The impact of AI on SQM: Mapping the current state

Lidija Vincekovič[1], Tina Beranič[1]

[1]Faculty of Electrical Engineering and Computer Science - University of Maribor, Koroška cesta 46, Maribor, Slovenia

Abstract

One of the big potentials to optimise and improve Software Quality Assurance (SQA) activities and their results is by using emerging technologies like Artificial Intelligence (AI). During our research into the application of AI in SQA, we identified inconsistencies in the definitions of SQA and its related concepts as well as ambiguity in their interrelationships. This paper thus presents a literature review on the application of AI within the domain of Software Quality Management (SQM). The paper categorizes AI applications across SQM categories: SQA, Software Quality Control (SQC), Software Quality Planning (SQP), and Software Process Improvement (SPI). We identified 24 papers in selected digital libraries and categorized them by SQM area, AI subset application, and topics that the papers addressed. Results show a dominant trend in the use of Machine Learning (ML) and the most frequently addressed topics are test generation, test execution, and fault/defect prediction. SQP and SPI, including their specific topics, are rarely presented in AI-related research. We propose research opportunities for a more comprehensive application of AI across the SQM domain.

Keywords

software quality management, SQM, software quality assurance, SQA, SQC, SQP, SPI, Artificial Intelligence, AI

1. Introduction

Software Quality Assurance (SQA) is one of the most important parts of the Software Development Lifecycle. Time is becoming a critical factor in releasing applications that must be tested in detail to achieve compliance with more and more complex requirements. In order to be ahead of competitors, organizations must address the challenges of the growing demands on the software market. One of the big potentials to optimise and improve SQA activities and their results is by using emerging technologies like Artificial Intelligence (AI). AI can be used in many areas related to SQA and its usage could result in reducing time to market, optimizing costs of many SQA activities, and consequently more stakeholders' satisfaction [1].

While researching the area of how AI is used for SQA we noticed a number of different definitions of SQA and its related terms (e.g. software quality management, software quality planning, software testing, software validation) and their mutual relationships. Based on different definitions and understandings, we decided to explore the broader picture of SQA, which is Software Quality Management, according to Mistrik et al. [2]. For the purpose of this paper, we have adopted the classification from [2], who classify SQA as a category of Software Quality Management (SQM). SQM is according to this classification comprised of three basic categories: already mentioned SQA, software quality planning (SQP), and software quality control (SQC), and an additional separate category: software process improvement (SPI).

With this paper, our goal is to gain better insight in:

- RQ1: what is the coverage of research papers related to AI for each SQM area (SQA, SQP, SQC, SPI),
- RQ2: what is the most common topic of identified papers categorized per each category of SQM.

By answering these questions, we seek to understand the trend of research and interests in the domain of SQM and the potential gaps for future research, which means a broader research area than the narrower one of SQA.

SQAMIA 2025: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 10–12, 2025, Maribor, Slovenia

lidija.vincekovic@um.si (L. Vincekovič); tina.beranic@um.si (T. Beranič)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-4077/paper18.pdf

The usage of AI for SQA has been a topic of increasing academic interest, particularly within the domains of software testing and defect prediction. Several systematic literature reviews (SLRs) and mapping studies have been conducted to assess the state of the art (SotA) in these focused areas. Authors of [3] presented a systematic review of machine learning methods in software testing, outlining various models used for test case generation, prioritization, and optimization. Similarly, authors of [4] conducted a broad review of AI applications in software testing, focusing primarily on machine learning and deep learning techniques for improving testing effectiveness and efficiency. Other notable contributions, such as [5], have surveyed datasets and techniques for defect prediction using AI, emphasizing performance comparisons and methodological trends.

While these works provide valuable insights into AI's role in software testing and defect prediction, they generally address SQA as a single concept, without distinguishing it from other important aspects of SQM (SQC, SQP, and SPI. Moreover, few of these studies classify the reviewed works by AI subset (e.g., machine learning, natural language processing) or by topic (e.g., test execution, root cause analysis, code smell detection), which limits their usefulness for identifying research gaps in under-explored SQM categories. To the best of our knowledge, no previous literature review has provided such a multi-dimensional mapping of AI applications across the complete SQM spectrum. Our work is therefore complementary to existing reviews and aims to broaden the research focus from a testing-centric view of quality assurance to a more holistic understanding of SQM.

This paper is structured as follows: Section 2 is an overview of SQM, its categories, and AI. In Section 3, we have presented our research methodology and an overview of the paper review on how AI is applied to SQM's categories and coverage of research topics in identified papers. Section 4 is a summary of the review articles we have done that present AI applied to SQM categories and the topics of each article per SQM category. Lastly, Section 5 is a conclusion and a proposal for future work.

2. Overview of Software Quality Management, its categories and AI

According to Mistrik et al. [2], SQM is the collection of all processes that ensure that software products, services, and life cycle process implementations meet organizational software quality objectives and achieve stakeholder satisfaction. SQM comprises three basic categories: SQP, SQA, and SQC. Very often, as in the Software Engineering Body of Knowledge [6], SPI is described as a separate category of SQM, but could be part of any of the first three categories. The visual representation of the SQM categories is shown in Figure 1.

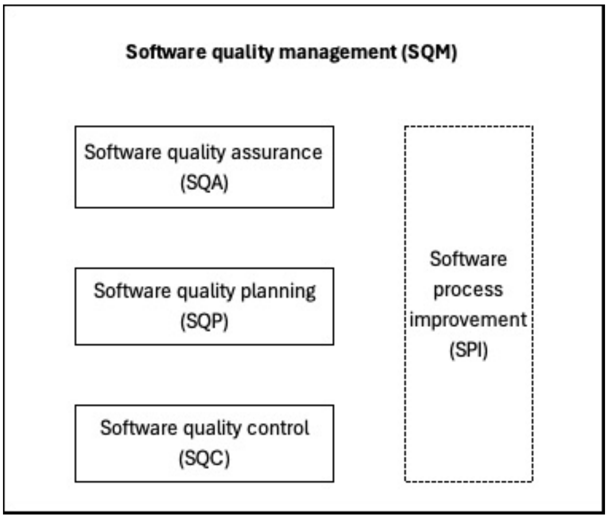

Figure 1: SQM categories [2]

There are many definitions of **SQA** in different literature. The main accepted definition is the ISO/IEC/IEEE 24765:2017 standard definition, which defines SQA as "a set of activities that assess adherence to and the adequacy of the software processes used to develop and modify software products. SQA also determines the degree to which the desired results from software quality control are being obtained". According to this definition, SQC is a subset of SQA, but for the purpose of this paper, we will consider it as a parallel category as shown in Figure 1.

SQC activities ensure that project artefacts (e.g. documentation, design, code) are checked for quality before they are delivered. This means that an examination is made on whether artefacts comply with standards established for the project, including functional and non-functional requirements and constraints. SQC activities are for example technical reviews, code inspection, and testing.

SQP can be explained as the project commitment to respect the selected and applicable set of standards, regulations, procedures, and tools during the development life cycle. SQP defines the quality goals to be achieved, expected risks and risk management, and the estimation of the effort and schedule of software quality activities.

SPI activities aim to improve process quality, including effectiveness and efficiency, with the objective to improve the overall software quality. Usually, an SPI project starts by mapping the organization's existing processes to a process model that is then used for assessing the existing processes. Based on the results of the assessment, an SPI activity aims to achieve process improvement.

According to AI Act [7] **AI system** means a machine-based system that is designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, and that, for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments.

The key pillars of AI are: Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP), expert systems, and others. AI covers many areas like: data analysis, prediction, decision making, intelligent systems, and many others [1].

3. Overview of AI applied to SQM categories

3.1. Research Methodology

We performed a literature review methodology to identify papers related to any of the SQM areas and artificial intelligence at the same time. We searched in Web of Science, ScienceDirect and IEEE digital libraries by combining keywords, their acronyms and Boolean operators: "software quality management", "SQM", "software quality assurance", "SQA", "software quality control", "SQC", "software quality planning", "SQP", "software process improvement", "SPI", "Artificial intelligence" and "AI".

Inclusion criteria for the selection of papers were the following: the paper must address one of the SQM areas, the paper must be accessible electronically, the paper must be written in English, the paper must be published between 2014 and 2025, the paper must be published in computer science literature, and its subject is to explore the use of AI in any of the SQM categories. An important note is also that the relationship of AI and SQM areas was that AI was to be applied for any of the SQM areas and not that any SQM areas was used for AI (e.g., SQA for AI). Our methodology resulted in 24 papers that met our criteria and are presented in this paper.

We classified the selected papers after thoroughly examining them in the following order:

1. Classification per SQM area: When classifying papers, we looked closely at the definitions of each paper, whether directly or indirectly, and listed it under a category that matched identically or to a category which matched the definition the closest to the definition presented in this paper. If any paper covered multiple SQM areas matching the stated definition, we classified it with all of the identified SQM categories.
2. Classification per AI subset: We classified results per AI subsets as they were mentioned in papers. Identified categories were: AI, explainable AI (XAI), Neural Networks (NN), NLP, ML, Large Language Model (LLM), DL. The general AI category was used if no specific AI subset was addressed, but only AI in general. We have presented NN as a separate category, even though it is as a subset of ML, and LLM as a separate category, even though it is a subset of NLP, because they

were explicitly addressed in some papers. If any paper covered multiple AI subsets, we classified it with all of the identified AI subsets.

3. Classification per topic: Topics for answering RQ2 were found based on the results, meaning whenever a new topic was discovered in one of the papers, we included it on the list of topics and analysed all selected papers whether they covered either of the topics on the list. If any paper covered multiple identified topics, we classified it with all of the identified topics. Identified categories were: test generation, test execution, fault/defect predictions, root-cause (defect) analysis, automatic debugging, code smell detection, SPI diagnostics.

3.2. AI applied to SQM's categories

The landscape of AI approaches for all SQM is diverse and dynamic. Looking at the absolute numbers, and as illustrated in Figure 2, ML emerges as the most commonly applied AI subset (17 papers), followed by the Explainable Artificial Intelligence (6 papers). Others related to DL (4 papers), LLM (3 papers), NLP (2 papers) and NN (2 papers), and 1 paper was addressing AI approaches in general.

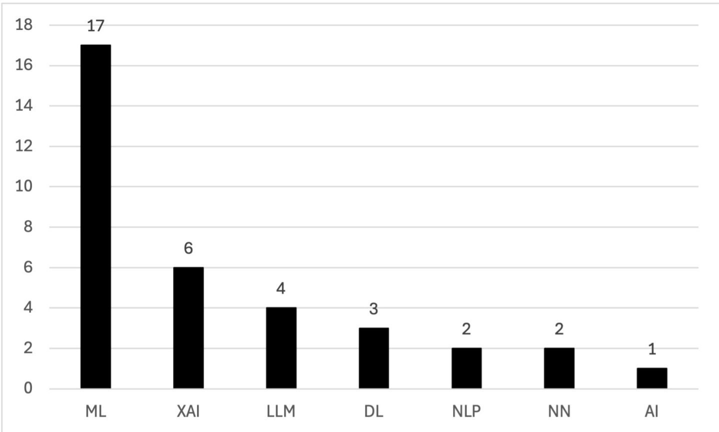

Figure 2: Number of papers per AI

As a result of our research 17 out of 24 identified papers that met our criteria were related to **SQA** - see Figure 3. These papers covered SQA in general and spanned from covering topics related to test generation, test execution, fault detection, code smell detection, root cause analysis and automatic debugging. Papers were classified under **SQC** if this term was directly mentioned or if the paper covered testing in general, as testing, code inspection, and technical reviews are SQC activities according to the classification we are basing our paper on. There were 14 papers that were classified as papers related to SQC. The majority of these papers were also classified as related to SQA. Most of the SQC classified papers addressed ML, which highlights the evolving nature of ML applications in software testing, indicating a great potential for improvement of software quality and reliability. As the authors of [8] have stated, ML is the foundation of AI-driven testing, as software testing can gain from predictive analytics, anomaly detection, and automated decision-making by utilizing ML algorithms, which results in increasing the efficacy and efficiency of testing activities. On the other hand, the reviewed studies reveal also many challenges for using AI for testing, which were summarized by [3] as the need for empirical research, scalability, test coverage, test input generation, failure management, interoperability, test Oracle, accuracy, trust, security, and access to high-quality training datasets, all being potential for future research. The only paper that we identified that was classified as **SQP** was paper [9]. Authors have proposed four types of guidance to support SQA planning and an AI-Driven SQAPlanner approach for generating four types of guidance and their associated risk thresholds, adding also an evaluated visualization of this SQAPlanner approach.

Paper that was classified as the only paper in SQP category of SQM was also classified as **SPI** related paper. Besides this one, out of 24 identified papers, there was only one more that we could classify

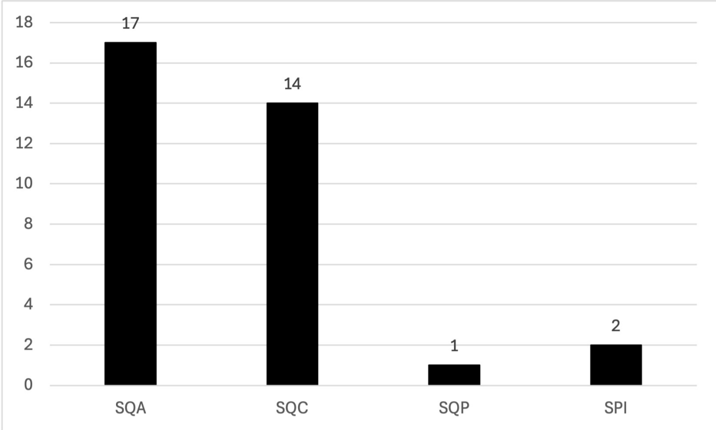

Figure 3: Number of papers per SQM

as SPI paper, and that was paper [10]. Authors have performed a Systematic Literature Review (SLR) to research solutions for the SPI Diagnostic process. 14 solutions were identified, out of which 5 of them implement some AI technique. The main conclusion of the paper was that AI can empower SPI Diagnostic software tools and propose to further investigate the potential of AI techniques in their support of the SPI Diagnostic.

3.3. Topics covered per SQM's categories

Looking at the entire selection of papers that met our criteria most of the topics were related to software testing (test generation, test execution), followed by fault/defect predictions and few papers with topics on automatic debugging, code smell detection and SPI diagnostics as presented in Table 1.

Topic	Frequency
Test generation	16
Test execution	13
Fault/defect predictions	10
Root-cause (defect) analysis	5
Automatic debugging	2
Code smell detection	2
SPI diagnostics	1

Table 1
Frequency of topics in identified papers

As software testing is, per our selected definition for the purpose of this paper, an SQC activity, papers that addressed this topic were classified under SQC. A lot of them were also at the same time classified as SQA due to its content matching the definition of SQA as set in section 2.2. of this paper. This shows a growing trend towards leveraging AI to automate these activities, which could significantly improve the efficiency and effectiveness of SQA, which was also emphasized in [11].
The next most frequent topic was fault/defect prediction, which demonstrates an increase in effort to optimise SQC efforts. Less frequent topics of selected studies were root cause/defect analysis, quality assessment (testing of quality metrics), code smell detection, and automatic debugging, but are still noteworthy as they indicate potential to proactively address software quality challenges. The only paper that was classified as addressing the topic of SPI diagnostic was a paper that was classified in SPI category of SQM [10]. The paper that covered the most identified topics was paper [12]. There were two identified papers (see Figure 5) that wrote about automatic debugging, these were paper [12] mentioned in the previous paragraph and paper [13]. As it can be seen from Figure 4, the only paper [9] that was classified to SQP area was also, per its content, classified to SQA and SPI. The topic it

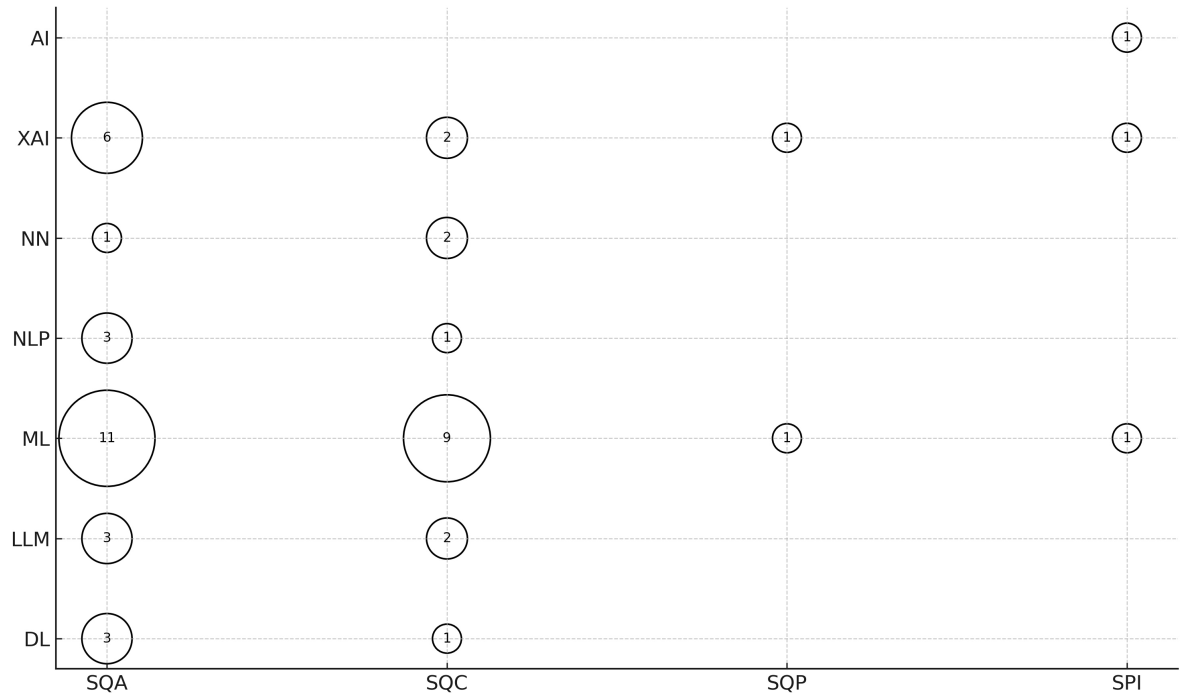

Figure 4: Number of papers per AI applied to SQM categories

addressed was software quality planning and fault/defect prediction.
Code smell detection was also a specific topic, addressed by two identified papers: [14] and [15]. [14] explores how XAI can help developers understand and prioritize code smells and shows that focusing on relevant features can improve the clarity and usefulness of XAI in identifying critical code smells. Paper [15] shows that AI tools can improve software quality by identifying code-smell issues early, helping developers write cleaner and more maintainable code.

4. Results and Discussion

With this paper and RQ1, we wanted to gain better insight into the coverage of research papers related to AI for each SQM area as defined at the beginning of this paper. Summary of applied AI for each SQM area as classified from the identified 24 papers is presented in Table 2. Rows represent identified AI subset or AI in general, while columns represent SQM category. Cells include identified papers that are addressing an AI subset or more of them related to the respective SQM category or more of them.

AI subset	SQA	SQC	SQP	SPI
AI				[10]
DL	[11], [15], [4]	[4]		
LLM	[11], [16], [17]	[16], [18]		
ML	[11], [8], [19], [14], [4], [12], [20], [21], [22], [9], [17], [1]	[11] [3], [13], [14], [4], [23], [24], [20], [5], [1]	[9]	[9]
NLP	[25], [17]	[25]		
NN	[26]	[23], [26]		
XAI	[19], [14], [21], [27], [20], [9]	[14], [20]	[9]	[9]

Table 2
Mapping of papers per AI and SQM category

Most papers we identified are related to SQA, followed closely by SQC, and the usage of ML is predominant in these papers. Several factors may have contributed to this fact, including recent breakthroughs in ML techniques. SQP and SPI are the least covered SQM categories, as we identified only two papers related to these two categories, out of which one referred to the application

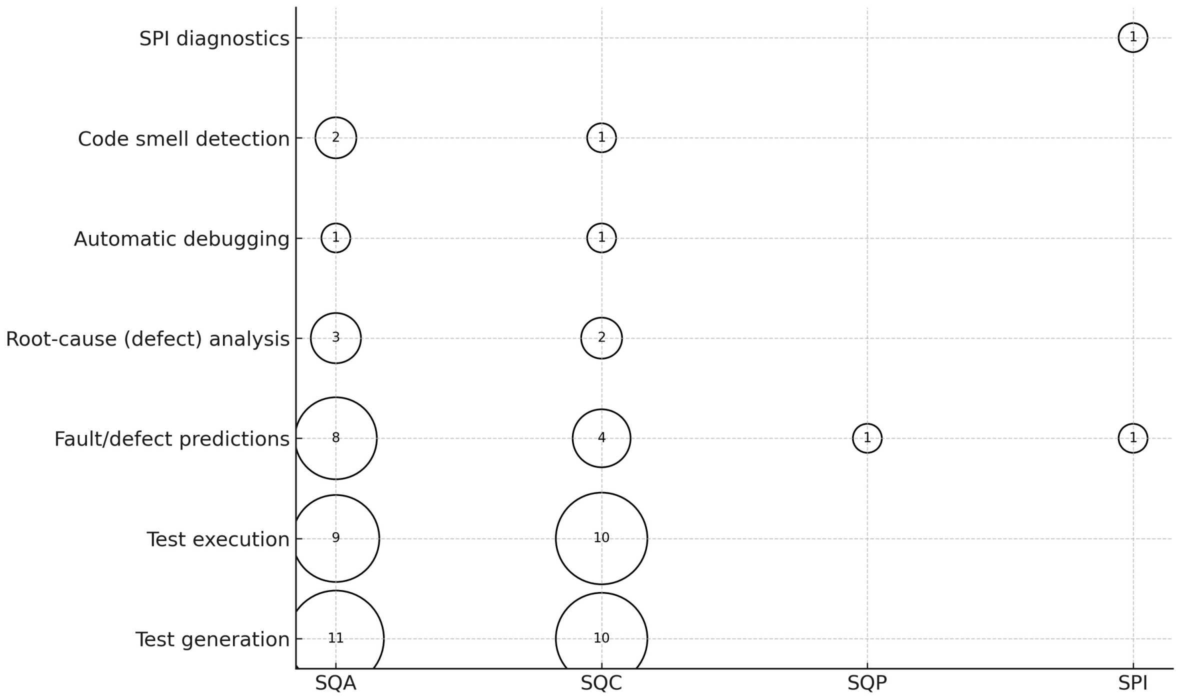

Figure 5: Number of papers per topics and SQM categories

of AI in general, and the second one addressed the application of ML and XAI. The reasons for under-representation could be the different understanding of related concepts, lack of tooling, or less industry interest. These categories typically involve more abstract, strategic activities, which may explain the lower focus in existing AI research. Nonetheless, we believe these areas are critical for long-term quality improvements and represent promising directions for future work.

With RQ2 we aimed at gaining insight in what is the most common topic of identified papers categorized per each category of SQM. Summary of addressed topics for each SQM area, as classified from the identified 24 papers, is presented in Table 3. Rows represent identified topics, while columns represent the SQM category. Cells include identified papers that are addressing identified topic or more of them related to the respective SQM category or more of them.

Topic	SQA	SQC	SQP	SPI
Test generation	[11], [8], [19], [4], [16], [12], [25], [22], [26], [17], [1]	[11], [3], [13], [4], [23], [24], [18], [25], [26], [1]		
Test execution	[11], [19], [4], [16], [12], [25], [26], [17], [1]	[11], [3], [13], [4], [16], [24], [18], [25], [26], [1]		
Fault/defect predictions	[19], [27], [12], [20], [21], [22], [9], [1]	[24], [20], [5], [1]	[9]	[9]
Root-cause (defect) analysis	[19], [12], [17]	[13], [24]		
Automatic debugging	[12]	[13]		
Code smell detection	[14], [15]	[14]		
SPI diagnostics				[10]

Table 3
Mapping of papers per topic and SQM category

The topics that prevailed for SQA and SQC related papers were test generation and test execution, followed closely with fault/defect prediction. Several topics were addressed less commonly, which shows smaller interest or smaller potential for the application of AI. These less represented topics include root-cause analysis, code smell detection, automatic debugging, and SPI diagnostics. Their complexity or lack of labelled data may contribute to the lower research volume, though their value for improving software quality is significant.

In addition, we briefly analysed the time distribution of the selected articles to better understand the evolution of AI's role in SQM over time. Although a complete trend analysis is beyond the scope of this paper, we observe that most articles have been published in the last 4-5 years (2021–2025). This shows that interest in using AI for software quality activities has increased significantly recently.

5. Conclusion and Future Work

For the purpose of this paper, we have adopted the classification of SQM with the SQA, SQC, SQP, and SPI categories to evaluate the landscape of AI integration in the SQM domain, also researching the topics most addressed. According to the results of our study, there is a lot of research in the area of AI application to SQA and SQC, and a research gap related to AI application for SQP and SPI. There is a strong research focus on the application of ML, DL, and LLM, mostly in the context of test generation, test execution, and fault/defect prediction. This reflects the current maturity of AI tools in automating and optimizing operational testing processes.

There is a lack of research on topics such as cause analysis, automatic debugging, code smell detection, and especially software quality planning and software process improvement. Analysis of the details of selected papers shows that there are different understandings of each of these areas, thus we allow the possibility that areas of planning and process improvement are interesting for research, but are not perceived as a separate area of research, but as a part of SQA, being a broader term. The reason for the lesser focus on SQP and SPI may come from their perceived abstraction or indirect impact on product-level quality, but these domains are crucial for long-term strategic improvements in SQM. This suggests a need to shift some research focus toward areas that support early planning and long-term process improvement.

Addressing these under-represented topics offers promising directions for future research. Potential areas include the development of AI-supported tools for quality goal setting, risk-aware planning, and continuous process optimization. As AI technologies evolve, new opportunities will emerge for integrating AI systems into the entire SQM domain.

Although we have provided an initial observation indicating an increase in relevant publications after 2021, it would be interesting to further research on how the focus areas, AI techniques, and SQM categories have evolved over time. Results could offer an insightful view of how the AI supported SQM landscape is maturing and what are the future opportunities.

While the classification of AI subsets used in this paper was carefully derived from the terminology and categorization present in the reviewed papers, it is important to point out that this process was conducted internally by the authors. However, to strengthen the reliability and objectivity of the categorization, especially in cases where AI techniques overlap or are not precisely described, involving external domain experts in future classification efforts would be very valuable and important.

Acknowledgments

The authors acknowledge financial support from the Slovenian Research and Innovation Agency (Research Core Funding No. P2-0057).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4 for figures 4 and 5 in order to generate bubble charts from provided data. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] H. Hourani, A. M. Hammad, M. Lafi, The impact of artificial intelligence on software testing, 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology

(JEEIT) (2019) 565–570. URL: https://api.semanticscholar.org/CorpusID:159042474.
[2] I. Mistrik, R. Soley, N. Ali, J. Grundy, B. Tekinerdogan, Software Quality Assurance: In Large Scale and Complex Software-intensive Systems, Morgan Kaufmann, 2015. URL: https://books.google.si/books?id=NVaZBQAAQBAJ.
[3] S. Ajorloo, A. Jamarani, M. Kashfi, M. H. Kashani, A. Najafizadeh, A systematic review of machine learning methods in software testing, 2024. doi:10.1016/j.asoc.2024.111805.
[4] M. Islam, F. Khan, S. Alam, M. Hasan, Artificial intelligence in software testing: A systematic review, in: IEEE Region 10 Annual International Conference, Proceedings/TENCON, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 524–529. doi:10.1109/TENCON58879.2023.10322349.
[5] J. Pachouly, S. Ahirrao, K. Kotecha, G. Selvachandran, A. Abraham, A systematic literature review on software defect prediction using artificial intelligence: Datasets, data validation methods, approaches, and tools, 2022. doi:10.1016/j.engappai.2022.104773.
[6] H. Washizaki (Ed.), Guide to the Software Engineering Body of Knowledge - SWEBOK V4.0, IEEE Computer Society, 2024. URL: www.swebok.org.
[7] European Union, Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), *Official Journal of the European Union*, L 1689, 12 July 2024, 2024. Available at https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.
[8] C. Deming, M. A. Khair, S. R. Mallipeddi, A. Varghese, Software testing in the era of ai: Leveraging machine learning and automation for efficient quality assurance, Asian Journal of Applied Science and Engineering 10 (2021) 66–76. doi:10.18034/ajase.v10i1.88.
[9] D. Rajapaksha, C. Tantithamthavorn, J. Jiarpakdee, C. Bergmeir, J. Grundy, W. Buntine, Sqaplanner: Generating data-informed software quality improvement plans, IEEE Transactions on Software Engineering 48 (2022) 2814–2835. doi:10.1109/TSE.2021.3070559.
[10] M. Ecar, J. P. S. D. Silva, N. Amorim, E. M. Rodrigues, F. Basso, T. G. Solda, Software process improvement diagnostic: A snowballing systematic literature review, in: Proceedings - 2020 46th Latin American Computing Conference, CLEI 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 156–164. doi:10.1109/CLEI52000.2020.00025.
[11] A. Ahammad, M. E. Bajta, M. Radgui, Automated software testing using machine learning: A systematic mapping study, in: 2024 10th International Conference on Optimization and Applications (ICOA), 2024, pp. 1–6. doi:10.1109/ICOA62581.2024.10754031.
[12] M. Kalech, R. Stern, Ai for software quality assurance blue sky ideas talk, in: 34th AAAI Conference on Artificial Intelligence / 32nd Innovative Applications of Artificial Intelligence Conference / 10th AAAI Symposium on Educational Advances in Artificial Intelligence, volume 34, 2020, pp. 13529–13533.
[13] A. Elmishali, R. Stern, M. Kalech, An artificial intelligence paradigm for troubleshooting software bugs, Engineering Applications of Artificial Intelligence 69 (2018) 147–156. doi:10.1016/j.engappai.2017.12.011.
[14] Z. Huang, H. Yu, G. Fan, Z. Shao, M. Li, Y. Liang, Aligning xai explanations with software developers' expectations: A case study with code smell prioritization, Expert Systems with Applications 238 (2024). doi:10.1016/j.eswa.2023.121640.
[15] I. Ali, S. S. H. Rizvi, S. H. Adil, Enhancing software quality with ai: A transformer-based approach for code smell detection, Applied Sciences (Switzerland) 15 (2025). doi:10.3390/app15084559.
[16] M. Jiri, B. Emese, P. Medlen, Leveraging large language models for python unit test, in: Proceedings - 6th IEEE International Conference on Artificial Intelligence Testing, AITest 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 95–100. doi:10.1109/AITest62860.2024.00020.
[17] Y. Yao, J. Wang, Y. Hu, L. Wang, Y. Zhou, J. Chen, X. Gai, Z. Wang, W. Liu, Bugblitz-ai: An intelligent qa assistant, in: Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, IEEE Computer Society, 2024, pp. 57–63. doi:10.1109/ICSESS62520.2024.10719045.
[18] Y. Li, P. Liu, H. Wang, J. Chu, W. E. Wong, Evaluating large language models for software testing,

Computer Standards and Interfaces 93 (2025). doi:10.1016/j.csi.2024.103942.
[19] L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, Causal reasoning in software quality assurance: A systematic review, Information and Software Technology 178 (2025) 107599. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950584924002040. doi:10.1016/j.infsof.2024.107599.
[20] G. Lee, S. U. J. Lee, An empirical comparison of model-agnostic techniques for defect prediction models, in: Proceedings - 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2023, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 179–189. doi:10.1109/SANER56733.2023.00026.
[21] M. Ali, T. Mazhar, A. Al-Rasheed, T. Shahzad, Y. Y. Ghadi, M. A. Khan, Enhancing software defect prediction: a framework with improved feature selection and ensemble machine learning, PeerJ Computer Science 10 (2024). doi:10.7717/peerj-cs.1860.
[22] K. Phung, E. Ogunshile, M. Aydin, A novel software fault prediction approach to predict error-type proneness in the java programs using stream x-machine and machine learning, in: Proceedings - 2021 9th International Conference in Software Engineering Research and Innovation, CONISOFT 2021, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 168–179. doi:10.1109/CONISOFT52520.2021.00032.
[23] S. Ji, Q. Chen, P. Zhang, Neural network based test case generation for data-flow oriented testing, in: Proceedings - 2019 IEEE International Conference on Artificial Intelligence Testing, AITest 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 35–36. doi:10.1109/AITest.2019.00-11.
[24] N. Klimov, Using ai and machine learning in qa testing (????). URL: https://asrjetsjournal.org/index.php/American_Scientific_Journal/index.
[25] Y. Liu, Natural language processing technology based on artificial intelligence in software testing, in: 2024 3rd International Conference on Artificial Intelligence and Computer Information Technology, AICIT 2024, Institute of Electrical and Electronics Engineers Inc., 2024. doi:10.1109/AICIT62434.2024.10730603.
[26] Y. Zhang, New approaches to automated software testing based on artificial intelligence, in: 2024 5th International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 806–810. doi:10.1109/ICAICE63571.2024.10863866.
[27] J. Jiarpakdee, C. K. Tantithamthavorn, J. Grundy, Practitioners' perceptions of the goals and visual explanations of defect prediction models, in: Proceedings - 2021 IEEE/ACM 18th International Conference on Mining Software Repositories, MSR 2021, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 432–443. doi:10.1109/MSR52588.2021.00055.