


Open Access Article

 <https://doi.org/10.55463/issn.1674-2974.50.12.1>

Software Quality Assurance Using Artificial Intelligence Techniques: A Survey of the Software Industry of Pakistan

Sheereen Qazi^{1*}, Irfana Memon², Shahzana Memon³, Beenish Qureshi Hashmi⁴, M. Suleman Memon⁵, Qazi Ejaz Ali⁶

¹ PhD Scholar, Department of IT, QUEST Nawab Shah, Pakistan

² Associate Professor, Department of CSE, QUEST Nawab Shah, Pakistan

³ Assistant Professor, Department of CSE, QUEST Nawab Shah, Pakistan

⁴ Lecturer, Department of CSE, QUEST Nawab Shah, Pakistan

⁵ Professor, Department of CSE, QUEST Nawab Shah, Pakistan

⁶ Department of Computer Science, University of Peshawar, Pakistan

* Corresponding author: kazi.shireen2k10@gmail.com

Received: September 17, 2023 / Revised: October 10, 2023 / Accepted: November 9, 2023 / Published: December 29, 2023

Abstract: The software industry is quality-based and paramount for market-oriented software products. In the software industry, quality assurance (SQA) plays the key role, and is also known as software testing. The testing strategy can include a few steps to consider, such as ownership, unscripted tests, documentation and reportage, and test frequency. Testing provides software quality for the proposed software product and fulfills customer satisfaction. Software product testing and delivery is more suitable than unverified software consistency, expectedness, and non-exceptional resources. A few important types of software testing are unit, integration, system, sanity, smoke, interface, regression, and beta testing. Artificial intelligence approaches will enhance the quality of applications through new approaches to testing and regression testing by describing the impact factor of AI SQA tools. It will also aid in developing steep-quality, less insecure, and safer software. AI can help manage complex digital realms, such as dynamically decentralized networks, surveillance, robotics, and digital services of the next era (often grouped under the term “Next Generation Internet”). Outputs can help promote work on specific AI-based domains that may be required to implement modern digital platform smart activities that involve near interrelationships involving innovations, semblance, and implementations in quality assurance.

Keywords: artificial intelligence, software quality assurance, software development cycle.

使用人工智能技术的软件质量保证：巴基斯坦软件行业调查

摘要：软件产业以质量为基础，以市场为导向的软件产品至关重要。在软件行业中，质量保证（质量保证局）起着关键作用，也称为软件测试。测试策略可以包括几个需要考虑的步骤，例如所有权、无脚本测试、文档和报告文学以及测试频率。测试为提议的软件产品提供软件质量并满足客户满意度。软件产品测试和交付比未经验证的软件一致性、预期性和非异常资源更合适。一些重要的软件测试类型是单元测试、集成测试、系统测试、健全性测

试、冒烟测试、界面测试、回归测试和贝塔测试。人工智能方法将通过新的测试和回归测试方法，通过描述人工智能质量保证局工具的影响因子来提高应用程序的质量。它还将有助于开发高质量、不安全性较低且更安全的软件。人工智能可以帮助管理复杂的数字领域，例如动态分散的网络、监控、机器人和下一代数字服务（通常归为“下一代互联网”术语）。产出可以帮助促进基于人工智能的特定领域的工作，这些领域可能是实施现代数字平台智能活动所必需的，这些活动涉及涉及质量保证创新、外观和实施的密切相互关系。

关键词：人工智能、软件质量保证、软件开发周期。

1. Introduction

1.1. Quality Assurance

The industry is based on quality, which is crucial for the assurance of any product. In the software industry, quality assurance plays the key role of software products. Quality assurance is known as testing [1]. Hence, in the software industry, SQA is software testing. The testing strategy can include a few steps to consider, such as ownership, unscripted tests, documentation and reportage, and test frequency. The testing methods normally implemented by the software industry are manual testing and automated testing [2]. SQA is defined as “investigation of software from a quality perspective or service developed by the software team”. SQA provides comprehensive risk analysis for software products to provide a customer-independent view of quality satisfaction. It ensures that software products are of high quality and fulfill industry norms.

1.2. Importance of Software Quality Assurance

Testing provides software quality for software products and provides customer satisfaction. Tested and delivered software product is more suitable than unapproved software. A few important types of software testing are unit, integration, system, sanity, smoke, interface, regression, and beta testing.

The quality characteristics cover many phases depicted in Figure 1.



Fig. 1 AI system dimensions and SQA (Developed by the authors)

Non-functional testing involved in the software industry includes performance, load, stress, volume, security, compatibility, install, and recovery testing. Testing with software parameters is not prone to variability with respect to loss and theft of data [3]. At the enterprise level, business application daily activity is dependent on the software they use in the development cycle. The lack of testing and verification can cause great losses for business stockholders.

1.3. Quality Assurance Approaches

- *Stress Testing:* Exposing intrinsic flaws will implement and then use results to enhance the commodity and gain a higher quality.

- *Statistical Control:* Objective and subjective data analyses are used to measure accurate data and monitor them against trait heterogeneity.

- *Total Quality Management:* Quality entails commodity that depends on the contributing constituencies, some permanent and manageable will be tested.

- *Models and Standards:* Maintaining international standards with general competency criteria.

- *Company Quality:* Tests carried out by the directing board to establish a method for enhancing efficiency. This is accomplished via controls, job management, process, efficiency, knowledge, abilities and expertise, competence, convincing, and facilities.

1.4. Problem Statement

SQA is a very critical phase in the software development industry [3]. Several tools were introduced to deal with software quality assurance. Most software houses in Pakistan are unaware of these modern tools. Artificial intelligence-based SQA tools result in better improvement in software development [4]. The software development cycle needs quick and smart development process in which an AI-based model resolves SQA issues [5]. The latest AI-based techniques can help save time, cost, and quality assurance activities. In this regard, it is important to highlight the following issues and determine the factors that answer the following reasons:

- Why AI-based tool implementation is important for SQA.
- Are software houses in Pakistan aware of AI-based SQA?
- How SQA is maintained in software houses in Pakistan?

1.5. Research Objective

This study was conducted to focus on the following objectives:

- Highlighting the importance of SQA in the software industry by applying AI-based techniques;
- Surveying the types of techniques used by software houses in Pakistan;
- Providing detailed review and survey of AI-based techniques for SQA in Pakistan.

1.6. Research Design

The paper consists of several parts, as shown in Figure 2.

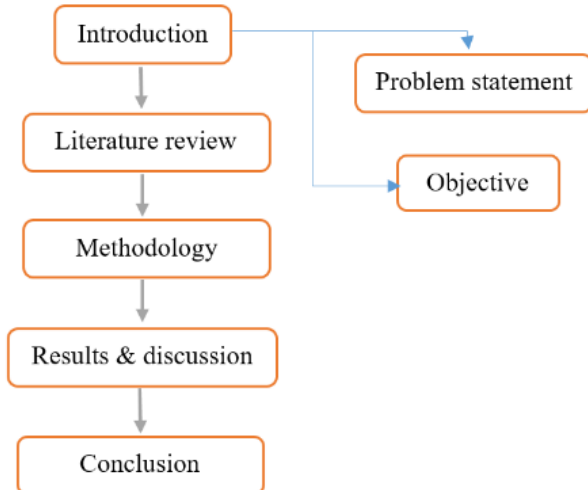


Fig. 2 Structure of the research (Developed by the authors)

The diagram provides the stepwise process of the study. The research findings are presented to discuss the outcome of the research. The last part of the paper

is a conclusion, summarizing the study presented and discussed.

2. Research Background

2.1. Artificial Intelligence and SQA

The traditional companies' data explosion and extensive powerful computational approach enhancement proposes an effective solution for the software industry. Complex applications in software development have been a very critical issue due to sensitive coding and testing. Artificial intelligence has recently made substantial efforts toward intelligent software applications [5]. Virtual assistants, autonomous vehicles, medical diagnoses, etc. are major artificial intelligence contributions to software quality improvement – these tasks few years ago were competing with the human approach. AI is a collection of algorithms that performs a human approach in a computational sense. The AI subfields of machine learning and deep learning [7] have contributed innovative steps toward the next future software technologies. The software industry has been continually approaching and implementing AI-based smart systems. AI-based systems, which are capable of penetrate almost every software application domain. Machine learning, deep learning, and NLP-related algorithms have achieved great innovation in the field of software quality and testing. Rapid and efficient growth in software development processes with the capability to assure automated quality checking [6] saves time and costs in the software development industry. This entails and imparts quality standardization with the acceleration of safe and accurate development. Automation and optimization of software development workflow generated with specific models. AI-based testing and validation studies are given below in Table 1.

Table 1 Studies focused on AI-based techniques for SQA (Developed by the authors)

No.	Source	Objectives	Parameters	Limitation
1	[6]	ML-based SQA model approaches	High level granularity	ML and SQA
2	[7]	SQA security using the DL	QA security Reusability Implementation of the DL process	DL-based approach from the perspective of SQA
3	[8]	To highlight the integration of AI in radiotherapy with SQA	Case-specific QA Routine QA	SQA, AI, and medical applications
4	[9]	To implement AI-based SQA testing	AI-driven testing in SQA	AI and Testing in SQA
5	[10]	To highlight quality evaluation domain-specific discussion	SQA domain-specific testing ML-based model testing	ML and SQA
6	[11]	To analyze holistic assessment using ML methods for identifying possible risks from the perspective of SQA	Quality risks Mitigation of Quality risks Customer confidence using the AI-based model at release time	SQA risks and ML Model Assessment for Risk Analysis

Continuation of Table 1

7	[10]	To discuss important guidelines for ML- and AI-based SQA in the software industry	Quality evaluation aspects Current state of the art of techniques	SQA evaluation methods of AI-based methods
8	[11]	Self- adaptive and self-learning models	Accuracy, process and correctness measures with software quality, quality-in-use and data quality	AI from the perspective of SQA
9	[12]	CPPS - Cyber physical production systems	V-models for QA Data quality holistic approach. Step-wise and level-wise control	AI and Data Quality in SQA
10	[13]	Diverse ML algorithms and pattern recognition strategies	Software testing approaches and quality by metamorphic testing methods	AI software testing and quality validation

2.2. Application of AI in the SQA

AI techniques are incorporated in many software industry companies. AI applications in the software industry are vast to cover. However, few important applications of AI in the software industry are discussed below:

Software and its engineering [14] used to control the software development cycle with effective AI-based SQA methods. The engineering of software includes the design, development, testing, deployment, and management of software. Software creation and management involves critical steps that cover management issues in SDC. This includes planning,

scheduling, developing, testing, deploying, and supporting [15]. The new versions of systems need to support IT systems products by applying AI-based techniques. Software techniques used for software authentication and validation during several SQA phases. Validation covers customer-specific requirements, and verification checks the process that meets software specifications. Software defects analysis is used to automate defect analysis [8] to ensure software quality. Software testing and debugging are used to debug errors by automated AI packages that cover errors with debugging techniques.

Table 2 Techniques of ML in SQA advantages and drawbacks (Developed by the authors)

Tech	Definition	Advantage	Drawback	Ref.
RNN	Recurrent network for every element sequence	Input of any length possibility is acceptable	Slow computation and difficulty in accessing information	[17]
LSTM	RNN subtype to avoid vanishing gradient problems	Long-time information saving and remembrance	Takes longer time for training and requires more memory to train	[18] [19]
CNN	It uses convolution of at least one layer	Detects automatically important features	Needs a huge data set and high computation cost	[19]-[21]
Stacked Auto-encoder	The type of neural network consists of several layers	Uses a pre-trained transfer learning model	Expensive in terms of computational power extremely uninterruptable	[22], [23]
LR	Describe data and relationship between one binary and independent variables	Easy and efficient training	Does not combine features for generation	[24]
SVM	Used for both regression and classification tasks	Uses kernel function and less computational power	Software with high density cannot be updated	[25]

3. Research Methodology

3.1. Inclusion and Exclusion Criteria

This study consisted of different phases. In the initial phase, data were collected by qualitative methods. The projected data pool were five different levels of software companies that used AI for SQA and tried to get data count up to 1200 responds on the designed questionnaires and interviews. The data were collected in an extremely careful manner so that the responses were accurately mapped. The interview and all data collection techniques were able to get

responses in the form of numbered responses from 1 to 5, where 1 is the excellent remark and 5 is the worst. The second stage consisted of scrutinizing valid responses from the expected 1200 responses. The criteria for using the data were logically and physically patterned. The corrected and reliable result was expected to be in the number of 700 to 800 (approximately) gathered for the analysis. At the third and last stage, the qualified responses were analyzed using the statistical tools for obtaining the required reliable results to fulfill the objective of the study.

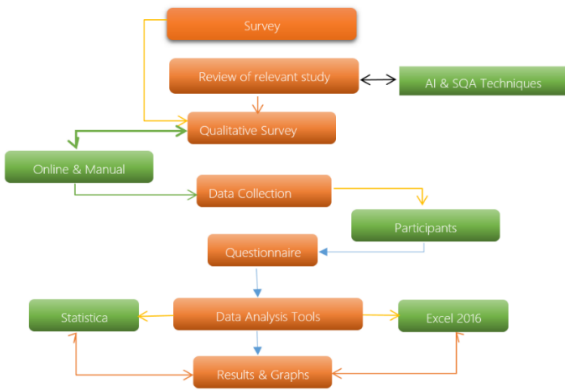


Fig. 3 Research methodology and working plan (Developed by the authors)

3.2. Data Collection and Analysis Discussion

A questionnaire was developed to determine the impact of AI in testing and quality assurance by finding the opinions of different software engineers and companies.

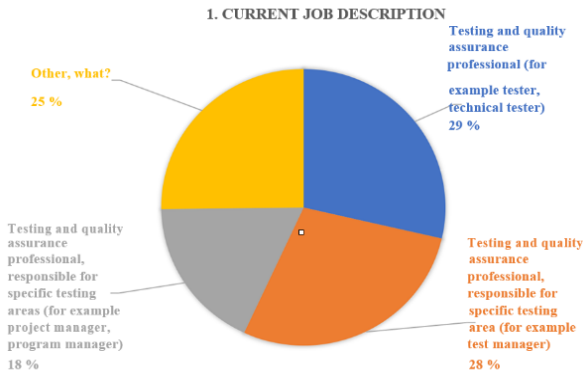


Fig. 4 Percentage of participants' current job descriptions (Developed by the authors)

Question 1 referred to participants' professions. It offered single mandatory selection for 93 participants. Figure 4 shows details of participants' job specification. Thus, 24 participants (25%) had existing job descriptions. Ten participants had existing jobs directly related to SQA, and 14 participants' current job descriptions were related to IT departments.

Question 2. Do you work for an SWE house that specializes in testing and quality assurance?

This was a Yes/No-question (mandatory) and 93 participants were targeted for the survey. The survey statistics are given in Figure 5.

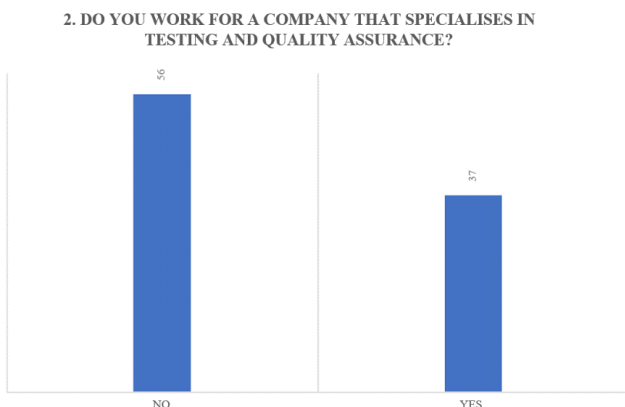


Fig. 5 Participants' responses to Q2

Question 3. How many years have you worked in testing and quality assurance?

This was a Yes/No-question (mandatory), and 93 participants were targeted for the survey. The survey statistics are given in Figure 6, which shows participants' experience according to their designation and career.

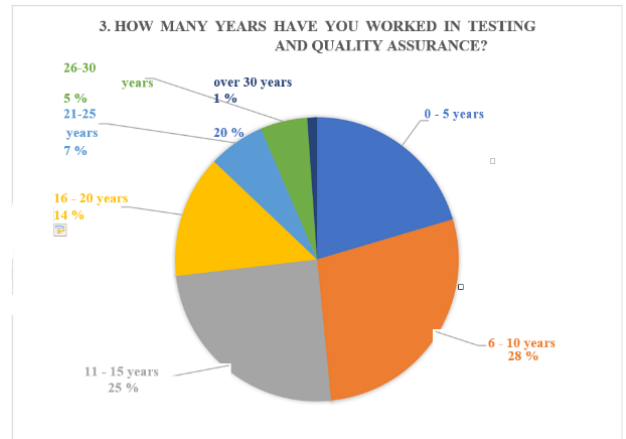


Fig. 6 Participants working in testing and quality assurance (Developed by the authors)

Question 4 referred to the properties of quality assurance experts. It offered evaluation (mandatory) selection for 93 participants, who assessed these properties choosing the following answers: minor, important, critical, and don't know. Figure 7 shows the survey statistics of the respondents' opinions.

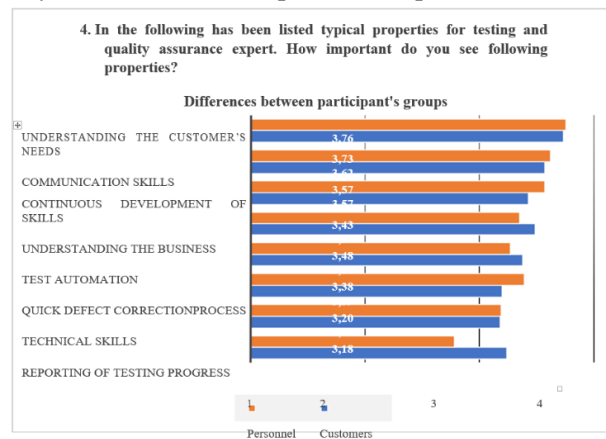


Fig. 7 Participants' opinions about the importance of properties for testing and quality assurance experts (Developed by the authors)

Question 5. Which testing and quality assurance areas will be added in the future?

This was a multi-line text box (optional) and 41 participants were targeted for the survey. Table 3 provides the survey statistics.

Table 3 Quality assurance and testing expert specialty as a career in the future (Developed by the authors)

Statement	Time Mentioned
Picture theme understanding	15
Test automation	6

Continuation of Table 3

Evolving complete testing	6
Developing methods understanding (Scrum, DevOps, SAFe, Kanban)	6
Developing processes and understanding	5
Initiative	4
Technical aspect understanding of the system or systems	4
Testing by applying exploratory and experimental approaches	3
Testing needed for accountability and profitability	3
Cooperation skills	3

Changes in methods for testing can cause different results. Properties with skills needed to add 1-2 times. Other important aspects included test case planning, maintenance of test data, integrity testing, security aspect, tools, and user experience. These parameters are important and are considered in Table 3. The testing expert can further add and perform other important tasks if needed.

Question 6. How important do you find the matters listed below in the future?

It offered evaluation (mandatory) selection for 93 participants, who assessed the matters choosing the following answers: minor, important, very important, and do not know. The survey revealed how important these matters are from the perspective of SQA, which is shown in Figures 8 and 9.

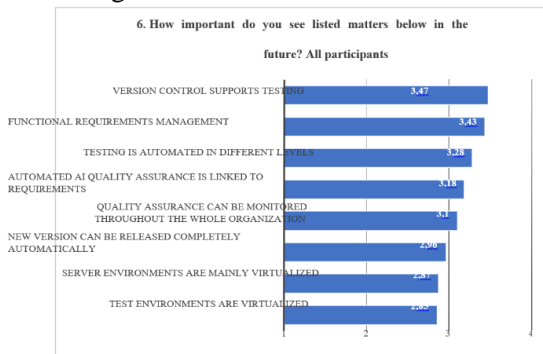


Fig. 8 Participants' answers to Q6 (Developed by the authors)

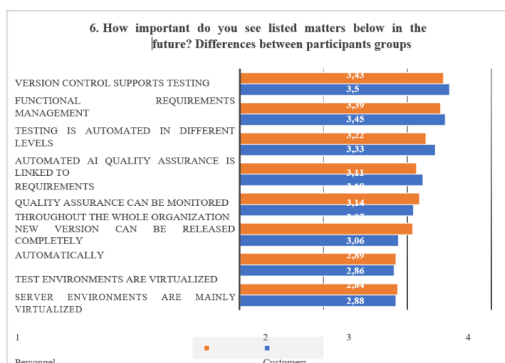


Fig. 9 Participants' answers to Q6 (Developed by the authors)

Question 7. How imperative are issues/matters related to quality reassurance production in the next few years?

This multiline text box was intended for 34 participants. The question referred to quality assurance issues/matters for software industry development.

Table 4 SQA for software quality production in the future

Statement	Time Mentioned
Overall test automation, including services and environments. Automate only beneficial parts	15
A working development team, cooperation and communication	9
Automate daily deployment to production	6
Branching and managing different software versions and environments. Acceptance test environment should correspond to the production	7
Shift left testing. Testing should be initiated right from the start. Provide transparency of quality assurance	4
Business and customer requirements are in the core of the project. Provide requirement traceability	4

Question 8. How important do you see the automation of the following test types in the future?

It offered evaluation (mandatory) selection for 91 participants targeted for the survey. Who chose among the following answer options: minor, important, very important, and do not know, respectively. Figures 10 and 11 show the survey statistics for this question.

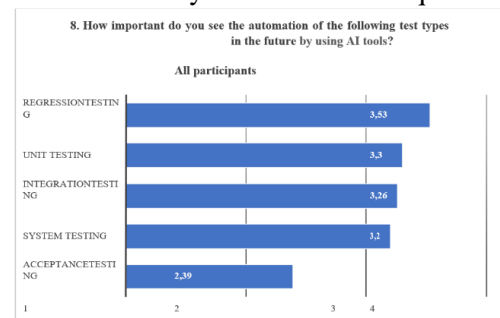


Fig. 10 Participants' answers to Q8 (Developed by the authors)

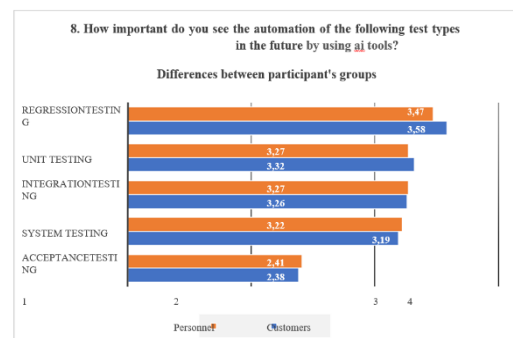


Fig. 11 Participants' answers to Q8 (Developed by the authors)

Question 9. How comprehensive automation should be performed in future using AI tools?

This question offered selection for 91 participants targeted for the survey. Figure 12 shows the customers' answers, and Figure 13 shows the personnel's answers.

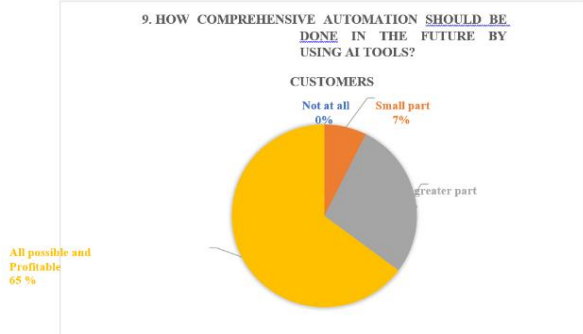


Fig. 12 Customers' answers to Q9 (Developed by the authors)

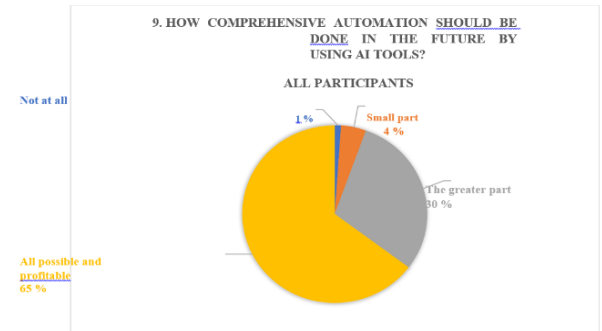


Fig. 13 Participants' answers to Q9 (Developed by the authors)

4. Results

Artificial intelligence approaches will enhance the quality of applications through new approaches to testing and regression testing by describing the impact factor of AI SQA tools (Table 5).

Table 5 Survey of AI tools practices in software quality assurance in Pakistan (Developed by the authors)

No.	Software House	AI tools used	AI Tools Not used
1	Digital market industry		Uses traditional debugging for SQA
2	Softech house	Uses AI tools for SQA	
3	Modern Software's		Uses traditional debugging for SQA
4	An agile industry		Uses traditional debugging for SQA
5	The IThub		Uses traditional debugging for SQA
6	The SoftwareTech	Uses AI tools for SQA	
7	The new trend software		Uses traditional debugging for SQA
8	The SOFTIT		Uses traditional debugging for SQA
9	NewITTECH		Uses traditional debugging for SQA
10	NEWTECHIT		Uses traditional debugging for SQA

It will also aid in developing steep-quality, less insecure, and safer software. AI can help manage complex digital realms, such as dynamically decentralized networks, surveillance, robotics, and digital services of the next era (often grouped under the term "Next Generation Internet"). Outputs can help promote work on specific AI-based domains that may be required to implement modern digital platform smart activities that involve near interrelationships involving innovations, semblance, and implementations in quality assurance. The algorithms would potentially have a major effect on all aspects of our existence. Artificial intelligence is part of a continuum of technologies developed to help individuals, businesses, and society provide quality work with the best user experience.

5. Conclusion

This study entails a precise discussion of software quality assurance using AI-based tools and methods. The software industry challenges, methods, and

implementation of tools can be easily applied to overcome debugging issues. This was achieved by designing a short survey from software houses in Pakistan. The survey questions were asked to know AI tools related awareness. Artificial intelligence-based tools in the software industry help build high-quality software. The research will help in the improvement of identification and classification, recognition and profiling, behavioral detection and analysis, context identification, and classification of human-created errors in development. This will also cope with credibility and wellness monitoring, systems integration, decision-making, suggestion and collection, intelligent direction and behavior, and the ability to interpret and anticipate.

5.1. Limitations of the Study

This study is limited to SQA from the perspective of artificial intelligence tools. This research is based on software testing and quality assurance in software houses of Pakistan.

5.2. Recommendations

This study can further be extended to focus hybridized approaches of ML and DL models to verify the quality assurance of software.

References

- [1] PERETZ-ANDERSSON E, and TORKAR R. Empirical AI Transformation Research: A Systematic Mapping Study and Future Agenda. *e-Informatica Software Engineering Journal*, 2022, 16(1): 220108, <https://doi.org/10.37190/e-Inf220108>
- [2] MAXIME CARLOS T, and IBRAHIM M N. Practices in software testing in Cameroon challenges and perspectives. *Electronic Journal of Information Systems in Developing Countries.*, 2021, 87(30): 1–17, <https://doi.org/10.1002/isd2.12165>.
- [3] BATTINA D S. The Challenges and Mitigation Strategies of Using DevOps during Software Development. *International Journal of Creative Research Thoughts*, 2021, 9(1): 4760–4765, <https://ssrn.com/abstract=4004335>
- [4] WANG J, and ZHANG C. Software reliability prediction using a deep learning model based on the RNN encoder–decoder. *Reliability Engineering and System Safety*, 2018, 170: 73–82, <https://doi.org/10.1016/j.res.2017.10.019>.
- [5] PAIVA D M B, FREIRE A P, and DE MATTOS FORTES R P. Accessibility and Software Engineering Processes: A Systematic Literature Review. *Journal of Systems and Software*, 2021, 171: 110819, <https://doi.org/10.1016/j.jss.2020.110819>.
- [6] SIDHU B K, SINGH K, and SHARMA N. A machine learning approach to software model refactoring. *International Journal of Computers and Applications*, 2022, 44(2): 166–177, <https://doi.org/10.1080/1206212X.2020.1711616>.
- [7] MIHOLCA D-L, TOMESCU V-I, and CZIBULA G. An in-Depth Analysis of the Software Features’ Impact on the Performance of Deep Learning-Based Software Defect Predictors. *IEEE Access*, 2022, 10: 64801–64818, <https://doi.org/10.1109/access.2022.3181995>.
- [8] TUNC M A, E. GURES E, and SHAYEA I. A Survey on IoT Smart Healthcare: Emerging Technologies, Applications, Challenges, and Future Trends. *ArXiv abs/2109.02042*, 2019, pp. 1–15, <https://doi.org/10.48550/arXiv.2109.02042>
- [9] ARORA S, KUMAR V, and SABO T. Designing Software with Artificial. 2021, 08(03): 839–846.
- [10] GORDIEIEV O, GORDIEIEVA D, KHARCHENKO V, KONDIUS I, and BREZHNIEV I. Extended Model of Software Quality Assessment Scenario: Concept, Operations, Application. *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems*, May 12–13, 2022, Gliwice, Poland, 9388: 1–10.
- [11] YUCALAR F, OZCIFT A, BORANDAG E, and KILINC D. Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability. *Engineering Science and Technology, an International Journal*, 2020, 23(4): 938–950, <https://doi.org/10.1016/j.jestch.2019.10.005>.
- [12] RAJAPAKSHA D, TANTITHAMTHAVORN C, JIARPAKDEE J, et al. SQA Planner: Generating Data-Informed Software Quality Improvement Plans. *IEEE Transactions on Software Engineering*, 2021, 5589(c): 1–24, <https://doi.org/10.1109/TSE.2021.3070559>.
- [13] RAJABLI, N FLAMMINI F, NARDONE R, and VITTORINI V. Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review. *IEEE Access*, 2020, 9: 4797–4819, <https://doi.org/10.1109/ACCESS.2020.3048047>.
- [14] ARVANITOU E M, AMPATZOGLOU A, CHATZIGEORGIOU A, and CARVER J C. Software engineering practices for scientific software development: A systematic mapping study. *Journal of Systems and Software*, 2021, 172: 110848, <https://doi.org/10.1016/j.jss.2020.110848>.
- [15] HUTCHINSON B, et al. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. *FAccT 2021 – Proceedings of 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 560–575, <https://doi.org/10.1145/3442188.3445918>.
- [16] SHYNKARENKO V, and ZHUCHYI L. Semantic Checking of Different Type Information Sources About Permitted Speeds in Railway Transport. *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems*, May 12-13, 2022, Gliwice, Poland, 3171: 53.
- [17] WANG J, & ZHANG C. Software reliability prediction using a deep learning model based on the RNN encoder–decoder. *Reliability Engineering and System Safety*, 2018, 170: 73–82. <https://doi.org/10.1016/j.res.2017.10.019>
- [18] DAM H K, PHAM T, NG S W, et al. Lessons learned from using a deep tree-based model for software defect prediction in practice. *IEEE International Working Conference on Mining Software Repositories*, 2019 May, Montreal, Canada (pp. 46–57). <https://doi.org/10.1109/MSR.2019.00017>
- [19] DAM H K, PHAM T, NG S W, et al. (2018). A deep tree-based model for software defect prediction. *arXiv e-prints*, arXiv-1802.00921, <https://doi.org/10.48550/arXiv.1802.00921>.
- [20] MOU L, LI G, ZHANG L, et al. Convolutional neural networks over tree structures for programming language processing. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, February (pp. 1287-1293). <https://doi.org/10.48550/arXiv.1409.5718>
- [21] PHAN A V, LE NGUYEN M, & BUI L T. Convolutional neural networks over control flow graphs for software defect prediction. *Proceedings of the IEEE 29th International Conference on Tools with Artificial Intelligence*, 2017, November (pp. 45-52).
- [22] MANJULA C, & FLORENCE L. Deep neural network based hybrid approach for software defect prediction using software metrics. *Cluster Computing*, 2019, 22(4): 9847-9863.
- [23] TONG H, LIU B, & WANG S. Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning. *Information and Software Technology*, 2018, 96: 94–111.
- [24] KAMEI Y, SHIHAB E, ADAMS B, et al. A large-scale empirical study of just-in-time quality assurance. *IEEE Transactions on Software Engineering*, 2012, 39(6): 757-773.
- [25] ELISH K O, & ELISH M O. Predicting defect-prone software modules using support vector machines. *Journal of Systems and Software*, 2008, 81(5): 649-660.

参考文献:

- [1] PERETZ-ANDERSSON E 和 TORKAR R. 实证人工智能转型研究: 系统映射研究和未来议程. 电子信息学软件工程杂志, 2022, 16(1): 220108, <https://doi.org/10.37190/e-Inf220108>
- [2] MAXIME CARLOS T 和 IBRAHIM M N. 喀麦隆软件测试实践的挑战和观点. 发展中国家信息系统电子杂志, 2021, 87(30): 1-17, <https://doi.org/10.1002/isd2.12165>.
- [3] BATTINA D S. 在软件开发过程中使用开发运营的挑战和缓解策略. 国际创造性研究思想杂志, 2021, 9(1): 4760-4765, <https://ssrn.com/abstract=4004335>
- [4] WANG J, 和 ZHANG C. 基于循环神经网络编解码器的深度学习模型的软件可靠性预测. 可靠性工程与系统安全, 2018, 170: 73-82, <https://doi.org/10.1016/j.res.2017.10.019>.
- [5] PAIVA D M B, FREIRE A P 和 DE MATTOS FORTES R P. 可访问性和软件工程过程: 系统文献综述. 系统与软件杂志, 2021年, 171: 110819, <https://doi.org/10.1016/j.jss.2020.110819>.
- [6] SIDHU B K, SINGH K 和 SHARMA N. 软件模型重构的机器学习方法. 国际计算机与应用杂志, 2022, 44(2): 166-177, <https://doi.org/10.1080/1206212X.2020.1711616>.
- [7] MIHOLCA D-L, TOMESCU V-I 和 CZIBULA G. 软件功能对基于深度学习的软件缺陷预测器性能影响的深入分析. IEEE 接入, 2022, 10: 64801-64818, <https://doi.org/10.1109/access.2022.3181995>.
- [8] TUNC M A, E. GURES E 和 SHAYEA I. 物联网智能医疗保健调查: 新兴技术、应用、挑战和未来趋势. ArXiv abs/2109.02042, 2019年, 第 1-15 页, <https://doi.org/10.48550/arXiv.2109.02042>
- [9] ARORA S, KUMAR V 和 SABO T. 用人工设计软件. 2021, 08(03): 839-846.
- [10] GORDIEIEV O, GORDIEIEVA D, KHARCHENKO V, KONDIUS I 和 BREZHNEV I. 软件质量评估场景的扩展模型: 概念、操作、应用. 第六届计算语言学和智能系统国际会议论文集, 2022年5月12-13日, 波兰格利维采, 9388: 1-10.
- [11] YUCALAR F, OZCIFT A, BORANDAGE E 和 KILINC D. 软件质量工程中的多分类器: 组合预测器以提高软件故障预测能力. 工程科学与技术, 国际期刊, 2020, 23(4): 938-950, <https://doi.org/10.1016/j.jestch.2019.10.005>.
- [12] RAJAPAKSHA D, TANTITHAMTHAVORN C, JIARPAKDEE J 等. 质量保证局规划师: 生成基于数据的软件质量改进计划. IEEE 软件工程汇刊, 2021, 5589(c): 1-24, <https://doi.org/10.1109/TSE.2021.3070559>.
- [13] RAJABLI, N FLAMMINI F, NARDONE R 和 VITTORINI V. 安全自动驾驶汽车的软件验证和验证: 系统文献综述. IEEE 接入, 2020, 9: 4797-4819, <https://doi.org/10.1109/ACCESS.2020.3048047>.
- [14] ARVANITOU E M, AMPATZOGLOU A, CHATZIGEORGIOU A 和 CARVER J C. 科学软件开发的软件工程实践: 系统映射研究. 系统与软件杂志, 2021, 172: 110848, <https://doi.org/10.1016/j.jss.2020.110848>.
- [15] HUTCHINSON B, 等人. 迈向机器学习数据集的责
- 任: 软件工程和基础设施的实践. 流式细胞术 2021 - 会议记录. 2021年 ACM 公平、问责和透明度会议, 2021, 第 560-575 页, <https://doi.org/10.1145/3442188.3445918>.
- [16] SHYNKARENKO V 和 ZHUCHYI L. 铁路运输允许速度的不同类型信息源的语义检查. 第六届计算语言学和智能系统国际会议论文集, 2022年5月12-13日, 波兰格利维采, 3171: 53.
- [17] WANG J, 和 ZHANG C. 使用基于循环神经网络编码器-解码器的深度学习模型进行软件可靠性预测. 可靠性工程与系统安全, 2018, 170: 73-82. <https://doi.org/10.1016/j.res.2017.10.019>
- [18] DAM H K, PHAM T, NG S W, 等. 在实践中使用基于深度树的模型进行软件缺陷预测的经验教训. IEEE 国际采矿软件存储库工作会议, 2019年5月, 加拿大蒙特利尔 (第 46-57 页). <https://doi.org/10.1109/MSR.2019.00017>
- [19] DAM H K, PHAM T, NG S W, 等. (2018). 用于软件缺陷预测的基于深度树的模型. arXiv 电子印刷品, arXiv-1802.00921, <https://doi.org/10.48550/arXiv.1802.00921>.
- [20] MOU L, LI G, ZHANG L, 等. 用于编程语言处理的树结构上的卷积神经网络. 第 30 届 AAAI 人工智能会议论文集, 2016年2月 (第 1287-1293 页). <https://doi.org/10.48550/arXiv.1409.5718>
- [21] PHAN A V, LE NGUYEN M 和 BUI L T. 用于软件缺陷预测的控制流图上的卷积神经网络. IEEE 第 29 届人工智能工具国际会议论文集, 2017年11月 (第 45-52 页).
- [22] MANJULA C, 和 FLORENCE L. 基于深度神经网络的混合方法, 使用软件指标进行软件缺陷预测. 集群计算, 2019, 22(4): 9847-9863.
- [23] TONG H, LIU B, 和 WANG S. 使用堆叠去噪自动编码器和两阶段集成学习的软件缺陷预测. 信息与软件技术, 2018, 96: 94-111.
- [24] KAMEI Y, SHIHAB E, ADAMS B, 等. 及时质量保证的大规模实证研究. IEEE 软件工程汇刊, 2012年, 39(6): 757-773.
- [25] ELISH K O 和 ELISH M O. 使用支持向量机预测容易出现缺陷的软件模块. 系统与软件学报, 2008, 81(5): 649-660.