy
- Technological advancement - innovations
- Social influence - fear of missing out, hype, enthusiasm, collective feelings
- Economic considerations - reducing costs, return-of-investment (ROI)
- Operational and strategic improvements - increased efficiency, productivity, scalability

As can be seen, there are more motivations to technology adoption than just operational improvement and technological advancement. The other motivations, such as market dynamics and social influence, can weigh in more than the proven benefits of the new technology.

Monitoring is also one of the ways to tackle the barriers of lack of transparency and the lack of trust in AI. Soomro, Fan, Sohu, Soomro and Shaikh [32] have found that "negative perceptions of AI can slow down its adoption in various sectors". Open and honest internal communication on AI were highlighted by Laine [25] and Khan et at [21] as a way of addressing the lack of trust. Another issue causing lack of trust in AI are the data privacy and security issues. Kinney, Anastasiadou, Naranjo-Zolotov and Santos [33] summarized that the loss of privacy due to data needs of AI systems are a significant factor in the rejection of AI technology".

The marketing of AI benefits, whether to customers or in company's internal communication, should be based on evidence. In Purovesi's [26] study interviewees had observed aggressive marketing, unrealistic promises and a "hype peak" [26]. On the other hand, there were also reports of resistance to change [25][26]. If the lack of trust in AI and resistance is based on legitimate concerns, such as lack of produced value and usefulness, or security and data privacy problems, it should not be addressed by only trying to convince the audience of AI benefits. In summary, we need more research and empirical evidence of successful AI implementations in software testing to back up the message of AI benefits.

A. Limitations of the study

The studies we analyzed were from 2020-2024, so it is possible, that there have been new developments in AI adoption in software testing after their data collection phases. It can also be, that we have incomplete knowledge of AI adoption in software testing, as it may be undocumented. Companies may be unwilling to reveal their experiences of AI-assisted testing, in order to maintain business secrets or competitive edge, or even due to failed adoption attempts.

In this study we did not cover the impact of legislation and regulation to AI adoption, which could be viewed as both barriers and enablers. It was noted in the earlier studies, that for some domains, such as healthcare and banking, the laws and regulations of handling confidential data are stricter [27]. In these cases, the laws and regulations are acting as a barrier, and for a good reason. Laws and regulations were also seen as a way of increasing trust in AI, making them also an enabler. Some experts also raised the issue of problems in regulation, as it was viewed as outdated, and needed to be extended to cover AI use [27][28]. However, there were not enough details to analyze the impact of laws and legislation to AI adoption in software testing in higher detail, as this is a complex phenomenon. We also address additional limitations in our earlier study [9].

V. Conclusion and Future Work

In order to investigate the reasons behind low AI adoption rates in software testing, we performed a reflexive thematic analysis of 17 earlier empirical studies on AI adoption in software testing. We found that AI adoption is software testing is still suffering from lack of reference implementations and standards, as well as perceived usefulness and value. The limited empirical evidence on the benefits of AI in testing, combined with significant investments and resources required are one explanation for the low rates of AI adoption in software testing.

Future research is needed in evaluating the usefulness of different AI technologies (such as LLMs) in different testing tasks. We need more empirical evidence and detailed descriptions of successful AI adoptions, as well as lessons learned from unsuccessful AI adoption projects. We plan to continue our research by conducting interviews in software development organizations in order to further identify the reasons behind the slow AI adoption, as well as identifying the success factors of AI adoption in software testing.

References

[1] H. Ahven, "Utilization of artificial intelligence in software quality assurance," Finnish, M.S. thesis, University of Jyväskylä, 2022.

[2] Perforce, *The 2024 State of Continuous Testing*, 2024. [Online]. Available: https://www.perfecto.io/resources/state-of-continuous-testing, Accessed: 2025-02-14.

[3] Perforce, *The 2025 State of Continuous Testing*, Feb. 2025. [Online]. Available: https://www.blazemeter.com/resources/2025-state-of-continuous-testing-report, Accessed: 2025-02-18.

[4] D. Amalfitano, S. Faralli, J. C. R. Hauck, S. Matalonga, and D. Distante, "Artificial Intelligence Applied to Software Testing: A Tertiary Study," *ACM Computing Surveys*, vol. 56, no. 3, Oct. 2023, ISSN: 15577341. DOI: 10.1145/3616372.

[5] M. Hossain and H. Chen, "Application of machine learning on software quality assurance and testing: A chronological survey," *International Journal of Computers and their Applications*, vol. 29, no. 3, pp. 150–157, 2022.

[6] A. Nguyen-Duc *et al.*, "Generative Artificial Intelligence for Software Engineering A Research Agenda," preprint, Oct. 2023. DOI: 10.48550/arXiv.2310.18648.

[7] A. Nguyen-Duc *et al.*, "Generative Artificial Intelligence for Software Engineering A Research Agenda," *Software: Practice and Exprience*, pp. 1–38, Jun. 2025. DOI: 10.1002/spe.70005.

[8] T. M. King *et al.*, "Ai for testing today and tomorrow: Industry perspectives," in *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, 2019, pp. 81–88. DOI: 10.1109/AITest.2019.000-3.

[9] K. Karhu, J. Kasurinen, and K. Smolander, "Expectations vs Reality – A Secondary Study on AI Adoption in Software Testing," unpublished, Apr. 2025. DOI: 10.48550/arXiv.2504.04921.

[10] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, Aug. 2015, ISSN: 09505849. DOI: 10.1016/j.infsof.2015.03.007.

[11] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006, ISSN: 14780887. DOI: 10.1191/1478088706qp063oa.

[12] V. Braun and V. Clarke, "Toward good practice in thematic analysis: Avoiding common problems and be(com)ing a knowing researcher," *International Journal of Transgender Health*, vol. 24, no. 1, pp. 1–6, 2023, ISSN: 26895277. DOI: 10.1080/26895269.2022.2129597.

[13] M. Gutiérrez, "AI-Powered Software Engineering: Integrating Advanced Techniques for Optimal Development," *International Journal of Engineering and Techniques*, vol. 2020, no. 6, pp. 1–9, 2020, ISSN: 2395-1303. DOI: 10.5281/zen.

[14] M. Barenkamp, J. Rebstadt, and O. Thomas, "Applications of AI in classical software engineering," *AI Perspectives*, vol. 2, no. 1, pp. 1–15, Dec. 2020. DOI: 10.1186/s42467-020-00005-4.

[15] S. Ramchand, S. Shaikh, and I. Alam, "Role of Artificial Intelligence in Software Quality Assurance," in *Intelligent Systems and Applications*, K. Arai, Ed., Cham: Springer International Publishing, 2022, pp. 125–136, ISBN: 978-3-030-82196-8.

[16] J. Bhuvana, V. Ranjan, and N. Bhaduariya, "Integration of AI in the Realm of Software Development," in *2023 International Conference on Advances in Computation, Communication and Information Technology, ICAICCIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 974–978, ISBN: 9798350344387. DOI: 10.1109/ICAICCIT60255.2023.10465893.

[17] R. Kumar, V. Naveen, P. Kumar Illa, S. Pachar, and P. Patil, "The Current State of Software Engineering Employing Methods Derived from Artificial Intelligence and Outstanding Challenges," in *1st IEEE International Conference on Innovations in High Speed Communication and Signal Processing, IHCSP 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 105–108, ISBN: 9798350345957. DOI: 10.1109/IHCSP56702.2023.10127112.

[18] S. Qazi, I. Memon, B. Q. Hashmi, M. S. Memon, and E. A. Qazi, "Software Quality Assurance Using Artificial Intelligence Techniques: A Survey of the Software Industry of Pakistan," *Journal of Hunan University Natural Sciences*, vol. 50, no. 12, pp. 1–9, 2023. DOI: 10.55463/issn.1674-2974.50.12.1.

[19] A. I. Amarasekara, "Challenges of outdated software test automation tools in the sri lankan it industry and the proposal of suitable ml strategies for a sustainable quality assurance process," M.S. thesis, Uppsala University, 2023.

[20] D. Amalfitano, R. Coppola, D. Distante, and F. Ricca, "AI in GUI-Based Software Testing: Insights from a Survey with Industrial Practitioners," in *Quality of Information and Communications Technology*, A. Bertolino, J. Pascoal Faria, P. Lago, and L. Semini, Eds., Cham: Springer Nature Switzerland, 2024, pp. 328–343, ISBN: 978-3-031-70245-7.

[21] S. A. Khan *et al.*, "AI-Based Software Testing," in *Proceedings of World Conference on Information Systems for Business Management*, A. Iglesias, J. Shin, B. Patel, and A. Joshi, Eds., Singapore: Springer Nature Singapore, 2024, pp. 323–334, ISBN: 978-981-99-8346-9.

[22] R. Santos, I. Santos, C. Magalhaes, and R. De Souza Santos, "Are We Testing or Being Tested? Exploring the Practical Applications of Large Language Models in Software Testing," in *Proceedings - 2024 IEEE Conference on Software Testing, Verification and Validation, ICST 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 353–360, ISBN: 9798350308181. DOI: 10.1109/ICST60714.2024.00039.

[23] G. K. Adu, "Artificial Intelligence in Software Testing: Test scenario and case generation with an AI model (gpt-3.5-turbo) using Prompt engineering, Fine-tuning and Retrieval

augmented generation techniques," M.S. thesis, University of Eastern Finland, 2024.

[24] H. Jauhiainen, *Artificial Intelligence In Software Testing*, 2024, Bachelor's thesis. HAMK.

[25] A. Laine, "A Change Management Approach to Incorporating AI into Software Testing," M.S. thesis, LUT University, 2024.

[26] R. Purovesi, "Test Automation and AI," M.S. thesis, LUT University, 2024.

[27] T. Hossain *et al.*, "AI based solutions for Software Testing in Bangladeshi software companies," unpublished, Jan. 2024. DOI: 10.13140/RG.2.2.17371.95526.

[28] L. Layman and R. Vetter, "Generative Artificial Intelligence and the Future of Software Testing," *Computer*, vol. 57, no. 1, pp. 27–32, Jan. 2024, ISSN: 15580814. DOI: 10.1109/MC.2023.3306998.

[29] F. Ricca, A. Marchetto, and A. Stocco, "A multi-year grey literature review on ai-assisted test automation," *Information and Software Technology*, vol. 186, p. 107 799, 2025, ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2025.107799.

[30] M. Simmler and R. Frischknecht, "A taxonomy of humanmachine collaboration: capturing automation and technical autonomy," *AI and Society*, vol. 36, no. 1, pp. 239–250, Mar. 2021, ISSN: 14355655. DOI: 10.1007/S00146-020-01004-Z/FIGURES/1.

[31] M. Gulzar and K. Smolander, "Explaining the motivational drivers in technology adoption: Triggers & Consequences," in *Proceedings of the 58th Hawaii International Conference on System Sciences*, 2025, ISBN: 9780998133188.

[32] S. Soomro, M. Fan, J. M. Sohu, S. Soomro, and S. N. Shaikh, "AI adoption: a bridge or a barrier? The moderating role of organizational support in the path toward employee well-being," *Kybernetes*, vol. ahead-of-print, no. ahead-of-print, 2024, ISSN: 0368492X. DOI: 10.1108/K-07-2024-1889/FULL/XML.

[33] M. Kinney, M. Anastasiadou, M. Naranjo-Zolotov, and V. Santos, "Expectation management in AI: A framework for understanding stakeholder trust and acceptance of artificial intelligence systems," *Heliyon*, vol. 10, no. 7, e28562, Apr. 2024, ISSN: 24058440. DOI: 10.1016/J.HELIYON.2024.E28562/ASSET/4DE3D17E-E79C-41C4-9450-0E69B166AC58/MAIN.ASSETS/GR1.JPG.