



**UNIVERSITY  
OF LATVIA**

**FACULTY OF ECONOMICS AND SOCIAL SCIENCES**

**VadZM007: PROJECT MANAGEMENT**

**MASTER'S PROGRAMME**

Automation in QA: A Project Management Framework for Delivering  
AI-Driven Testing Solutions

Kvalitātes nodrošināšanas automatizācija: projektu pārvaldības sistēma  
AI vadītu testēšanas risinājumu nodrošināšanai

Master Thesis

Author: Sarin Preman Sindhu

Student's ID:- sp23038

Advisor:- Andrejs Cekuls

## **Abstract**

Quality assurance (QA) relates to determining the meeting of quality of product or service with standard requirements. It establishes and maintains standards requirements while manufacturing reliable and quality products. A well QA process enhances customer satisfaction and build a company's credibility and brand image. It also helps a company to achieve a competitive position in the market. The concept of QA is important in the manufacturing industry. The QA process is not limited to manufacturing industries, but it is also applicable to software engineering. Software quality assurance (SQA) systematically identifies patterns, and the actions and improve development cycles involved in software development. However, finding and fixing coding errors are beneficial during the process of SQA. It is essential for software developers to mitigate errors before they occur, this can save software development time and expenses involved in software. These errors are known as bugs. These have a significant impact on the performance of software. AI models for software testing are gaining rapid growth. By employing generative AI in software testing, SQA process can be enhanced. It further automates the testing procedures and simultaneously boosts the testing quality.

Generative AI models are more suited for software development. This ensures that AI testing can enhance the testing methods. Therefore, SQA teams can enhance the process of testing. AI tools can automate the testing methods using generative AI models. This can help in the identification of bugs. Generative AI models allow to visualize the testing data. This can enable to meet the customer requirements. However, there are many challenges in the adoption of AI tools that can automate the testing methods. The aim of the study is to design and propose a project management framework tailored for delivering AI-driven testing solutions in QA. Results revealed that technical complexity, skill and knowledge gaps, financial constraints, and organizational resistance are the major factors in implementing AI-driven QA automation in IT industries. The proposed framework can be used by the IT industry.

## Keywords

<b>Key words</b>
Defect prevention
AI-based testing tools
Performance
Codeless Automation
AI driven Framework
Generative AI
SQA (Software Quality Assurance)
Challenges
ROI (Return On Investment)
Automation Testing

## Table of Contents

<b>Introduction</b> .....	<b>5</b>
<b>1. CURRENT STATE OF QA AUTOMATION AND AI-DRIVEN TESTING SOLUTIONS</b>	<b>7</b>
<b>1.1 Concept of quality assurance</b> .....	<b>7</b>
<b>1.2 Understanding of AI-based Software Testing and quality assurance</b> .....	<b>10</b>
<b>1.3 Future Trends of AI-based Software Testing and quality assurance</b> .....	<b>13</b>
<b>2. IDENTIFY CHALLENGES IN IMPLEMENTING AI-DRIVEN QA AUTOMATION</b> ...	<b>18</b>
<b>2.1 Understanding data quality</b> .....	<b>18</b>
<b>2.2 Inconsistent data</b> .....	<b>22</b>
<b>2.3 Data privacy and security</b> .....	<b>24</b>
<b>2.4 Survey analysis</b> .....	<b>26</b>
<b>3. BUILDING AI DRIVEN TEST AUTOMATION FRAMEWORK FOR QA</b> .....	<b>74</b>
<b>3.1 Plan for the adoption of AI driven framework for QA</b> .....	<b>74</b>
<b>3.2 Planned costs for the adoption of AI driven framework for QA</b> .....	<b>75</b>
<b>3.3 Cost-benefit analysis for the adoption of AI driven framework for QA</b> .....	<b>78</b>
<b>4. GENERAL CONCLUSIONS</b> .....	<b>81</b>
<b>Literature and sources used</b> .....	<b>83</b>
<b>Appendix</b> .....	<b>90</b>

## Introduction

Recently, the use of artificial intelligence (AI) and machine learning has gained significant attention in QA process improvement. In digital environment, AI and machine learning are used to inspect the products and machinery (Kassie and Singh, 2022). This process is called predictive maintenance. AI Algorithms are used to analyze the performance of the products by gathering the real-time data. This helps the developers to identify the issues before they arise. In today's dynamic business landscape, testing is integral to QA. When the right tools are at place, it simplifies the QA processes. QA team with advance expertise using AI tools can identify the early defect (Gao et al., 2003).

Testing of the product reduces the cost and time to rework. This will strengthen the customer-company relationship. In a software company, the testing phase starts at the development stage and continues till the software is deployed. However, it also depends on the customer requirement (Bajnaid, 2013). For example, customer is interested in the testing of software at multiple stages. On the other hand, testing software at different phases can meet the expectations of customers. The software application that is tested at the end of the project can ensure its quality and performance, which can meet the needs and expectations of customers. The software development life cycle is critical as it addresses the critical aspects of performance. AI-based software testing and QA refers to the integration of AI and machine learning (ML) techniques into the software testing process. The goal is to enhance the performance of software development (Fujii et al., 2024). AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing and enhances testing performance (Qazi et al., 2023). AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance testing productivity. AI models can predict complexity associated with bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts (Smith, 2024). Traditional testing methods are not capable for performing advance testing such as regression testing (in limited time and with limited resources), code discrepancies etc. These drawbacks greatly affect the SQA(Software Quality Assurance) teams performance.

AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI

tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gained more attention, driving further innovation and quality in software development. All mentioned above is the **topicality** of research.

The **problem** is that companies are facing a challenge toward the adoption of AI-driven testing solutions in QA.

The **object** of thesis: challenges faced by IT industry toward the adoption of AI-driven testing solutions in QA

The **subject** of thesis: IT industry

The **aim** of the study is to investigate challenges toward the adoption of AI-driven testing and propose a framework which will overcome these challenges.

The **research tasks** are as follows:

1. To investigate the current state of QA automation and AI-driven testing solutions.
2. To identify challenges in implementing AI-driven QA automation.
3. To develop a project management framework to address these challenges.
4. To conduct survey and gather data and analyze data for the identification of challenges toward the adoption of AI-driven testing solutions in QA.
5. To present conclusions and Recommendations.

### **Research methodology**

Present study is undertaken to investigate the current state of QA automation and AI-driven testing solutions. Further, identify challenges in implementing AI-driven QA automation were also reported in the study. Additionally, current study also develops a project management framework to address these challenges. To achieve these, current study relies on survey method. Present study uses quantitative research method and collects data from employees working in IT companies in India. The employees working in the IT companies in Kerala, India is the target population of the study. The survey is conducted online using Google form. The questionnaires were distributed to the employees after obtaining consent from them. Overall, 152 responses were collected using random sampling technique. The collected data were analyzed using SPSS software version 23.

Thesis consist of three chapters. Chapter 1 investigates the current state of QA automation and AI-driven testing solutions. Chapter 2 identify challenges in implementing AI-driven QA automation. Chapter 3 developed a project management framework to address these challenges.

# **1. CURRENT STATE OF QA AUTOMATION AND AI-DRIVEN TESTING SOLUTIONS**

## **1.1 Concept of quality assurance**

Quality assurance (QA) relates with determining the meeting of quality of product or service with standard requirements (Atoum et al., 2021). It establishes and maintains standard requirements while manufacturing of reliable and quality products. A well QA process enhance customer satisfaction and build a company's credibility and brand image. It also helps a company to achieve a competitive position in the market (Pargaonkar, 2023). The concept of QA is important in the manufacturing industry. The QA process is not limited to manufacturing industries, but it is also applicable to software engineering. Software quality assurance (SQA) systematically identifies patterns, and the actions and improve development cycles involved in software development (Felderer and Ramler, 2021). However, finding and fixing coding errors are beneficial during the process of SQA. It is essential for software developers to mitigate errors before they occur, this can save software development time and expenses involved in software. These errors are known as bugs (Tian, 2005). These have a significant impact on the performance of software.

QA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA (Mnkandla and Dwolatzky, 2006). QA is broader than testing; it involves testing as well as improvement processes. Meanwhile, testing is oriented to identify issues while improvement process relates with the improvement in the process. The quality of products and services is an important indicator which differentiate a firm from their competitors (Naik and Tripathy, 2011). QA ensures that products meet customer demands and expectations. High-quality products enhance brand image, which leads to the development of customer loyalty (Mahfuz, 2016).

QA also reduces the cost of repairing of defective products and satisfy customers because QA can early detect issues in the products. Defective products increase the cost of the product and diminish the brand image (Maxim and Kessentini, 2016). It also increases the additional service cost to rectify the defects. Therefore, for streamlining the QA process, it requires additional investment. In the regard, the team members involved in QA process must define a process workflow and implement QA tools. However, the implementation involves time-consuming

process but when implemented it offers several benefits. If QA is not implemented, more serious issues can be generated which can increase the cost (Khanjani and Sulaiman, 2011).

Recently, the use of artificial intelligence (AI) and machine learning has gained significant attention in QA process improvement. In digital environment, AI and machine learning are used to inspect the products and machinery (Kassie and Singh, 2022). This process is called predictive maintenance. AI Algorithms are used to analyze the performance of the products by gathering real-time data. This helps the developers identify the issues before they arise. In today’s dynamic business landscape, testing is integral to QA. When the right tools are at place, it simplifies the QA processes. QA team with advance expertise using AI tools can identify the early defect (Gao et al., 2003).

Testing of the product can reduce the cost and time to rework. This will strengthen the customer-company relationship and build trust in the customer. Recently most companies adopt shift left testing, which means testing the product at the very early stages of development. However, it also depends on the customer requirement (Bajnaid, 2013). On the other hand, testing software at early phases can bring more clarity to the product that is being built. The software application is tested at the end of the project can ensure its quality and performance which can meet the needs of customers but now a days companies needs to make sure they deliver beyond expectations. The software development life cycle is critical as it addresses the critical aspects of performance.

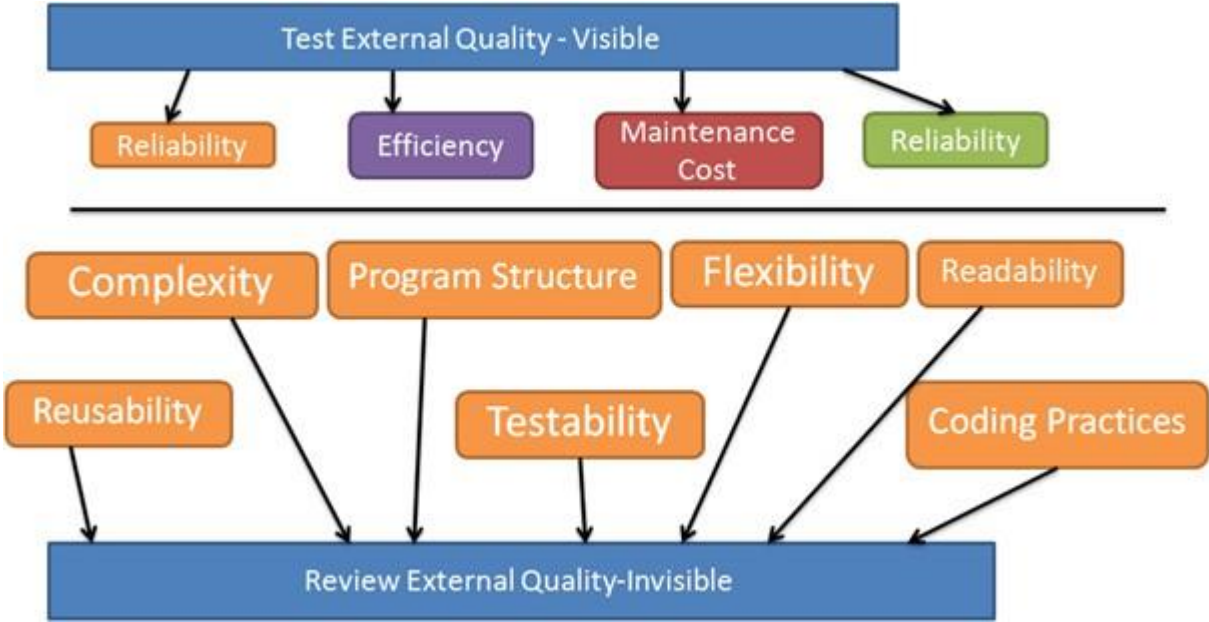
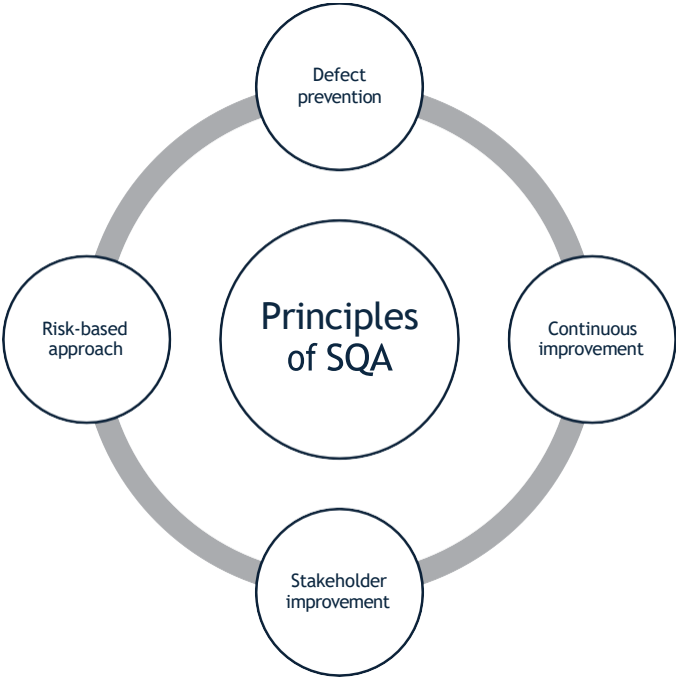


Figure 1.1. Schematic of QA (Test Institute, 2024)

Fig. 1.1 shows the schematic of QA. Testing at each stage of the software development life cycle must be done. Testing of software can be performed both manually and by using automation tools. The QA process of software is divided into various groups such as software security, performance, and usability. But in the broad sense, they can be divided into manual and automated testing phases (Mistrík et al., 2015).

SQA ensures the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance (Ramchand et al., 2021). Developers are now more interested in identifying the early defects in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standards with error free from the code. Automated software testing uses software tools to detect errors in the software (Ouaarous et al., 2023).

Every organization must ensure delivery of a high-quality product or service to their customers. Therefore, it is mandatory to check the quality of the product or application throughout the development process. In software development cycle, developers must ensure the performance of software (Felderer and Ramler, 2021). This can meet the quality standards of the products. QA teams can identify errors and bugs to ensure the software performance. Therefore, QA is crucial in the software development lifecycle. Its main role is beyond the software testing program is meeting exceeding the customer's expectations. By implementing SQA firm can better compete and differentiate themselves in the market. Overall, SQA is essential in the industry for meeting the demand of customers (Farah, 2024). Fig. 1.1 shows the principles involved in the SQA process.



## Figure 1.2. Principles of SQA

Fig. 1.2 shows the principles of SQA. The first principle is defect prevention. It is always better to prevent defects and errors in the early development stages of software. In this phase, emphasis is on the identification of potential issues in the software development lifecycle (Natarajan, 2020). SQA focuses on identifying the root cause of defects and errors. The second principle is continuous improvement. In this phase, continuous improvement is used to consistently monitor the software product. Therefore, continuous improvement is essential for improving the performance of software products. The third principle is stakeholders' involvement. The involvement of stakeholders in the software development process is essential as it improves the performance of software products (Trifunova et al., 2024). This phase also highlights the importance of collaboration and communication between the company and customer. By effective involvement of stakeholders can help to identify the issues in the software product. SQA teams can work and collaborate with customers and identify their needs and simultaneously mitigate defects. SQA teams use tools for the detection of issues and enhance the capability of software development process (Goericke, 2020). Last, principle is risk-based approach. SQA teams take risk and innovate software development process. The risk-taking behavior is essential as it enhances the software development process.

### **1.2 Understanding of AI-based Software Testing and quality assurance**

AI-based software testing and QA refers to the integration of AI and machine learning techniques into the software testing process. The goal is to enhance the performance of software development (Fujii et al., 2024). AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhances testing performance (Qazi et al., 2023). AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to fail. This can enhance testing productivity. AI models can predict complexity associated with bugs. Whenever, there will be a minor change, AI tools can automatically perform analysis on scripts (Smith, 2024). The traditional testing methods are not capable of performing advanced testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA teams can better focus on the performance of software.

Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gained more attention, driving further innovation and quality in software development (Tao et al., 2019).

The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages (Gezici and Tarhan, 2022). It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards (Shekhar, 2022).

The importance of detection of bugs is associated with the performance of software (Nama, 2024). QA is essential to achieving the performance of software. As technology advances, AI tools are now used for improving the software quality (Khankhoje, 2023). By using AI tools, SQA team can better focus on the performance of software. Fig. 1.3 shows the role of AI in SQA.

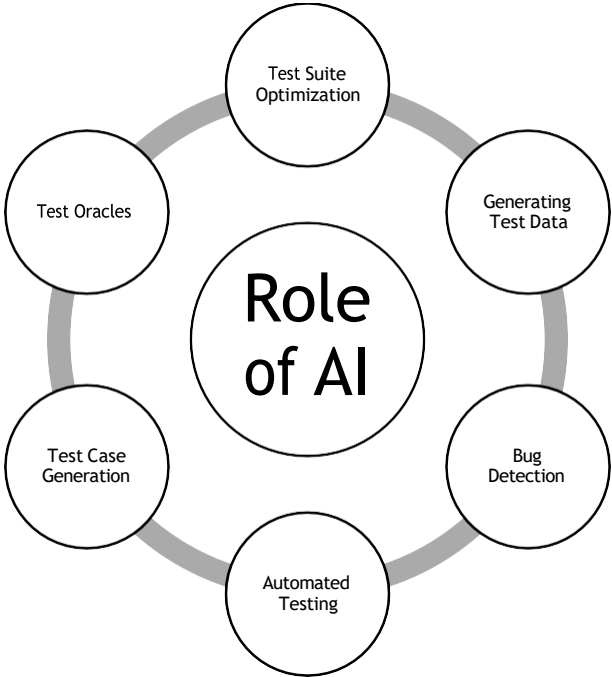


Figure 1.3. **Role of AI in SQA**

Fig. 1.3 shows the role of AI in SQA. From the fig. 1.3, AI has a six role. From bug detection to test suite optimization, AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance

of software (Martínez-Fernández et al., 2022). Based on the requirements, Generative AI models generate test data which can be analyzed by SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data (Awad et al., 2016). Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system (Khan et al., 2023). Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system (Wang et al., 2024). Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of the system(test oracle). Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system (Gezici & Tarhan, 2022).

Utilizing AI models for quality assurance can allow SQA teams to analyze the software quality standards. Improving the software quality is essential for achieving highest performance. By employing AI models, QA teams seek to identify defects in the software, hence, lowering risks is essential for optimized performance (Natarajan, 2020).

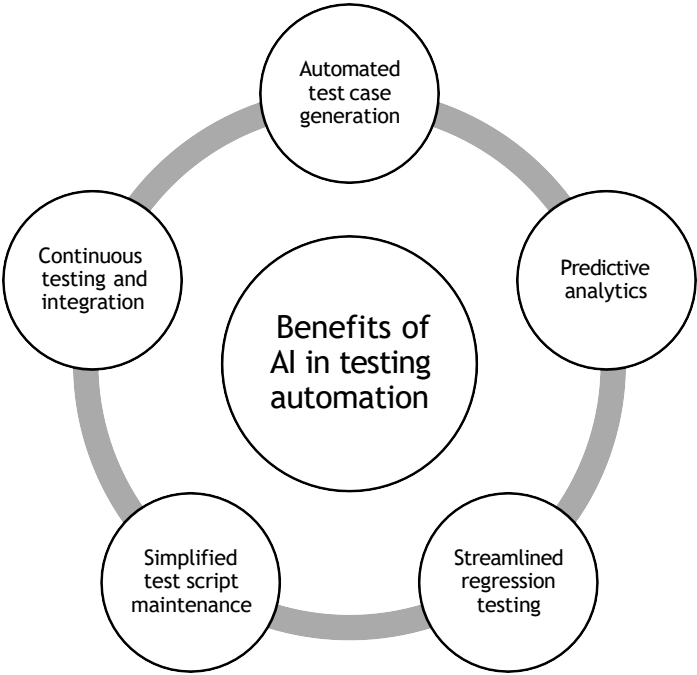


Figure 1.4. **Benefits of AI in testing automation**

Fig. 1.4 shows the benefits of AI in testing automation. Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system (Lenarduzzi et al., 2021). AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models (Ali et al., 2022). Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable (Singhal et al., 2021).

### **1.3 Future Trends of AI-based Software Testing and quality assurance**

AI models for software testing is gaining rapid growth. By employing generative AI in software testing, SQA process can be enhanced. It further automates the testing procedures and simultaneously boost the testing quality. Generative AI models are more suited for software development (Pham et al., 2022). This ensures that AI testing can enhance the testing methods. Therefore, SQA teams can enhance the process of testing. AI tools can automate the testing methods using generative AI models. This can help in the identification of bugs. Generative AI models allow to visualize the testing data (Job, 2021). This can enable to meet the customer requirements. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results (Krichen, 2022). With the evaluation of generative AI in software testing, SQA process can be optimized. This can bring more opportunities for testing and improving the software quality (Khaliq et al., 2022). AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software (Martínez-Fernández et al., 2022). Based on the requirements, Generative AI models generate test data which can be analyzed by SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of the system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

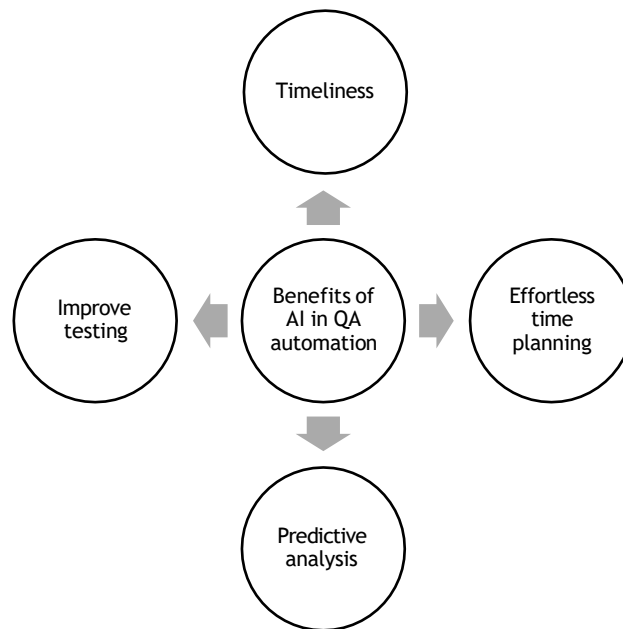


Figure 1.5. **Benefits of AI in SQA**

Fig. 1.5 shows the benefits of AI in SQA. Generative AI models improve the testing procedure. By analyzing the customer data, businesses can achieve the desired results. For instance, generative AI solutions can create testing data. This improvement in testing can enhance the system performance by automation (Al Alamin and Uddin, 2021). Generative AI model employs ML techniques and enhances the performance of the system (Lenarduzzi et al., 2021). Generative AI will dominate the testing landscape. Therefore, businesses can leverage generative AI models and enhance the testing performance. The software testing using AI models can save time, reduce costs, and allow SQA team to focus on system performance.

The generative AI models use past data and benchmark the performance of software development (Chinamanagonda, 2021). In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance (Neto et al., 2022). Generative AI models can lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed by SQA teams. Generative AI models can expand the testing capabilities (Poth et al., 2022). Generative AI models can improve performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. There are various tools available in the current market which can be used to achieve all the objectives listed below in the Fig 1.4.

The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. Fig. 1.4 shows the AI in software test automation.

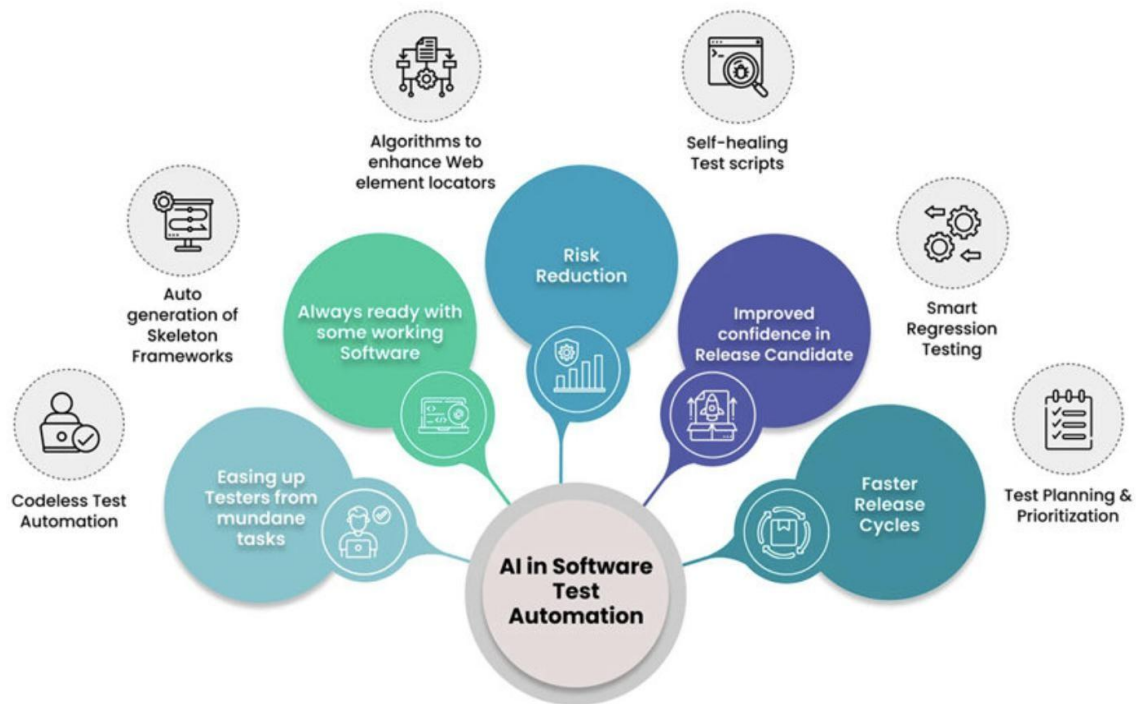


Figure 1.6. AI in software test automation (Gautam, 2024)

Fig. 1.6 shows the AI in software test automation. In the past, software testing has been done by using traditional methods. On the other hand, generative AI models are advancing the software testing. The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages (Veeramachaneni, 2021). Development of software testing can be more advanced using Generative AI models. These models can automate the testing methods. AI models can leverage AI models and automate software testing by analyzing past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns. Cyberattacks are growing rapidly, therefore, AI models can better predict the testing data (Chinamanagonda, 2021). The continuous development of generative AI in the context of software testing is helpful for achieving the desired results (Krichen, 2022). With the evaluation of generative AI in software testing, SQA process can be optimized.

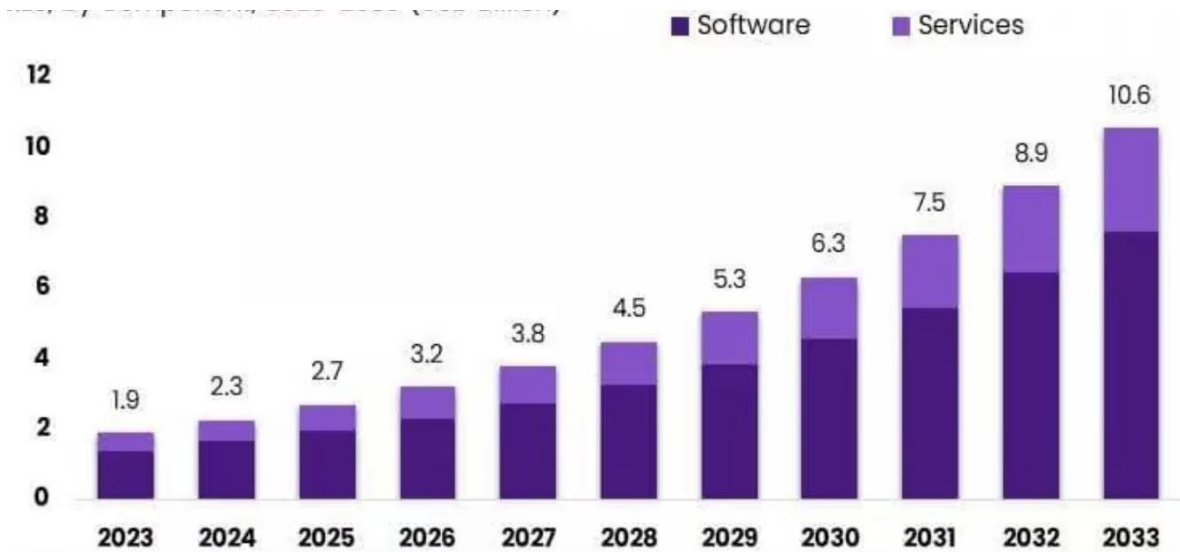


Figure 1.7. Global AI in software testing market (Global AI market, 2024)

Fig. 1.7 shows the global AI in software testing market. The market is expected to reach 884.95 Euro. There is a growing need for identifying bugs by code testing using AI tools. With the use of AI tools, SQA teams can automate test cases that can be helpful for improving the quality of software (Nigam et al., 2021). The QA process is resource-intensive because of the complexity of software (Osamy et al., 2022). AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages (Gezici and Tarhan, 2022). It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards (Shekhar, 2022).

The fourth-generation web advances testing strategies. The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages (Veeramachaneni, 2021). Development of software testing can be more advanced using Generative AI models (Barenkamp et al., 2020). When using modern automation tools like playwright for automating web applications with the help of locators like CSS selector or xpath, instead of CSS selectors or XPath locators, ZeroStep's AI assistant determines what actions to take at runtime based on plain-text instructions written by the SQA team in English language. Since Zero Step integrates directly into Playwright, SQA team can incorporate AI into as few or as many tests as you'd like, without changing development workflow.

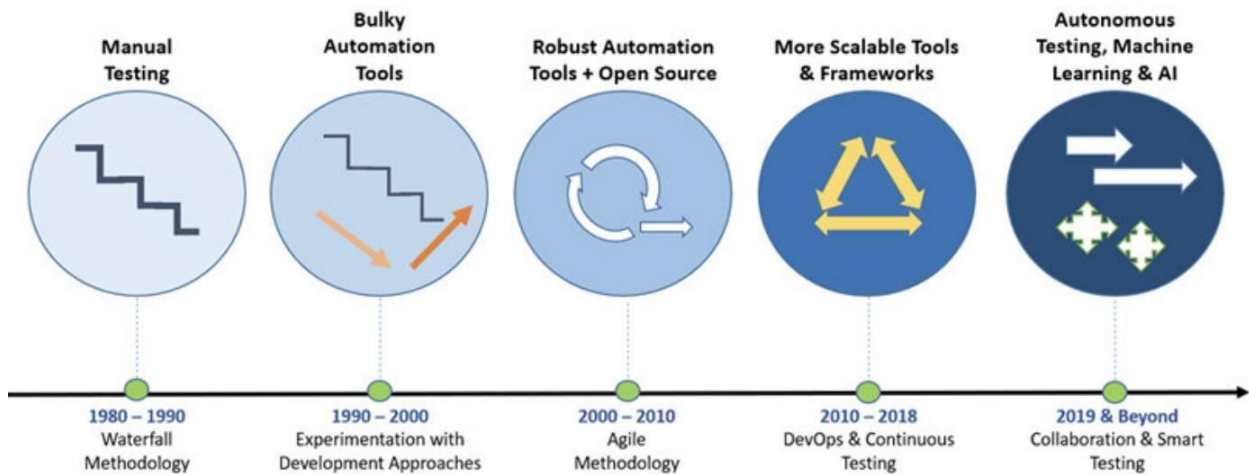


Figure 1.8. Evolution of software testing (Sify, 2024)

Fig. 1.8 shows the evolution of software testing. Generative AI models use test data and analyze the performance of software (Martínez-Fernández et al., 2022). Based on the requirements, Generative AI models generate test data which can be analyzed by SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data (Awad et al., 2016). Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system (Khan et al., 2023). Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system (Wang et al., 2024). Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system (Gezici & Tarhan, 2022).

## **2. IDENTIFY CHALLENGES IN IMPLEMENTING AI-DRIVEN QA AUTOMATION**

### **2.1 Understanding data quality**

Data quality in AI-based testing plays a vital role for software development processes. AI models train data and analyze which can improve the performance of software. However, data quality plays an important role. Additionally, raw data must be accurate, reliable, and representative of real-world scenarios. This will better predict the performance of software. The efficiency of AI-based testing directly depends on the data quality. Data quality is the measure of accuracy, reliability, consistency, and timeliness of data (Song et al., 2022). More specifically, data quality reflects the reliability of test data. Further, data must be reliable and free from errors. The data plays an important role in the improvement of software testing. Consistent data is uniform throughout the testing methods. However, the availability of data is essential because it plays an important role in the testing process. Data quality determines the reliability of a specific dataset. By ensuring reliability, the effectiveness of decision-making processes can be enhanced. In the context of AI-based testing, high-quality data can ensure the accuracy and efficiency of test results, leading to improved software quality.

The quality of data used in AI-based testing significantly impacts the accuracy and effectiveness of the testing results. AI algorithms rely on vast amounts of data, AI models learn, make predictions, and detect patterns. Poor data quality can present biased outcomes, leading to inaccurate conclusions (Ericsson et al., 2021). AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages (Gezici and Tarhan, 2022). It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards (Shekhar, 2022).

The data quality is important because it can impact decision making. For example, e-commerce companies are using AI tools for the recommendation of products to the customers based on their performance. If the data used for training the AI model is incomplete or inaccurate, the recommendations provided to customers may depart from the actual preferences. This can

dissatisfy the customers and negatively affect the performance of the company (Dandotiya, 2025). Moreover, the use AI tools in different sectors have been increased for critical decision-making. However, the impact of poor data quality on decision making is significant. Inaccurate or incomplete data can potentially lead to incorrect outcomes. This can negatively impact the organizations.

Therefore, ensuring the quality of data is crucial when implementing AI-based tools for the testing of software. By ensuring the quality of data, SQA team can predict the performance of the software (Gao et al., 2019). However, the high quality of data is not sufficient, continuous monitoring and maintenance of data is also essential. This will ensure the high quality of data over the time. Prioritizing data quality can be helpful for SQA team for achieving standard results. This will also help organizations to leverage the full potential of AI tools for testing. Also, generative AI models depend on high-quality, and representative data which can ensure the quality standards of software. However, poor data quality can lead to inaccurate test cases and false outcomes. Generative AI software testing has the potential to boost performance, therefore, it is important for SQA team to align on getting the right data sets. (Aleti, 2023).



Figure 2.1. Six dimensions of data quality (Rao, 2024)

Fig. 2.1 shows the six dimensions of data quality. It is also observed that excessive reliance on AI models can be harmful, therefore, a balance of human judgment with AI capabilities is required for achieving the accurate results. With the emergence of data-driven testing, SQA teams can analyze the various data sets and predict the behavior of system performance (Sugali, 2021). The right data set offer several benefits to the SQA teams such as automation test data. Traditionally, software testing consumes a lot of time and often produces errors, therefore, require multiple iterations. Manual testing of software also involves several issues that can hinder the ability of SQA teams in delivering the high-quality products (Shekhar, 2022). However, generative AI models for software testing revolutionize the process by automating if right data set is given. It can also optimize the testing methods which can accelerate the software development cycle.

High-quality data can help firms to innovate the testing methods. It can also accelerate the progress of software testing. Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results (Khan et al., 2023). In essence, the positive impact of data quality on test methods can also be enhanced. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.

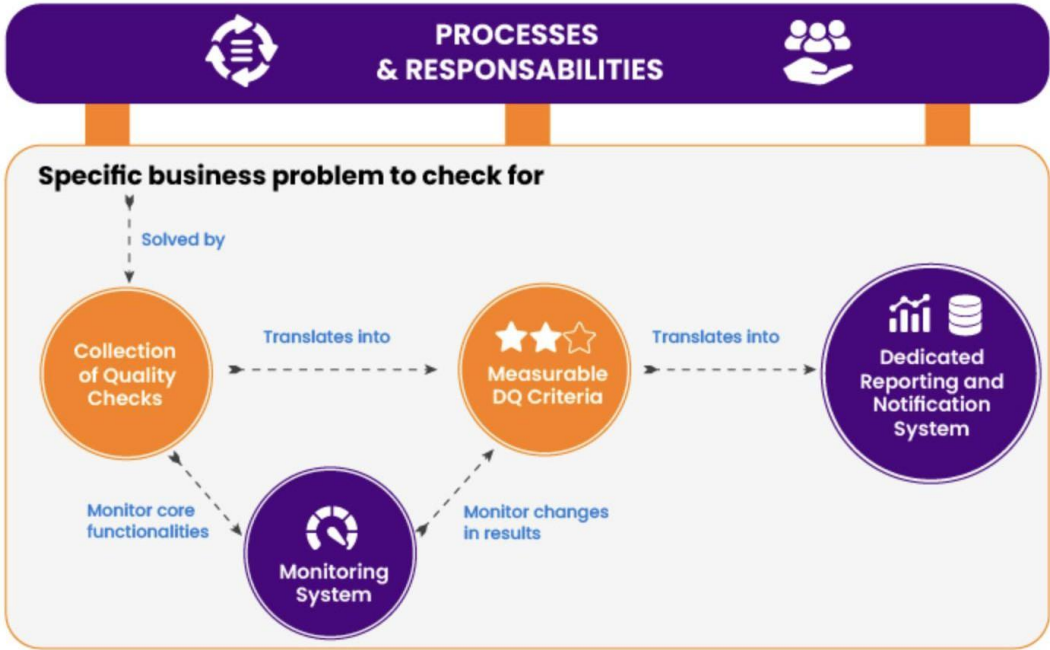


Figure 2.2. Data quality and its impact on AI outcomes (WisdomSchema, 2024)

Figure 2.2 shows the data quality and its impact on AI outcomes. The deep learning models integrating with neural networks can enhance the testing performance of software (Devalla et al., 2023). This can reduce the workload of SQA teams and improve testing performance. Further, software developers do not need to create many data rules to identify the bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively (Awad et al., 2024). Generative AI models use test data and analyze the performance of software (Martínez-Fernández et al., 2022). Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

The quality of data used in AI-based testing significantly impacts the accuracy and effectiveness of the testing results. AI algorithms rely on vast amounts of data, AI models learn, make predictions, and detect patterns. Poor data quality can present biased outcomes, leading to inaccurate conclusions (Ericsson et al., 2021). AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages (Gezici and Tarhan, 2022). It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards (Shekhar, 2022).

For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software (Khan et al., 2023). These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods (Aleti, 2023).

The ability of AI models to identify patterns in historical data can enable SQA team to detect bugs at early stages of software development cycles (Veeramachaneni, 2021). Development

of software testing can be more advanced using Generative AI models (Barenkamp et al., 2020). AI models can automate software testing by analyzing the past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns (Song et al., 2022).

### 2.2 Inconsistent data

AI-based testing systems often depends on the data quality. Further, raw data with insufficient information can make difficult for processing by AI model. Inconsistent quality of data can also be an issue for processing the data using AI models. This can lead to incorrect predictions of the outcome (Shekhar, 2022). Therefore, preprocessing techniques are essential before proceeding to data analysis using AI tools. Additionally, monitoring and refinement of AI algorithms can also help SQA team to handle the data. In AI-based testing, data biases are also a major issue. Biased data can affect the outcomes and performance of the software (Devalla et al., 2023). In this regard, training of data is essential to avoid the data biasness. Further, the biasness in the data can also affect the accuracy of software. Therefore, SQA team must continuously assess and mitigate biases, so that accurate outcomes can be achieved.

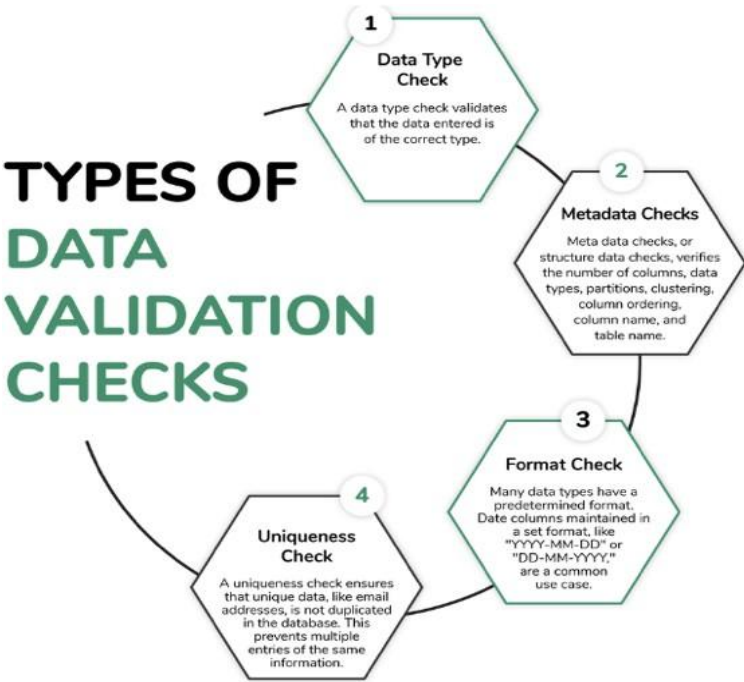


Figure 2.3. Types of data validation checks (Arramton Infotech, 2024)

Figure 2.3 shows the different types of data validation checks. Furthermore, the capability of GenAI for testing at different phases of software development life cycles is also important. Therefore, leveraging the full potential of GenAI for enhancing test practices can also improve the quality and reliability of software products (Ericsson et al., 2021). Therefore, organizations can leverage the full power of GenAI and enhance the testing performance. The evolution of generative AI is gaining significant attention in software testing. GenAI models can enhance the testing performance (Bajnaid, 2013). It interprets data outcome and provide measures for quality standards. This capability of AI models can enhance the testing and SQA of software testing. It simplifies the process of testing and automation by enabling the SQA (Khan et al., 2023).

Automation engineer can transform their roles using AI testing and automation. Traditionally, testing of software is a time-consuming process which uses significant time and resources. AI removes these barriers and accelerates the software development cycles, which improves the accuracy at all development level of software (Dandotiya, 2025).

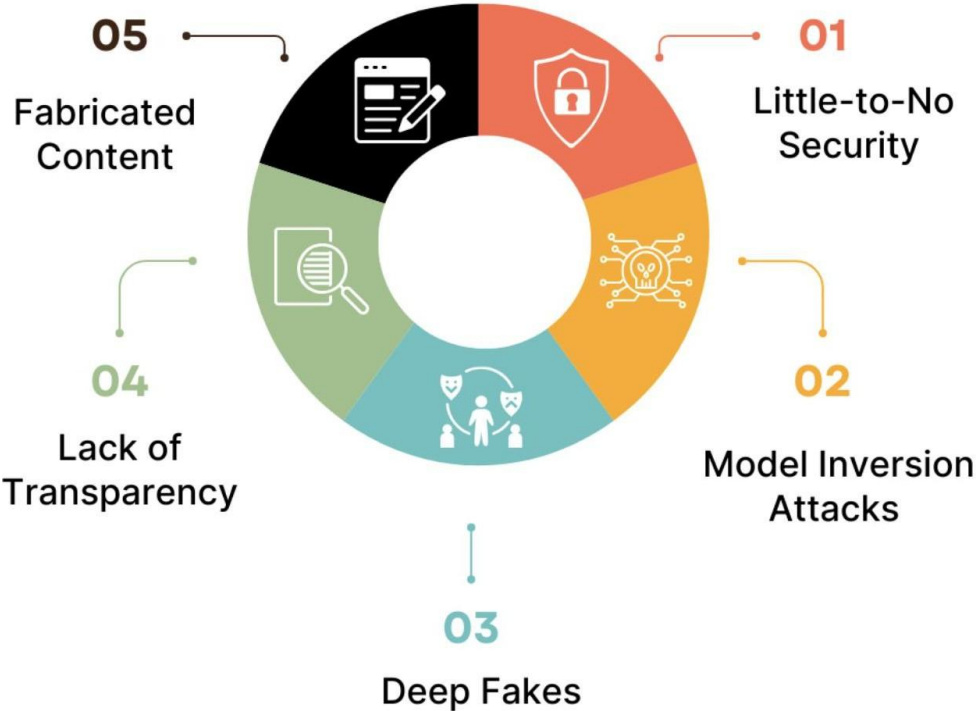


Figure 2.4. Major data privacy challenges (Ksolves, 2024)

AI-powered tools can execute different test scenarios and can automate the testing procedure involved in SQA. This significantly reduces the testing time, allowing QA professionals to focus more on testing and automation (Sofian et al., 2022). This can enhance the testing

procedure. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams (Farah, 2024). Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

The productivity of testing can also increase by adopting AI tools. While traditional testing is limited to the complexity of software. SQA team can now deal with complexities and enhance the performance of testing (Chinamanagonda, 2021). This not only improves the testing capabilities but also offer flexibility and efficiency in SQA process. While GenAI has the potential to deal with complexity, it can also enhance the testing procedures. Many testing tasks at different levels can be automated. However, data biases can reduce the efficiency of testing of software. Therefore, organizations need to mitigate the data biases. By doing so, SQA team can maximize the benefits of GenAI (Singhal et al., 2021).

For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software (Khan et al., 2023). These challenges significantly affect the AI-driven QA (Tao et al., 2019). The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods (Aleti, 2023). Training an AI model is essential, and biasness can affect the data outcomes. So, organizations must reduce biases (Shekhar, 2022). The mitigation of data biases is important as it can affect the decision-making. Organizations can mitigate data biases during the testing phases (Dandotiya, 2025).

### **2.3 Data privacy and security**

Data privacy and security plays an important role as it can lead to compromised results. With increasing focus on data privacy, SQA team can ensure data safety and security. Robust encryption measures are needed which can ensure data security and privacy, effectively (Mahfuz, 2016). Data security and privacy concerns are on top priority among researchers and practitioners. It's imperative to consider how customer information is handled with safety and security. Generative AI models open new challenges and possibilities in SQA testing. From encryption

protocols to ethical considerations, organizations must ensure data security and privacy when undergoing deployment. Understanding the complexities in data security can ensure the privacy of data.

Encryption of data is important during transmission and storage. Data should be encrypted using secure protocols like Transport Layer Security (TLS) or Secure Sockets Layer (SSL). This can ensure safety and privacy of data. Further, it can prevent the data from unauthorized access. Additionally, training of data sets is also important in the testing of SQA. Strict controls on data accessing are essential for managing the testing phases of software. Therefore, implementation and ensure security of data is important (Mistrík et al., 2015). Secure deployment of encryption helps in protecting the data (Naik and Tripathy, 2015). The ability of AI models to identify patterns in historical data can enable SQA team to detect bugs at early stages of software development cycles (Veeramachaneni, 2021). Development of software testing can be more advanced using Generative AI models (Barenkamp et al., 2020). AI models can automate software testing by analyzing the past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance security testing by identifying the patterns (Song et al., 2022).

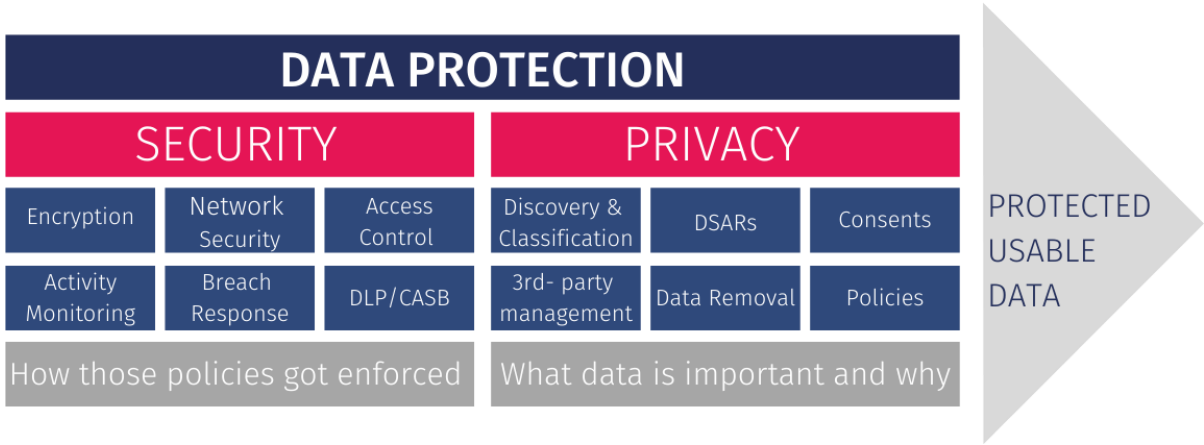


Figure 2.5. Role of data privacy and security (Data privacy manager, 2024)

Fig. 2.5 shows the role of data privacy and security. Data minimization is related with the minimum requirement of data necessary for training. This principle helps to mitigate the privacy risk of data (Natarajan, 2020). Thereby, avoiding unnecessary data for training. The threat of data breaches or unauthorized access can also be reduced by using this approach. Therefore, robust monitoring and logging mechanisms is essential for tracking the data security and privacy issues (Smith, 2024). By implementing such mechanisms SQA team can better handle the data privacy

and security issues. The mitigation of security breaches is essential as it affects the performance of software (Nama, 2024).

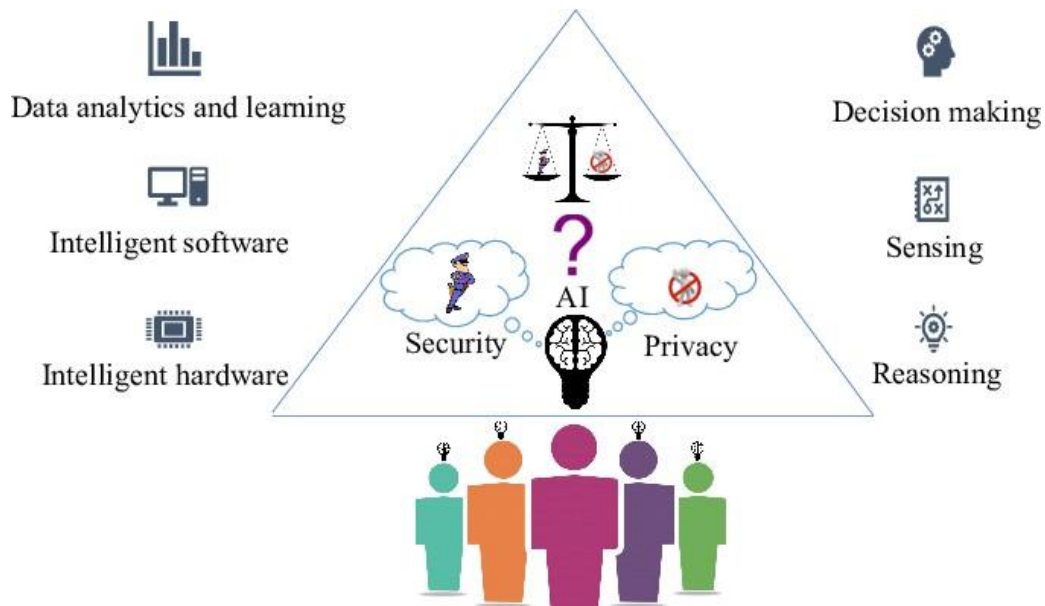


Figure 2.6. **Role of data privacy and security in AI (Li et al., 2020)**

Figure 2.6 shows the role of data privacy and security in AI. The generative AI models use past data and benchmark the performance of software development (Chinamanagonda, 2021). In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance (Neto et al., 2022). Generative AI models can lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. Generative AI models can expand the testing capabilities (Poth et al., 2022). Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. Future automated software testing can be shaped by using generative AI techniques.

## 2.4 Survey analysis

Present study is undertaken to investigate the current state of QA automation and AI-driven testing solutions. Further, identify challenges in implementing AI-driven QA automation were also reported in the study. Additionally, current study also develops a project management framework to address these challenges. To achieve these, current study relies on quantitative research method. Present study uses survey method and collects data from employees working in IT companies in

India. The employees working in the IT companies in Kerala, India are the target population of the study. The survey is conducted online using Google form. The questionnaires were distributed to the employees after obtaining consent from them. The selection of respondents is based on random sampling techniques. Overall, 152 responses were collected using random sampling techniques. The collected data were analyzed using SPSS software v23. The survey results were discussed and presented below:

#### Gender

152 responses

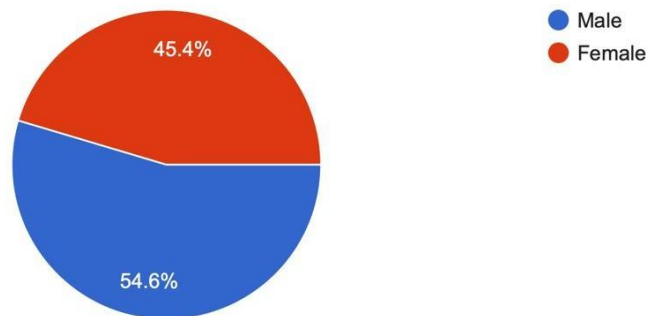


Figure 2.7. Percentage distribution of gender

From the fig. 2.7, it is observed that 54.6% are male and 45.4% are female.

#### Age

152 responses

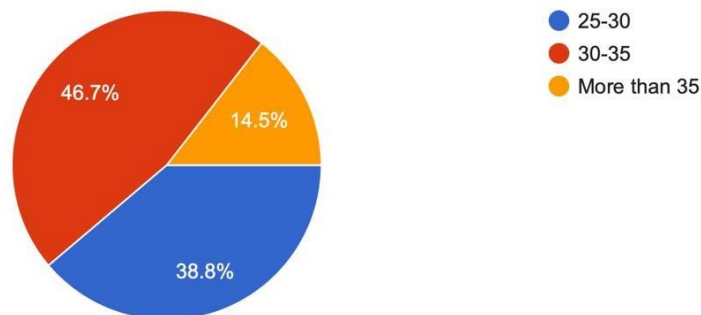


Figure 2.8. Percentage distribution of age

From the fig. 2.8, it is observed that 38.8% respondents have age group of 25-30, 46.7% respondents have age group of 30-35 and 14.5% of respondents have age group of more than 35.

Job position  
152 responses

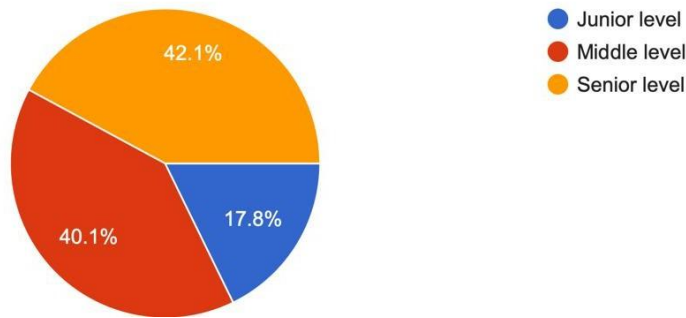


Figure 2.9. Percentage distribution of job position

From the fig. 2.9, it is observed that 17.8% respondents are at junior level, 40.1% respondents are at middle level and 42.1% respondents are at senior level.

Job experience  
151 responses

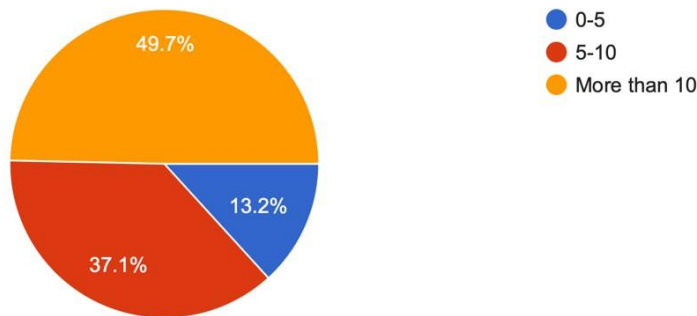


Figure 2.10. Percentage distribution of job experience

From the fig. 2.10, it is observed that 13.2% respondents have job experience of 0-5 years, 37.1% respondents have job experience of 5-10 and 49.7% respondents have job experience of more than 10.

For testing, my company has implemented QA automation

152 responses

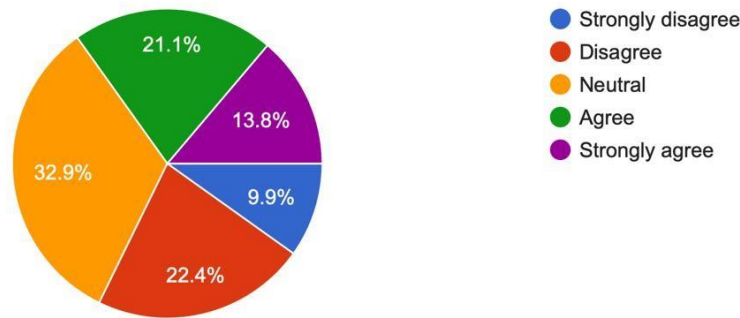


Figure 2.11. **Percentage distribution of implementation of QA automation**

From the fig. 2.11, it is observed that 9.9% respondents are strongly disagreed, 22.4% respondents are disagreed, 32.9% respondents are neutral, 21.1% respondents are agreed, and 13.8% respondents are strongly agreed. Modern software development has been transformed by automated testing using AI tools. These tools make the testing and development easier. All tests have been done by automation. Developers are now automated their software testing. This enables to enhance the automation process. In automation, various testing of modules of software has been done. AI technologies in QA Automation help to automate testing at different stages of software development. QA automation using AI technologies have transformed the testing automation. But due to high cost of automated QA software, many companies have not implemented it.

Consumer demand is growing rapidly while software gets ever more complicated. Although, software developers are under pressure to meet the demand of customers. The software program quality can be enhanced by using AI tools. Many big players have already employed automated testing to expedite the quality standards. Using QA automation SQA team can enhance the quality standards. Automating repetitious tasks can reduce the manual SQA team efforts. QA automation technologies are growing rapidly. AI Algorithms are used to analyze the performance of the products by gathering the real-time data. This helps the developers to identify the issues before they arise. In today's dynamic business landscape, testing is integral to QA. When the right tools are at right place, it simplifies the QA processes. QA team with advance expertise using AI tools can identify the early defect.

Testing of the product can reduces the cost and time to rework. This will strengthen the customer-company relationship. In a software company, the testing phase starts at the development stage and continues till the software is deployed. However, it also depends on the customer

requirement. For example, customer is interested in the testing of software at the multiple stages. On the other hand, testing of software at different phases can meet the expectations of customers. The software application is tested at the end of the project can ensure its quality and performance which can meet the needs of customers. The software development life cycle is critical as it address the critical aspects of performance.

Testing at each stage of the software development life cycle must be done. Testing of software can be performed both manually and by using automation tools. The QA process of software is divided into various groups such as software security, performance, and usability. But in the broad sense, they can be divided into manual and automated testing phases. For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software. These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods. Training an AI model is essential, and biasness can affect the data outcomes. So, organizations must reduce biases The mitigation of data biases is important as it can affect the decision-making. Organizations can mitigate data biases during the testing phases.

Our current project needs the implementation of QA automation tools  
152 responses

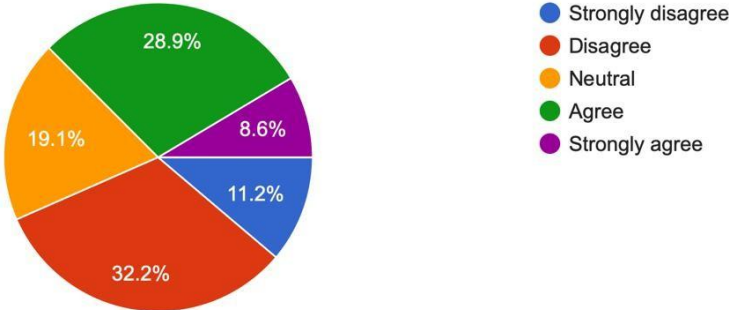


Figure 2.12. Percentage distribution of implementation of QA automation

From the fig. 2.12, it is observed that 11.2% respondents are strongly disagreed, 32.2% respondents are disagreed, 19.1% respondents are neutral, 28.9% respondents are agreed, and 8.6% respondents are strongly agreed. Automation of QA testing has revolutionized the software development process. It ensures that given criteria and performs as expected on several platforms

and environments. Furthermore, the use of automated tools and frameworks to run tests on a software application help to enhance the QA standards. Moreover, QA automation AI technologies enhance the software development stages. QA automation has been increasingly important in this competitive world. Transition from manual testing to automated testing helps to increase the efficiency of QA automation.

SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

Every organization must ensure the delivery of a high-quality product or service to their customers. Therefore, it is mandatory to check the quality of the product or application throughout the development process. In software development cycle, developers must ensure the performance of software. This can meet the quality standards of the products. QA teams can identify errors and bugs to ensure the software performance. Therefore, QA is crucial in the software development lifecycle. Its main role is beyond the software testing program is meeting exceeding the customer's expectations. By implementing SQA firm can better compete and differentiate themselves in the market. Overall, SQA is essential in the industry for meeting the demand of customers.

SQA focuses on identifying the root cause of defects and errors. The second principle is continuous improvement. In this phase, continuous improvement is used to consistently monitor the software product. Therefore, continuous improvement is essential for improving the performance of software products. The third principle is stakeholders' involvement. The involvement of stakeholders in the software development process is essential as it improve the performance of software product. This phase also highlights the importance of collaboration and communication between the company and customer. By effective involvement of stakeholders can help to defect the issues in the software product. SQA teams can work and collaborate with customers and identify their needs and simultaneously mitigate defects. SQA team use tools for the detection of issues and enhance the capability of software development process. Last, principle is risk-based approach. SQA team take risk and innovate software development process. The risk-taking behavior is essential as it enhance the software development process.

Therefore, companies must improve the quality of software products by using AI technologies. Every software development project ensures high quality and reliability. In QA

automation AI technologies take care of quality standards at every stage of development. Modern software development greatly depends on QA automation. By effective implementation, SQA team ensure high-quality software products which is faster and more reliable. Therefore, automation tools let companies penetrate the market, simplify testing procedures, and enhance product quality. Automated test scripts driven by QA automation can innovate software development process. Therefore, continuous improvement is essential for improving the performance of software products.

Our team is familiar with AI-driven testing tools and frameworks

152 responses

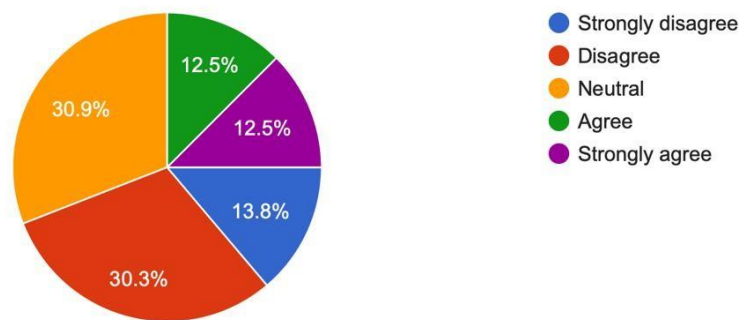


Figure 2.13. **Percentage distribution of team is familiar with AI-driven testing tools and frameworks**

From the fig. 2.13, it is observed that 13.8% respondents are strongly disagreed, 30.3% respondents are disagreed, 30.9% respondents are neutral, 12.5% respondents are agreed, and 12.5% respondents are strongly agreed. AI based testing is a subset of software testing which can automate testing across different stages of software development cycle. It establishes and maintain standards requirements while manufacturing of reliable and quality products. A well QA process enhance customer satisfaction and build company's credibility and brand image. SQA systematically identifies patterns, and the actions and improve development cycles involved in software development. However, finding and fixing coding errors are beneficial during the process of SQA. It is become essential for software developers to mitigate errors before they occur, this can save software development time and expenses involved in software. These errors are known as bugs. These have a significant impact on the performance of software.

AI testing aims to automate tasks such as test execution, data validation, and bugs finding. Using AI capabilities, SQA team can improve the SQA process. The manual testing of software poses several difficulties which are time consuming. AI based QA automation, testing process has

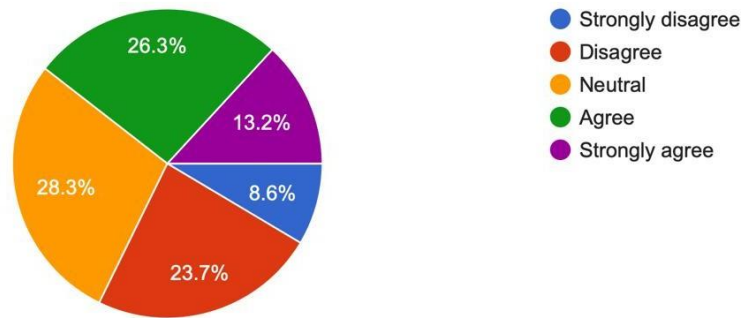
been enhanced. Bugs identification can easily be done. SQA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA. It is broader than testing; it involves testing as well as improvement processes. Meanwhile, testing is oriented to identify issues while improvement process relates with the improvement in the process. The quality of products and services is an important indicator which differentiates firm from their competitors. SQA ensures that product meet customer demands and expectations. High-quality products enhance brand image, which leads to the development of customer loyalty.

In SQA, AI testing marks has growing rapidly. The fundamental concept is to enhance software testing by means of AI technology and algorithms. This seeks to simplify, improve accuracy, and efficiency of testing. By automating testing, the firm can enhance the quality standards. SQA also reduces the cost of repairing of defective products and satisfy customers because SQA can early detect issues in the products. Defective products increase the cost of the product and diminish the brand image. It also increases the additional service cost to rectify the defects. Therefore, for streamlining the testing process, it requires additional investment. In the regard, the team members involved in testing process must define a process workflow and implement QA testing tools. However, the implementation involves a time-consuming process but when implemented it offers several benefits. If QA testing is not implemented, more serious issues can be generated which can increase the cost.

Recently, the use of AI and machine learning has gained significant attention in QA process improvement. In digital environment, AI and machine learning are used to inspect the products and machinery. This process is called predictive maintenance. AI Algorithms are used to analyze the performance of the products by gathering real-time data. This helps the developers to identify the issues before they arise. In today's dynamic business landscape, testing is integral to QA. When the right tools are at the right place, it simplifies the QA processes. QA team with advance expertise using AI tools can identify the early defect.

The productivity of testing can also increase by adopting AI tools. While traditional testing is limited to the complexity of software. SQA team can now deal with complexities and enhance the performance of testing. This not only improves the testing capabilities but also offer flexibility and efficiency in SQA process. While GenAI has the potential to deal with complexity, it can also enhance the testing procedures. Many testing tasks at different levels can be automated. However, data biases can reduce the efficiency of testing of software. Therefore, organizations need to mitigate the data biases. By doing so, SQA team can maximize the benefits of GenAI.

The efficiency of our software testing has been improved by AI-based testing  
152 responses



**Figure 2.14 Percentage distribution of efficiency of software testing has been improved by AI-based testing**

From the fig. 2.14, it is observed that 8.6% respondents are strongly disagreed, 23.7% respondents are disagreed, 28.3% respondents are neutral, 26.3% respondents are agreed, and 13.2% respondents are strongly agreed. Testing of the product can reduce the cost and time to rework. This will strengthen the customer-company relationship. In a software company, the testing phase starts at the development stage and continues till the software is deployed. However, it also depends on the customer requirement. For example, customer is interested in the testing of software at the multiple stages. On the other hand, testing of software at different phases can meet the expectations of customers. The software application is tested at the end of the project can ensure its quality and performance which can meet the needs of customers. The software development life cycle is critical as it address the critical aspects of performance.

Testing at each stage of the software development life cycle must be done. Testing of software can be performed both manually and by using automation tools. The QA process of software is divided into various groups such as software security, performance, and usability. But in the broad sense, they can be divided into manual and automated testing phases.

SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

Every organization must ensure the delivery of a high-quality product or service to their customers. Therefore, it is mandatory to check the quality of the product or application throughout the development process. In software development cycle, developers must ensure the performance of software. This can meet the quality standards of the products. QA teams can identify errors and bugs to ensure the software performance. Therefore, QA is crucial in the software development lifecycle. Its main role is beyond the software testing program is meeting exceeding the customer's expectations. BY implementing SQA firm can better compete and differentiate themselves in the market. Overall, SQA is essential in the industry for meeting the demand of customers. AI testing aims to automate tasks such as test execution, data validation, and bugs finding. AI capabilities can enhance the efficiency of software testing has been improved by AI-based testing. The manual testing of software poses several difficulties which are time consuming. On the other hand, AI based QA automation, testing process has been enhanced. Bugs identification can easily be done. SQA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA. It is broader than testing; it involves testing as well as improvement processes.

Our company has integrated both manual and automated testing effectively  
152 responses

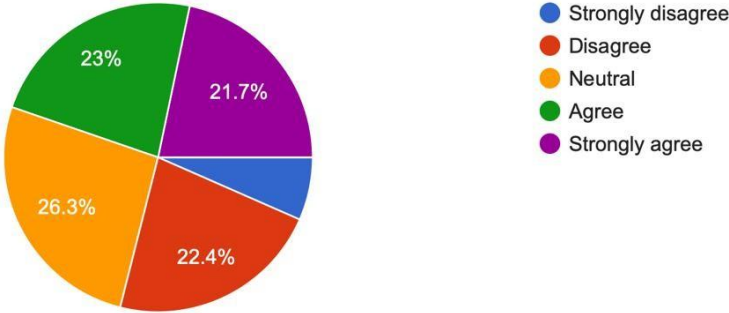


Figure 2.15. **Percentage distribution of company has integrated both manual and automated testing effectively**

From the fig. 2.15, it is observed that 6.6% respondents are strongly disagreed, 22.4% respondents are disagreed, 26.3% respondents are neutral, 23% respondents are agreed, and 21.7% respondents are strongly agreed. The goal is to enhance the performance of software development. AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhance testing performance. AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance the testing

productivity. AI models can predict complexity associated with the bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts. The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development.

The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generat test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards.

The importance of detection of bugs is associated with the performance of software. QA is essential to achieving the performance of software. As technology advances, AI tools are now used for improving the software quality. By using AI tools, SQA team can better focus on the performance of software. Future of use of AI in test automation can transform various industries. The fast-developing subject of artificial intelligence presents almost endless opportunities for application in test automation. AI algorithms becoming more complex with every year that goes by, which produces sophisticated intelligent test automation systems. Even if many AI test automation systems are still in their early years, there is unquestionable transformative potential. Imagine gadgets driven by artificial intelligence that learn and adapt to sophisticated software behaviors in addition to automating chores. This can result in AI-driven test prioritizing depending on risk assessment and user impact or self-healing tests automatically adjusting to UI changes.

Future of AI in test automation looks promising to transform the manual approach of software testing to automated approach. It produces a more effective results, and more advanced testing procedure. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By

analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

We are actively exploring or adopting AI solutions in our QA strategy

152 responses

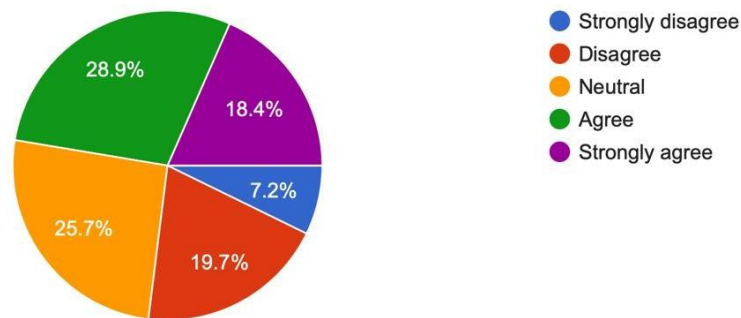


Figure 2.16. **Percentage distribution of actively exploring or adopting AI solutions in QA strategy**

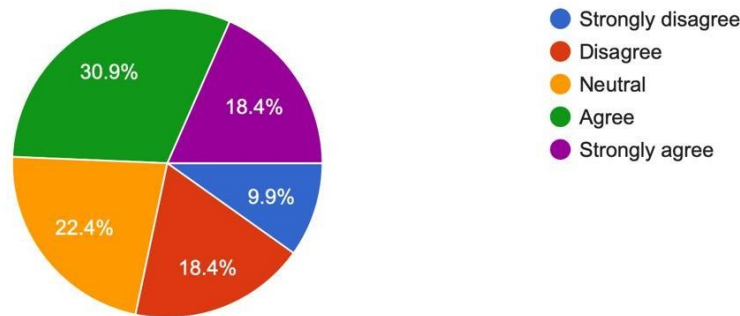
From the fig. 2.16, it is observed that 7.2% respondents are strongly disagreed, 19.7% respondents are disagreed, 25.7% respondents are neutral, 28.9% respondents are agreed, and 18.4% respondents are strongly agreed. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data. Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system. Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system.

Utilizing AI models for quality assurance can allow SQA teams to analyze the software quality standards. Improving the software quality is essential for achieving highest performance. By employing AI models, QA teams seek to identify defects in the software, hence, lowering risks is essential for optimized performance. The deep learning models integrating with neural networks can enhance the testing performance of software. This can reduce the workload of SQA teams and improve testing performance. Further, software developers do not need to create many data rules to identify the bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. AI capabilities can enhance the efficiency of software testing has been improved by AI-based testing. The manual testing of software poses several difficulties which are time consuming. On the other hand, AI based QA automation, testing process has been enhanced. Bugs identification can easily be done. SQA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA. It is broader than testing; it involves testing as well as improvement processes.

The quality of data used in AI-based testing significantly impacts the accuracy and effectiveness of the testing results. AI algorithms rely on vast amounts of data, AI models learn, make predictions, and detect patterns. Poor data quality can present biased outcomes, leading to inaccurate conclusions. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards.

## AI-driven testing has helped us reduce bugs and defects in production

152 responses



**Figure 2.17. Percentage distribution of AI-driven testing has helped in reducing bugs and defects in production**

From the fig. 2.17, it is observed that 9.9% respondents are strongly disagreed, 18.4% respondents are disagreed, 22.4% respondents are neutral, 30.9% respondents are agreed, and 18.4% respondents are strongly agreed. Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

AI models for software testing is gaining rapid growth. By employing generative AI in software testing, SQA process can be enhanced. It further automates the testing procedures and simultaneously boost the testing quality. Generative AI models are more suited for software development. This ensures that AI testing can enhance the testing methods. Therefore, SQA teams can enhance the process of testing. AI tools can automate the testing methods using generative AI models. This can help in the identification of bugs. Generative AI models allow to visualize the testing data. This can enable to meet the customer requirements. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of

generative AI in software testing, SQA process can be optimized. This can bring more opportunities for testing and improving the software quality. AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data. Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system. Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

Our organization regularly evaluates new AI tools for potential integration in QA automation  
152 responses

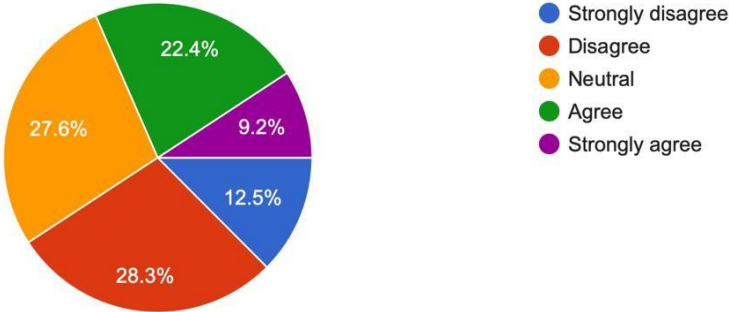


Figure 2.18. **Percentage distribution of organization regularly evaluates new AI tools for potential integration in QA automation**

From the fig. 2.18, it is observed that 12.5% respondents are strongly disagreed, 28.3% respondents are disagreed, 27.6% respondents are neutral, 22.4% respondents are agreed, and 9.2% respondents are strongly agreed. Integration of generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

By employing generative AI in software testing, SQA process can be enhanced. It further automates the testing procedures and simultaneously boost the testing quality. Generative AI models are more suited for software development. This ensures that AI testing can enhance the testing methods. Therefore, SQA teams can enhance the process of testing. AI tools can automate the testing methods using generative AI models. This can help in the identification of bugs. Generative AI models allow to visualize the testing data. This can enable to meet the customer requirements. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized. This can bring more opportunities for testing and improving the software quality. AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

It is always better to prevent defects and errors in the early development stages of software. Therefore, emphasizes is on the identification of potential issues in the software development lifecycle. SQA focuses on identifying the root cause of defects and errors. The continuous

improvement is used to consistently monitor the software product. Therefore, continuous improvement is essential for improving the performance of software product. The third principle is stakeholders' involvement. The involvement of stakeholder in the software development process is essential as it improve the performance of software product. This phase also highlights the importance of collaboration and communication between the company and customer. By effective involvement of stakeholders can help to defect the issues in the software product. SQA teams can work and collaborate with customers and identify their needs and simultaneously mitigate defects. SQA team use tools for the detection of issues and enhance the capability of software development process. Last, principle is risk-based approach. SQA team take risk and innovate software development process. The risk-taking behavior is essential as it enhance the software development process.

The return on investment (ROI) from QA automation is satisfactory  
151 responses

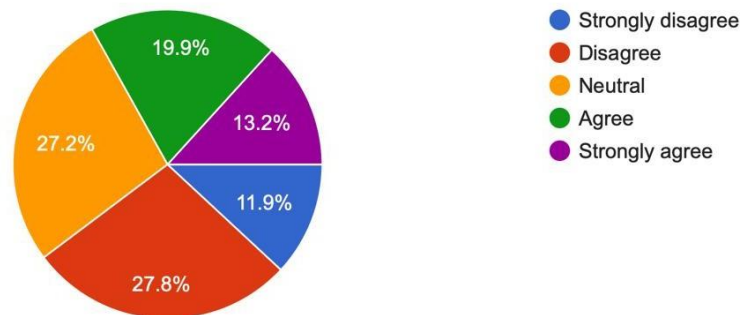


Figure 2.19. **Percentage distribution of return on investment (ROI) from QA automation is satisfactory**

From the fig. 2.19, it is observed that 11.9% respondents are strongly disagreed, 27.8% respondents are disagreed, 27.2% respondents are neutral, 19.9% respondents are agreed, and 13.2% respondents are strongly agreed. QA automation processes offer several benefits with impressive ROI. By identifying bugs in the software development cycle, SQA process can be enhance. This can help to improve the testing process and quality standards. Furthermore, by integrating AI into QA automation, bugs can easily be identified. This supports the agile framework of software development which are faster and reliable.

AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhances testing performance. AI can analyze historical data

and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance the testing productivity. AI models can predict complexity associated with the bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts. The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development.

The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards. The importance of detection of bugs is associated with the performance of software. QA is essential to achieving the performance of software. As technology advances, AI tools are now used for improving the software quality. By using AI tools, SQA team can better focus on the performance of software.

Variety of test data can be possible in the generation of test data. By learning from patterns, AI models can predict the behavior of system. Utilizing AI models for quality assurance can allow SQA teams to analyze the software quality standards. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. With the evaluation of generative AI in software testing, SQA process can be optimized. AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models can study software code or system behaviors based on gathered data over a time. AI models can automate the testing procedure and performance of software. Generative AI models can improve the performance of software testing by automation. The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages. AI models can test the complexity in the codes.

## AI-driven testing aligns well with our agile or DevOps development methodologies

152 responses

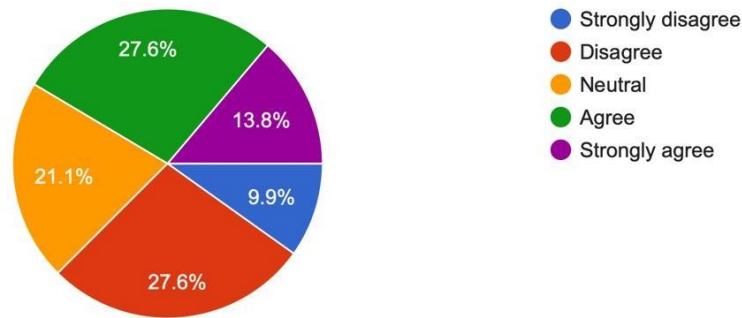


Figure 2.20. **Percentage distribution of AI-driven testing aligns well with agile or DevOps development methodologies**

From the fig. 2.20, it is observed that 9.9% respondents are strongly disagreed, 27.6% respondents are disagreed, 21.1% respondents are neutral, 27.6% respondents are agreed, and 13.8% respondents are strongly agreed. For agile software development, automated testing offers various advantages like faster feedback, more coverage, less costs, and improved quality. SQA systematically identifies patterns, and the actions and improve development cycles involved in software development. However, finding and fixing coding errors are beneficial during the process of SQA. It is become essential for software developers to mitigate errors before they occur, this can save software development time and expenses involved in software. These errors are known as bugs. These have a significant impact on the performance of software. QA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA. Testing of the product can reduces the cost and time to rework. This will strengthen the customer-company relationship.

Testing at each stage of the software development life cycle must be done. Testing of software can be performed both manually and by using automation tools. The QA process of software is divided into various groups such as software security, performance, and usability. But in the broad sense, they can be divided into manual and automated testing phases. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

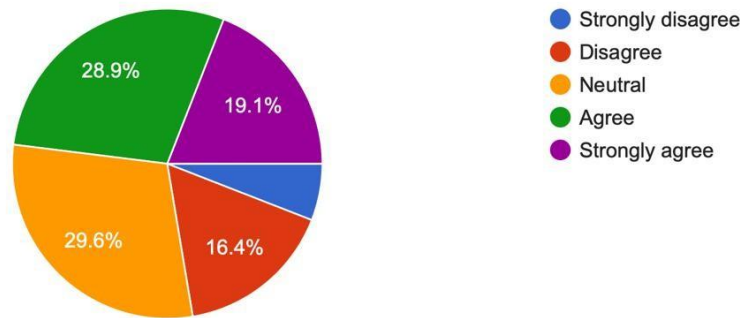
Every organization must ensure the delivery of a high-quality product or service to their customers. Therefore, it is mandatory to check the quality of the product or application throughout the development process. In software development cycle, developers must ensure the performance of software. This can meet the quality standards of the products. QA teams can identify errors and bugs to ensure the software performance. Therefore, QA is crucial in the software development lifecycle. Its main role is beyond the software testing program is meeting exceeding the customer's expectations. BY implementing SQA firm can better compete and differentiate themselves in the market. Overall, SQA is essential in the industry for meeting the demand of customers.

SQA focuses on identifying the root cause of defects and errors and continuous improvement. SQA teams can work and collaborate with customers and identify their needs and simultaneously mitigate defects. SQA team use tools for the detection of issues and enhance the capability of software development process. AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhance testing performance. AI can analyze historical data and detect issues at early stages. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development.

The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards. The importance of detection of bugs is associated with the performance of software. QA is essential to achieving the performance of software. As technology advances, AI tools are now used for improving the software quality.

## Implementing AI-based testing tools requires advanced technical knowledge

152 responses



**Figure 2.21. Percentage distribution of implementing AI-based testing tools requires advanced technical knowledge**

From the fig. 2.21, it is observed that 5.9% respondents are strongly disagreed, 16.4% respondents are disagreed, 29.6% respondents are neutral, 28.9% respondents are agreed, and 19.1% respondents are strongly agreed. Knowledge plays an important role in implementing AI-based testing tools. If integrated with SQA process, it can generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards. The importance of detection of bugs is associated with the performance of software. QA is essential to achieving the performance of software. As technology advances, AI tools are now used for improving the software quality. By using AI tools, SQA team can better focus on the performance of software.

AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhance testing performance. AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance the testing productivity. AI models can predict complexity associated with the bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts. The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on

routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development.

Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system. Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system.

Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software. These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods. The ability of AI models to identify patterns in historical data can enable SQA team to detect bugs at early stages of software development cycles. Development of software testing can be more advanced using Generative AI models. AI models can automate software testing by analyzing the past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns.

## Our existing systems are not compatible with AI testing tools

152 responses

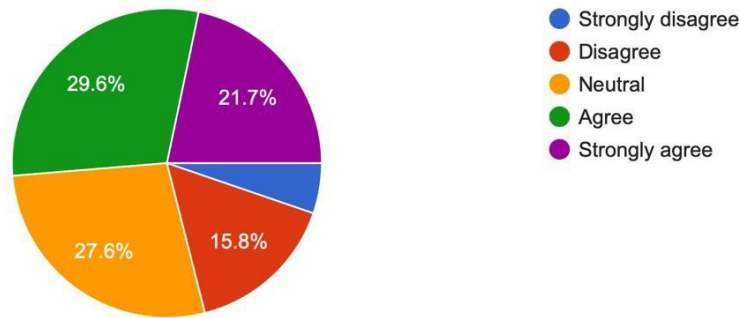


Figure 2.22. **Percentage distribution of existing systems is not compatible with AI testing tools**

From the fig. 2.22, it is observed that 5.3% respondents are strongly disagreed, 15.8% respondents are disagreed, 27.6% respondents are neutral, 29.6% respondents are agreed, and 21.7% respondents are strongly agreed. AI models for software testing is gaining rapid growth. By employing generative AI in software testing, SQA process can be enhanced. It further automates the testing procedures and simultaneously boost the testing quality. Generative AI models are more suited for software development. This ensures that AI testing can enhance the testing methods. Therefore, SQA teams can enhance the process of testing. AI tools can automate the testing methods using generative AI models. This can help in the identification of bugs. Generative AI models allow to visualize the testing data. This can enable to meet the customer requirements. Generative AI models can improve the performance of software testing by automation.

Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized. This can bring more opportunities for testing and improving the software quality. AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative

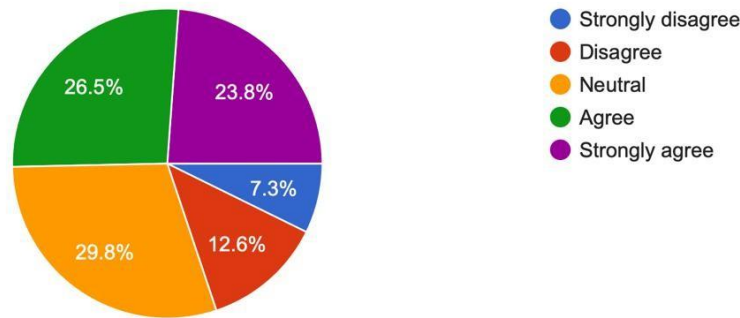
AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance. Generative AI models can lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. Generative AI models can expand the testing capabilities. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. This advancement can enhance the security testing by identifying the patterns. Cyberattacks are growing rapidly, therefore, AI models can better predict the testing data. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized.

Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data. Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system. Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system.

### Lack of proper documentation makes AI tool integration difficult

151 responses



**Figure 2.23. Percentage distribution of lack of proper documentation makes AI tool integration difficult**

From the fig. 2.23, it is observed that 7.3% respondents are strongly disagreed, 12.6% respondents are disagreed, 29.8% respondents are neutral, 26.5% respondents are agreed, and 23.8% respondents are strongly agreed. Lack of proper documentation makes AI tool integration difficult. Since, implementing AI tool can create difficulties even if it offers great benefits such as predictive analytics, data-driven insights, and faster bug discovery. The quality of data used in AI-based testing significantly impacts the accuracy and effectiveness of the testing results. AI algorithms rely on vast amounts of data, AI models learn, make predictions, and detect patterns. Poor data quality can present biased outcomes, leading to inaccurate conclusions. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards.

Therefore, proper documentation is crucial when implementing AI-based tools for the testing of software. By ensuring proper documentation, SQA team can predict the performance of the software. However, the high quality of data is not sufficient, continuous monitoring and maintenance of data is also essential. This will ensure the high quality of data over the time. Prioritizing data quality can be helpful for SQA team for achieving standard results. This will also help organizations to leverage the full potential of AI tools for testing. Also, generative AI models

depend on high-quality, and representative data which can ensure the quality standards of software. However, poor data quality can lead to inaccurate test cases and false outcomes. Generative AI software testing has the potential to boost performance, therefore, it is important for SQA team.

High-quality data can help firm to innovate the testing methods. It can also accelerate the progress of software testing. Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results. In essence, the positive impact of data quality on test methods can also be enhanced. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.

The quality of data used in AI-based testing significantly impacts the accuracy and effectiveness of the testing results. AI algorithms rely on vast amounts of data, AI models learn, make predictions, and detect patterns. Poor data quality can present biased outcomes, leading to inaccurate conclusions. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards.

The setup and maintenance of AI-driven QA solutions are complex

152 responses

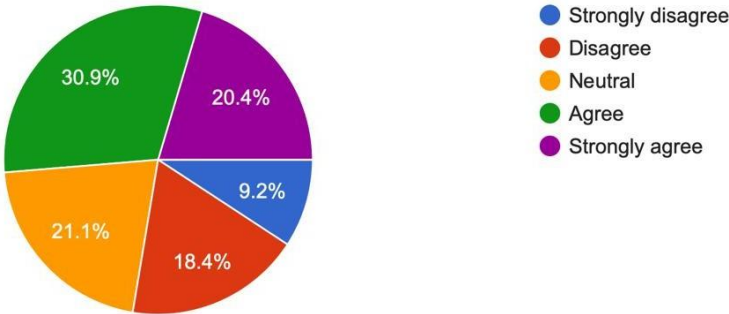


Figure 2.24. Percentage distribution of setup and maintenance of AI-driven QA solutions are complex

From the fig. 2.24, it is observed that 9.2% respondents are strongly disagreed, 18.4% respondents are disagreed, 21.1% respondents are neutral, 30.9% respondents are agreed, and 20.4% respondents are strongly agreed. The setup and maintenance of AI-driven QA solutions is important because it can impact decision making. For example, e-commerce companies are using AI tools for the recommendation of products to the customers based on their performance. If the data used for training the AI model is incomplete or inaccurate, the recommendations provided to customers may depart from the actual preferences. This can dissatisfy the customers and negatively affect the performance of the company. Moreover, the use AI tools in different sectors have been increased for critical decision-making. However, the impact of poor data quality on decision making is significant. Inaccurate or incomplete data can potentially lead to incorrect outcomes. This can negatively impact the organizations.

Therefore, setup and maintenance of AI-driven QA solutions can ensure the quality of data is crucial when implementing AI-based tools for the testing of software. By ensuring the quality of data, SQA team can predict the performance of the software. However, the high quality of data is not sufficient, continuous monitoring and maintenance of data is also essential. This will ensure the high quality of data over the time. Prioritizing data quality can be helpful for SQA team for achieving standard results. This will also help organizations to leverage the full potential of AI tools for testing. Also, generative AI models depend on high-quality, and representative data which can ensure the quality standards of software. However, poor data quality can lead to inaccurate test cases and false outcomes. Generative AI software testing has the potential to boost performance, therefore, it is important for SQA team.

Setup and maintenance of AI-driven QA solutions can help firm to innovate the testing methods. It can also accelerate the progress of software testing. Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results. In essence, the positive impact of data quality on test methods can also be enhanced. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.

This can reduce the workload of SQA teams and improve testing performance. Further, software developers do not need to create many data rules to identify the bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively. Generative AI models use test data and

analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software. These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods.

The ability of AI models to identify patterns in historical data can enable SQA team to detect bugs at early stages of software development cycles. Development of software testing can be more advanced using Generative AI models. AI models can automate software testing by analyzing the past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns.

Our QA team lacks the necessary skills to implement AI automation  
151 responses

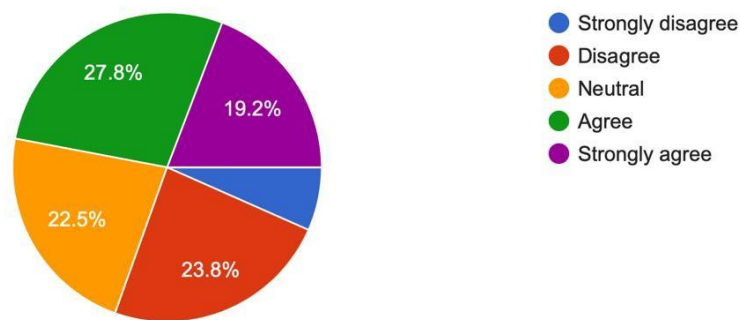


Figure 2.25. **Percentage distribution of QA team lacks the necessary skills to implement AI automation**

From the fig. 2.25, it is observed that 6.6% respondents are strongly disagreed, 23.8% respondents are disagreed, 22.5% respondents are neutral, 27.8% respondents are agreed, and 19.2% respondents are strongly agreed. QA team lacks the necessary skills to implement AI

automation. However, necessary skills to implement AI automation are the prerequisite for AI deployment. On the other hand, high licensing fees also hinder the adoption. In the past, software testing has been done by using traditional methods. On the other hand, generative AI models are advancing the software testing. The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages. Development of software testing can be more advanced using Generative AI models. These models can automate the testing methods. AI models can leverage AI models and automate software testing by analyzing past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns. Cyberattacks are growing rapidly, therefore, AI models can better predict the testing data. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized.

The generative AI models use past data and benchmark the performance of software development. In this regard, necessary skills to implement AI automation is essential. In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance. Generative AI models can lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. Generative AI models can expand the testing capabilities. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

By learning from patterns, AI models can predict the behavior of system. Generative AI models generate test data which can be analyzed SQA teams. With the evaluation of generative AI in software testing, SQA process can be optimized. Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As

technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

AI-based testing systems often depends on the data quality. Further, raw data with insufficient information can make difficult for processing by AI model. Inconsistent quality of data can also an issue for processing the data using AI models. This can lead to incorrect predictions of the outcome. Therefore, preprocessing techniques are essential before proceeding to data analysis using AI tools. Additionally, monitoring and refinement of AI algorithms can also help SQA team to handle the data. In AI-based testing, data biases are also a major issue. Biased data can affect the outcomes and performance of the software. In this regard, training of data is essential to avoid the data biasness. Further, the biasness in the data can also affect the accuracy of software. Therefore, SQA team must continuously assess and mitigate biases, so that accurate outcomes can be achieved.

There is limited training available on AI-driven QA technologies  
152 responses

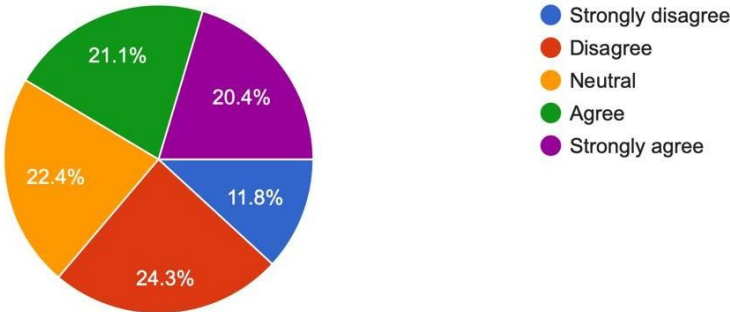


Figure 2.26. **Percentage distribution of limited training available on AI-driven QA technologies**

From the fig. 2.26, it is observed that 11.8% respondents are strongly disagreed, 24.3% respondents are disagreed, 22.4% respondents are neutral, 21.1% respondents are agreed, and 20.4% respondents are strongly agreed. From bug detection to test suite optimization, AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data. Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI

models SQA teams can analyze the system behavior based on test data. This enables automated testing of system. Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system.

AI-based software testing required skilled training. The goal is to enhance the performance of software development. AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhance testing performance. AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance the testing productivity. AI models can predict complexity associated with the bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts. The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development.

SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software. Furthermore, the capability of GenAI for testing at different phases of software development life cycles is also important. Therefore, leveraging the full potential of GenAI for enhancing test practices can also improve the quality and reliability of software products. Therefore, organizations can leverage the full power of GenAI and enhance the testing performance. The evolution of generative AI is gaining significant attention in software testing. GenAI models can enhance the testing performance. It interprets data outcome and provide

measures for quality standards. This capability of AI models can enhance the testing and SQA of software testing. It simplifies the process of testing and automation by enabling the SQA. For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software. These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods.

Hiring experts in AI testing is a major challenge

152 responses

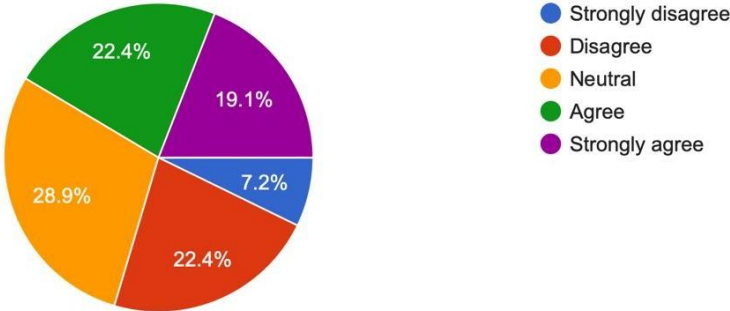


Figure 2.27. Percentage distribution of hiring experts in AI testing is a major challenge

From the fig. 2.27, it is observed that 7.2% respondents are strongly disagreed, 22.4% respondents are disagreed, 28.9% respondents are neutral, 22.4% respondents are agreed, and 19.1% respondents are strongly agreed. In the field of AI trained experts are also needed. Recently, the use of AI and ML has gaining significant attention in QA process improvement. In digital environment, AI and machine learning are used to inspect the products and machinery. This process is called predictive maintenance. AI Algorithms are used to analyze the performance of the products by gathering the real-time data. This helps the developers to identify the issues before they arise. In today’s dynamic business landscape, testing is integral to QA. When the right tools are at right place, it simplifies the QA processes. QA team with advance expertise using AI tools can identify the early defect.

Testing of the product can reduces the cost and time to rework. This will strengthen the customer-company relationship. In a software company, the testing phase starts at the development stage and continues till the software is deployed. However, it also depends on the customer requirement. For example, customer is interested in the testing of software at the multiple stages.

On the other hand, testing of software at different phases can meet the expectations of customers. The software application is tested at the end of the project can ensure its quality and performance which can meet the needs of customers. The software development life cycle is critical as it address the critical aspects of performance.

The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards.

From bug detection to test suite optimization, AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Variety of test data can be possible in the generation of test data. Additionally, it is also possible to generate the forecasted outcomes using generative AI techniques. Thereby, using AI models SQA teams can analyze the system behavior based on test data. This enables automated testing of system. Also, AI models are capable for test suite optimization. AI models can analyze past data and benchmark the performance of system. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. By gathering the system data, generative models can predict the performance of the system. It can also reduce the amount of manual work required to develop and test the system.

The quality of data used in AI-based testing significantly impacts the accuracy and effectiveness of the testing results. AI algorithms rely on vast amounts of data, AI models learn, make predictions, and detect patterns. Poor data quality can present biased outcomes, leading to inaccurate conclusions. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages. It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA

process, it can self-generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards.

#### Team members are hesitant to learn or adopt AI testing practices

151 responses

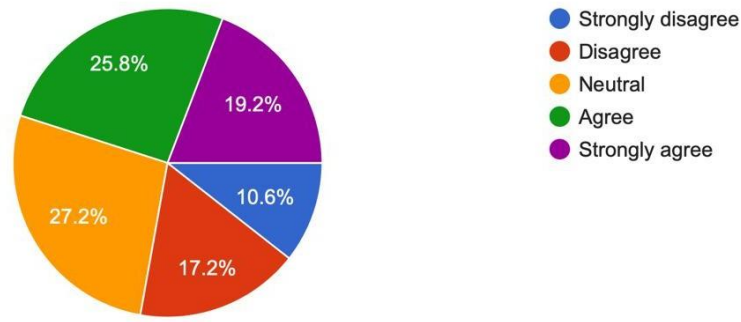


Figure 2.28. **Percentage distribution of team members is hesitant to learn or adopt AI testing practices**

From the fig. 2.28, it is observed that 10.6% respondents are strongly disagreed, 17.2% respondents are disagreed, 27.2% respondents are neutral, 25.8% respondents are agreed, and 19.2% respondents are strongly agreed. It is observed that many team members is hesitant to learn or adopt AI testing practices. Utilizing AI models for quality assurance can allow SQA teams to analyze the software quality standards. Improving the software quality is essential for achieving highest performance. By employing AI models, QA teams seek to identify defects in the software, hence, lowering risks is essential for optimized performance. Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

AI models for software testing is gaining rapid growth. By employing generative AI in software testing, SQA process can be enhanced. It further automates the testing procedures and simultaneously boost the testing quality. Generative AI models are more suited for software

development. This ensures that AI testing can enhance the testing methods. Therefore, SQA teams can enhance the process of testing. AI tools can automate the testing methods using generative AI models. This can help in the identification of bugs. Generative AI models allow to visualize the testing data. This can enable to meet the customer requirements. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized. This can bring more opportunities for testing and improving the software quality. AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

The generative AI models use past data and benchmark the performance of software development. In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance. Generative AI models can lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. Generative AI models can expand the testing capabilities. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results.

The deep learning models integrating with neural networks can enhance the testing performance of software. This can reduce the workload of SQA teams and improve testing performance. Further, software developers do not need to create many data rules to identify the bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using

generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

The cost of implementing AI in QA is too high  
152 responses

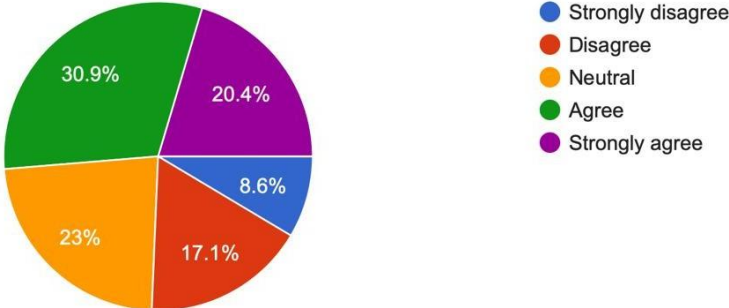


Figure 2.29. Percentage distribution of cost of implementing AI in QA is too high

From the fig. 2.29, it is observed that 8.6% respondents are strongly disagreed, 17.1% respondents are disagreed, 23% respondents are neutral, 30.9% respondents are agreed, and 20.4% respondents are strongly agreed. There are several benefits by the implementation of AI tools in the QA process. However, the cost of implementing AI in QA is too high. On the other hand, generative AI models are advancing the software testing. The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages. Development of software testing can be more advanced using Generative AI models. These models can automate the testing methods. AI models can leverage AI models and automate software testing by analyzing past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns. Cyberattacks are growing rapidly, therefore, AI models can better predict the testing data. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized but it can be costly for implementation.

The generative AI models use past data and benchmark the performance of software development. In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance. Generative AI models can

lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. Generative AI models can expand the testing capabilities. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results.

Generative AI models allow to visualize the testing data. This can enable to meet the customer requirements. Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized. This can bring more opportunities for testing and improving the software quality. AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

### Licensing AI testing tools exceeds our current budget

152 responses

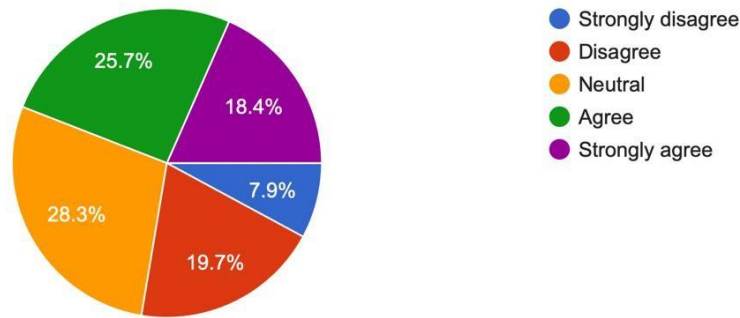


Figure 2.30. **Percentage distribution of licensing AI testing tools exceeds current budget**

From the fig. 2.30, it is observed that 7.9% respondents are strongly disagreed, 19.7% respondents are disagreed, 28.3% respondents are neutral, 25.7% respondents are agreed, and 18.4% respondents are strongly agreed. The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. Yet they exceeded the financial budget. If integrated with SQA process, it can generate test scenarios that cater the needs of customers. AI can make the SQA process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards. By effective involvement of stakeholders can help to defect the issues in the software product. SQA teams can work and collaborate with customers and identify their needs and simultaneously mitigate defects.

Testing at each stage of the software development life cycle must be done. Testing of software can be performed both manually and by using automation tools. The QA process of software is divided into various groups such as software security, performance, and usability. But in the broad sense, they can be divided into manual and automated testing phases. SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

QA relates with determining the meeting of quality of product or service with standard requirements. It establishes and maintain standards requirements while manufacturing of reliable

and quality products. A well QA process enhance customer satisfaction and build company's credibility and brand image. It also helps a company to achieve a competitive position in the market. The concept of QA is important in the manufacturing industry. The QA process is not limited to manufacturing industries, but it is also applicable to software engineering. SQA systematically identifies patterns, and the actions and improve development cycles involved in software development. However, finding and fixing coding errors are beneficial during the process of SQA. It is become essential for software developers to mitigate errors before they occur, this can save software development time and expenses involved in software. These errors are known as bugs. These have a significant impact on the performance of software.

Furthermore, the capability of GenAI for testing at different phases of software development life cycles is also important. Therefore, leveraging the full potential of GenAI for enhancing test practices can also improve the quality and reliability of software products. Therefore, organizations can leverage the full power of GenAI and enhance the testing performance. The evolution of generative AI is gaining significant attention in software testing. GenAI models can enhance the testing performance. It interprets data outcome and provide measures for quality standards. This capability of AI models can enhance the testing and SQA of software testing. It simplifies the process of testing and automation by enabling the SQA.

High-quality data can help firm to innovate the testing methods. It can also accelerate the progress of software testing. Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results. In essence, the positive impact of data quality on test methods can also be enhanced. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.

It is also observed that excessive reliance on AI models can be harmful, therefore, a balance of human judgment with AI capabilities is required for achieving the accurate results. With the emergence of data-driven testing, SQA teams can analyze the various data sets and predict the behavior of system performance. AI models offer several benefits to the SQA teams such as automation. Traditionally, software testing consumes a lot of time and often produces errors, therefore, require multiple iterations. Manual testing of software also involves several issues that can hinder the ability of SQA teams in delivering the high-quality products. However, generative

AI models for software testing revolutionize the process by automating. It can also optimize the testing methods which can accelerate the software development cycle.

ROI from AI-driven QA is unclear, making investment risky  
152 responses

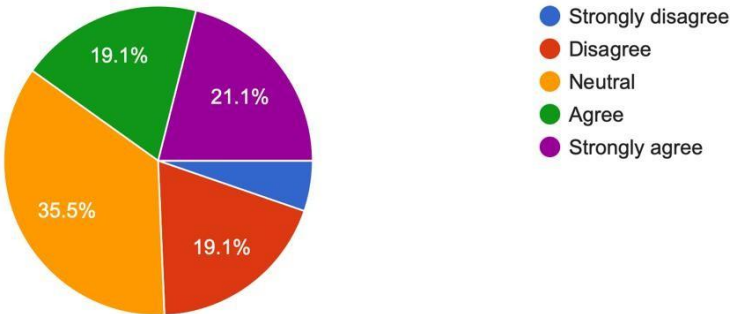


Figure 2.31. **Percentage distribution of ROI from AI-driven QA is unclear, making investment risky**

From the fig. 2.31, it is observed that 5.3% respondents are strongly disagreed, 19.1% respondents are disagreed, 35.5% respondents are neutral, 19.1% respondents are agreed, and 21.1% respondents are strongly agreed. ROI from AI-driven QA is unclear, making investment risky. However, companies all around are using AI to improve consumer experiences, streamline processes, and stimulate income generation. QA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA. QA is broader than testing; it involves testing as well as improvement processes. Meanwhile, testing is oriented for identifying issues while improvement process relates with the improvement in the process. The quality of products and services is an important indicator which differentiate firm from their competitors. QA ensures that product meet customer demands and expectations. High-quality products enhance brand image, which leads to the development of customer loyalty.

QA also reduces the cost of repairing of defective products and satisfy customers because QA can early detect issues in the products. Defective products increase the cost of the product and diminish the brand image. It also increases the additional service cost to rectify the defects. Therefore, for streamlining the QA process, it requires additional investment. In the regard, the team members involved in QA process must define a process workflow and implement QA tools. However, the implementation involves a time-consuming process but when implemented it offers several benefits. If QA is not implemented, more serious issues can be generated which can increase the cost.

Testing of the product can reduce the cost and time to rework. This will strengthen the customer-company relationship. In a software company, the testing phase starts at the development stage and continues till the software is deployed. However, it also depends on the customer requirement. For example, customer is interested in the testing of software at the multiple stages. On the other hand, testing of software at different phases can meet the expectations of customers. The software application is tested at the end of the project can ensure its quality and performance which can meet the needs of customers. The software development life cycle is critical as it address the critical aspects of performance. SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

Financial decision-makers are reluctant to allocate funds to AI QA tools

152 responses

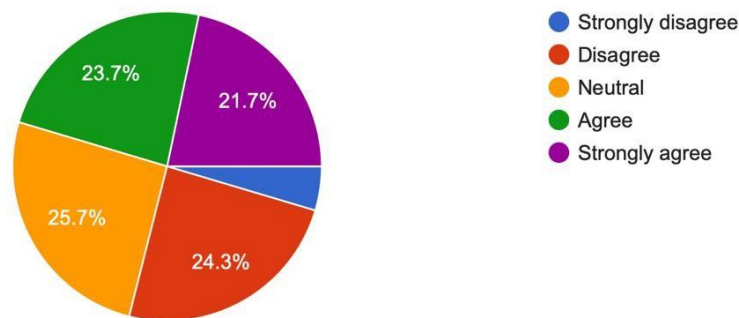


Figure 2.32. **Percentage distribution of financial decision-makers is reluctant to allocate funds to AI QA tools**

From the fig. 2.32, it is observed that 4.6% respondents are strongly disagreed, 24.3% respondents are disagreed, 25.7% respondents are neutral, 23.7% respondents are agreed, and 21.7% respondents are strongly agreed. Financial decision-makers is reluctant to allocate funds to AI QA tools. SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote

high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

Testing of the product can reduce the cost and time to rework. This will strengthen the customer-company relationship. In a software company, the testing phase starts at the development stage and continues till the software is deployed. However, it also depends on the customer requirement. For example, customer is interested in the testing of software at the multiple stages. On the other hand, testing of software at different phases can meet the expectations of customers. The software application is tested at the end of the project can ensure its quality and performance which can meet the needs of customers. The software development life cycle is critical as it address the critical aspects of performance.

QA is more focused on finding the bugs, the identification of bugs are essential during the process of SQA. QA is broader than testing; it involves testing as well as improvement processes. Meanwhile, testing is oriented for identifying issues while improvement process relates with the improvement in the process. The quality of products and services is an important indicator which differentiate firm from their competitors. QA ensures that product meet customer demands and expectations. High-quality products enhance brand image, which leads to the development of customer loyalty.

QA also reduces the cost of repairing of defective products and satisfy customers because QA can early detect issues in the products. Defective products increase the cost of the product and diminish the brand image. It also increases the additional service cost to rectify the defects. Therefore, for streamlining the QA process, it requires additional investment. In the regard, the team members involved in QA process must define a process workflow and implement QA tools. However, the implementation involves a time-consuming process but when implemented it offers several benefits. If QA is not implemented, more serious issues can be generated which can increase the cost.

Therefore, ensuring the quality of data is crucial when implementing AI-based tools for the testing of software. By ensuring the quality of data, SQA team can predict the performance of the software. However, the high quality of data is not sufficient, continuous monitoring and maintenance of data is also essential. This will ensure the high quality of data over the time. Prioritizing data quality can be helpful for SQA team for achieving standard results. This will also help organizations to leverage the full potential of AI tools for testing. Also, generative AI models depend on high-quality, and representative data which can ensure the quality standards of software.

However, poor data quality can lead to inaccurate test cases and false outcomes. Generative AI software testing has the potential to boost performance, therefore, it is important for SQA team.

Management lacks confidence in AI-driven QA solutions  
152 responses

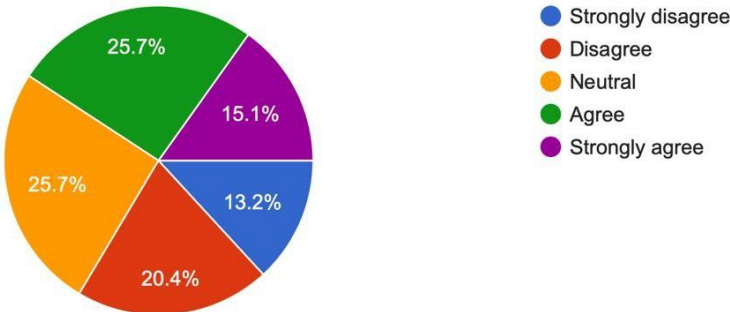


Figure 2.33. Percentage distribution of management lacks confidence in AI-driven QA solutions

From the fig. 2.33, it is observed that 13.2% respondents are strongly disagreed, 20.4% respondents are disagreed, 25.7% respondents are neutral, 25.7% respondents are agreed, and 15.1% respondents are strongly agreed. Management lacks confidence in AI-driven QA solutions. The use of AI and ML has gaining significant attention in QA process improvement. In digital environment, AI and machine learning are used to inspect the products and machinery. This process is called predictive maintenance. AI Algorithms are used to analyze the performance of the products by gathering the real-time data. This helps the developers to identify the issues before they arise. In today’s dynamic business landscape, testing is integral to QA. When the right tools are at right place, it simplifies the QA processes. QA team with advance expertise using AI tools can identify the early defect.

SQA ensure the software quality throughout its development lifecycle. The SQA process are designed to enhance the software performance. Developers are now more interested in identifying the early defect in the software. Developers develop code, therefore, reviewing of code is critical in SQA activity. This practice aims to detect errors in the early stages. This can promote high quality standard with error free from the code. Automated software testing uses software tools to and detect errors in the software.

Every organization must ensure the delivery of a high-quality product or service to their customers. Therefore, it is mandatory to check the quality of the product or application throughout the development process. In software development cycle, developers must ensure the performance

of software. This can meet the quality standards of the products. QA teams can identify errors and bugs to ensure the software performance. Therefore, QA is crucial in the software development lifecycle. Its main role is beyond the software testing program is meeting exceeding the customer's expectations. BY implementing SQA firm can better compete and differentiate themselves in the market. Overall, SQA is essential in the industry for meeting the demand of customers. AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhance testing performance.

AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance the testing productivity. AI models can predict complexity associated with the bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts. The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development.

Generative AI in software testing has received a lot of attention. In contrast to traditional systems that rely on manual methods, generative AI employs ML techniques and enhance the performance of the system. AI models can identify patterns, identifying anomalies, and potential issues which can affect the system performance. This method enables to detect issues at early stages. SQA teams can take preventative measures using AI models. Additionally, Generative AI for software testing can automate the testing procedure. As technology advances, the incorporation of generative AI into software development has the potential to streamline the testing methods. Further, it also allows for the development of software products that can be user-friendly and reliable.

AI tools can improve the software quality. AI tools can detect bug and enhance the quality of software. Generative AI models use test data and analyze the performance of software. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data

over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

Team members fear that AI automation might replace human roles  
151 responses

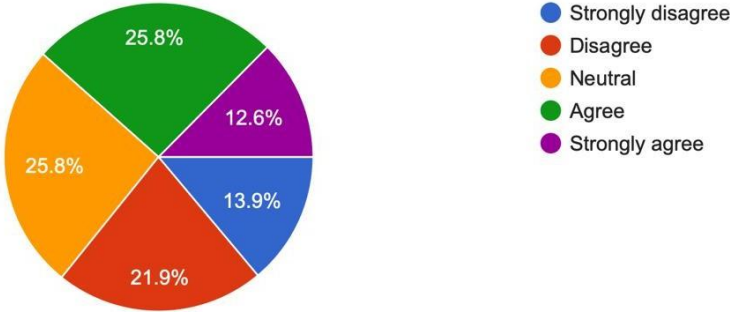


Figure 2.34. **Percentage distribution of team members fear that AI automation might replace human roles**

From the fig. 2.34, it is observed that 13.9% respondents are strongly disagreed, 21.9% respondents are disagreed, 25.8% respondents are neutral, 25.8% respondents are agreed, and 12.6% respondents are strongly agreed. Team members fear that AI automation might replace human roles. In the past, software testing has been done by using traditional methods. On the other hand, generative AI models are advancing the software testing. The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages. Development of software testing can be more advanced using Generative AI models. These models can automate the testing methods. AI models can leverage AI models and automate software testing by analyzing past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns. Cyberattacks are growing rapidly, therefore, AI models can better predict the testing data. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized.

The ability of AI models to identify patterns in historical data can enable SQA team to detect the issues or bugs at early stages. Development of software testing can be more advanced using Generative AI models. AI models can leverage AI models and automate software testing by analyzing past historical data. This can enhance the performance of software using automation. AI

models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns. Cyberattacks are growing rapidly, therefore, AI models can better predict the testing data. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results. With the evaluation of generative AI in software testing, SQA process can be optimized. The data quality is important because it can impact decision making. For example, e-commerce companies are using AI tools for the recommendation of products to the customers based on their performance. If the data used for training the AI model is incomplete or inaccurate, the recommendations provided to customers may depart from the actual preferences. This can dissatisfy the customers and negatively affect the performance of the company. Moreover, the use AI tools in different sectors have been increased for critical decision-making. However, the impact of poor data quality on decision making is significant. Inaccurate or incomplete data can potentially lead to incorrect outcomes. This can negatively impact the organizations.

For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software. These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods.

Organizational culture does not support AI innovation in QA  
147 responses

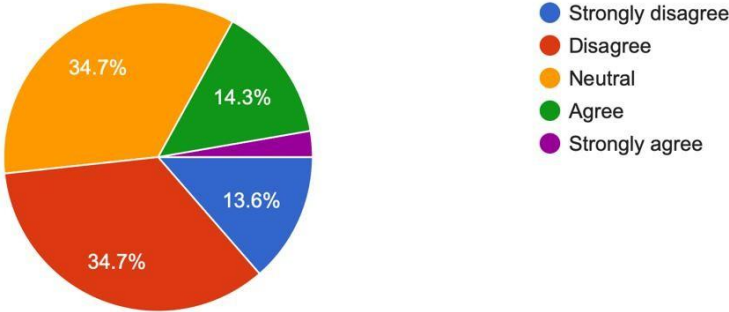


Figure 2.35 Percentage distribution of organizational culture does not support AI innovation

From the fig. 2.35, it is observed that 13.6% respondents are strongly disagreed, 34.7% respondents are disagreed, 34.7% respondents are neutral, 14.3% respondents are agreed, and 2.7%

respondents are strongly agreed. Organizational culture does not support AI innovation in QA. AI-based testing systems often depends on the data quality. Further, raw data with insufficient information can make difficult for processing by AI model. Inconsistent quality of data can also an issue for processing the data using AI models. This can lead to incorrect predictions of the outcome. Therefore, preprocessing techniques are essential before proceeding to data analysis using AI tools. Additionally, monitoring and refinement of AI algorithms can also help SQA team to handle the data. In AI-based testing, data biases are also a major issue. Biased data can affect the outcomes and performance of the software. In this regard, training of data is essential to avoid the data biasness. Further, the biasness in the data can also affect the accuracy of software. Therefore, SQA team must continuously assess and mitigate biases, so that accurate outcomes can be achieved.

Furthermore, the capability of GenAI for testing at different phases of software development life cycles is also important. Therefore, leveraging the full potential of GenAI for enhancing test practices can also improve the quality and reliability of software products. Therefore, organizations can leverage the full power of GenAI and enhance the testing performance. The evolution of generative AI is gaining significant attention in software testing. GenAI models can enhance the testing performance. It interprets data outcome and provide measures for quality standards. This capability of AI models can enhance the testing and SQA of software testing. It simplifies the process of testing and automation by enabling the SQA.

The productivity of testing can also increase by adopting AI tools. While traditional testing is limited to the complexity of software. SQA team can now deal with complexities and enhance the performance of testing (Chinamanagonda, 2021). This not only improves the testing capabilities but also offer flexibility and efficiency in SQA process. While GenAI has the potential to deal with complexity, it can also enhance the testing procedures. Many testing tasks at different levels can be automated. However, data biases can reduce the efficiency of testing of software. Therefore, organizations need to mitigate the data biases. By doing so, SQA team can maximize the benefits of GenAI.

For the success of AI-driven based SQA, the quality of data plays a vital role. As AI models depend on vast amounts of data, they learn patterns and interpret data. However, in SQA, data-related challenges are also on rise which affect the testing of software. These challenges significantly affect the AI-driven QA. The development of AI models can also depend on quality of data. These challenges are dynamic in nature. Therefore, in software testing, there is a need for continuous learning which can further improve the testing methods. Training an AI model is essential, and biasness can affect the data outcomes. So, organizations must reduce biasness

(Shekhar, 2022). The mitigation of data bias is important as it can affect the decision-making. Organizations can mitigate data bias by during the testing phases.

Table 2.1

**Descriptive analysis**

<b>Factors</b>	<b>Mean</b>	<b>Rank</b>
Technical Complexity	3.287	1
Skill and Knowledge Gaps	3.012	4
Financial Constraints	3.124	3
Organizational Resistance	3.156	2

Based on the Table 2.1, it is observed that technical complexity, skill and knowledge gaps, financial constraints, and organizational resistance are the major factors in implementing AI-driven QA automation in IT industries. Furthermore, the relationship between these factors were also examined by applying Pearson correlation test using SPSS software.

Table 2.2

**Pearson correlation test analysis**

<b>Factors</b>	Technical Complexity	Skill and Knowledge Gaps	Financial Constraints	Organizational Resistance
Technical Complexity	1	0.632**	0.591**	0.473**
Skill and Knowledge Gaps	0.632**	1	0.612**	0.676**
Financial Constraints	0.591**	0.612**	1	0.382**
Organizational Resistance	0.473**	0.676**	0.382**	1

Based on Table 2.2, it is observed that there is a significant (moderate to strong) relationship between these factors, p-value is less than 0.05.

### 3. BUILDING AI DRIVEN TEST AUTOMATION FRAMEWORK FOR QA

#### 3.1 Plan for the adoption of AI driven framework for QA

In this study the challenges regarding the adoption of AI driven framework for QA has been investigated. It is observed that technical complexity, skill and knowledge gaps financial constraints, and organizational resistance are the major factors in implementing AI-driven QA automation in IT industries. Fig. 3.1 shows the AI driven adoption framework for QA within the context of IT industry.

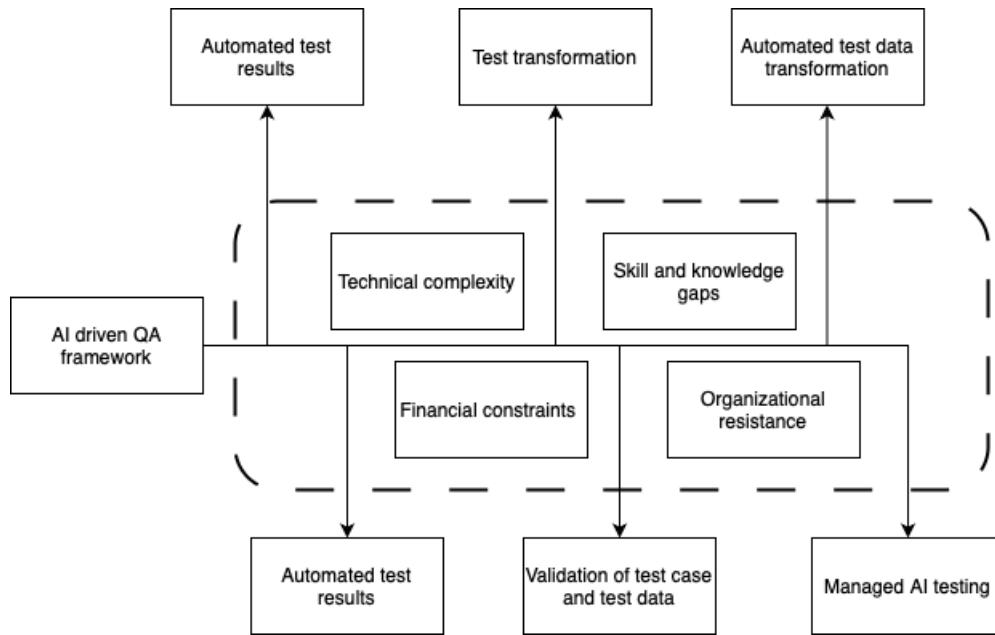


Fig. 3.1 AI driven adoption framework for QA

Fig. 3.1 shows the AI driven adoption framework for QA within the context of IT industry. The AI driven framework as shown in fig. 3.1 serve as a strategic guide for IT industry aiming to integrate AI in QA process. The proposed framework can be used by the IT industry. The proposed framework has different key outcomes such as automated test results, test automation, automated test data transformation, managed AI testing and validation of test case and test data. These outcomes collectively enhance the outcomes in QA. For instance, managed AI testing help IT industry for delivering end-to-end test model. This will support the testing activities driven by AI framework. Further, IT industry must design a transformation roadmap by identifying the weak areas. Additionally, custom made AI framework must integrate automation test suite with test execution logs which helps in the generation of automated test results.

Also, AI engine can also generate bulk test data and custom-made algorithms validate the accuracy of test cases and test data. Moreover, ML driven algorithms can also measure the test coverage, and it can identify the gaps with request to requirements. Ultimately, this framework acts as a strategic tool for IT industry which enable their transition from traditional QA to automate QA.

### **3.2 Planned costs for the adoption of AI driven framework for QA**

Generative AI models improve the testing procedure. By analyzing the customer data, businesses can achieve the desired results. For instance, generative AI solutions can create testing data. This improvement in testing can enhance the system performance by automation (Al Alamin and Uddin, 2021). Generative AI model employs ML techniques and enhance the performance of the system (Lenarduzzi et al., 2021). Generative AI will dominate the testing landscape. Therefore, businesses can leverage generative AI models and enhance the testing performance. The software testing using AI models can save time, reduce costs, and allow SQA team to focus on system performance.

The generative AI models use past data and benchmark the performance of software development (Chinamanagonda, 2021). In the future, AI models can automate the testing procedure and performance of software. Generative AI models is capable for enhancing the human creativity rather than replacing it. AI tool can help SQA team to better analyze its testing performance (Neto et al., 2022). Generative AI models can lead to software development. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. Generative AI models can expand the testing capabilities (Poth et al., 2022). Generative AI models can improve the performance of software testing by automation. Future automated software testing can be shaped by using generative AI techniques. Future automated software testing can be shaped by using generative AI techniques. The continuous development of generative AI in the context of software testing is helpful for achieving the desired results.

Data privacy and security plays an important role as it can lead to compromised results. With increasing focus on data privacy, SQA team can ensure data safety and security. Robust encryption measures are needed which can ensure data security and privacy, effectively (Mahfuz, 2016). Data security and privacy concerns are on top priority among researchers and practitioners. It's imperative to consider how customer information is handled with safety and security. Generative AI models open new challenges and possibilities in SQA testing. From encryption protocols to ethical considerations, organizations must ensure data security and privacy when

undergoing deployment. Understanding the complexities in data security can ensure the privacy of data.

Data should be encrypted using secure protocols like Transport Layer Security (TLS) or Secure Sockets Layer (SSL). This can ensure safety and privacy of data. Further, it can prevent the data from unauthorized access. Additionally, training of data sets is also important in the testing of SQA. Strict controls on data accessing are essential for managing the testing phases of software. Therefore, implementation and ensure security of data is important (Mistrík et al., 2015). Secure deployment of encryption helps in protecting the data (Naik and Tripathy, 2015). The ability of AI models to identify patterns in historical data can enable SQA team to detect bugs at early stages of software development cycles (Veeramachaneni, 2021). Development of software testing can be more advanced using Generative AI models (Barenkamp et al., 2020). AI models can automate software testing by analyzing the past historical data. This can enhance the performance of software using automation. AI models can test the complexity in the codes. This advancement can enhance the security testing by identifying the patterns (Song et al., 2022).

The productivity of testing can also increase by adopting AI tools. While traditional testing is limited to the complexity of software. SQA team can now deal with complexities and enhance the performance of testing (Chinamanagonda, 2021). This not only improves the testing capabilities but also offer flexibility and efficiency in SQA process. While GenAI has the potential to deal with complexity, it can also enhance the testing procedures. Many testing tasks at different levels can be automated. However, data biases can reduce the efficiency of testing of software. Therefore, organizations need to mitigate the data biases. By doing so, SQA team can maximize the benefits of GenAI (Singhal et al., 2021). The planned cost for AI driven framework as shown in table 3.1. It serves as a strategic guide for IT industry aiming to integrate AI in QA process. The table 3.1 has different categories of cost.

AI-powered tools can execute different test scenarios and can automate the testing procedure involved in SQA. This significantly reduces the testing time, allowing QA professionals to focus more on testing and automation (Sofian et al., 2022). This can enhance the testing procedure. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams (Farah, 2024). Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

Table 3.1

**Categories of cost**

<b>Components</b>	<b>Description</b>	<b>Cost (Euro)</b>
Licensing cost of AI tool	Licensing cost (Bharadwaj, 2025)	88485
Upgradation cost for infrastructure	Upgradation cost (DDI development, 2024)	619395
Skill and training development	Skill and training development cost (Bizzuka, 2024)	13272.75
Trained dataset	High quality trained dataset (Coherent solutions, 2025)	79636.50
Implementation cost for AI tools	One time implementation cost (Future Processing, 2024)	442425.00
Maintenance cost	Maintenance cost (Shalwa, 2024)	13271.87
	Total	1256486.12

Table 3.1 shows the planned cost for AI driven framework. The licensing cost will be Euro 88485. The upgradation cost will be Euro 619395. The skill and training development cost will be Euro 13272.75. The cost of high-quality trained dataset will be Euro 79636.50. The one-time implementation cost will be 442425.00 and the maintenance cost will be Euro 13271.87. Therefore, IT industry must make an investment of Euro 1256486.12. High-quality data can help firm to innovate the testing methods. It can also accelerate the progress of software testing. Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results (Khan et al., 2023). In essence, the positive impact of data quality on test methods can also be enhanced. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.

The deep learning models integrating with neural networks can enhance the testing performance of software (Devalla et al., 2023). This can reduce the workload of SQA teams and

improve testing performance. Further, software developers do not need to create many data rules to identify the bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively (Awad et al., 2024). Generative AI models use test data and analyze the performance of software (Martínez-Fernández et al., 2022). Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

### **3.3 Cost-benefit analysis for the adoption of AI driven framework for QA**

AI-based testing tools can directly detect the issues in the early stages of software development. AI tools can create different test cases by constructing different scenarios. This reduces manual testing process and enhance testing performance (Qazi et al., 2023). AI can analyze historical data and detect issues at early stages. It can also predict which part of code can likely to be fail. This can enhance the testing productivity. AI models can predict complexity associated with the bugs. Whenever, there will be a minor change, AI tools can automatically perform scripts (Smith, 2024). The traditional testing methods are not capable for performing the advance testing such as regression testing, code discrepancies etc. These drawbacks greatly affect the SQA performance. AI tools can perform regression testing and code discrepancies and enhance the SQA performance. By using AI tools, SQA team can better focus on the performance of software. They can detect complex issues which cannot be detected through traditional methods. Additionally, AI tools allow SQA team to focus on routine testing activities. Thus, the use of AI tools in software testing has gaining more attention, driving further innovation and quality in software development (Tao et al., 2019).

The QA process is resource-intensive because of the complexity of software. AI tools use advance technologies like neural networks, natural language processing etc. and find the bugs in the coding of software. AI tools can help to detect bugs in early development stages (Gezici and Tarhan, 2022). It can also advance the SQA process. More, specifically, AI can broaden the scope of SQA by detecting the issues and performance of the SQA process. If integrated with SQA process, it can self-generat test scenarios that cater the needs of customers. AI can make the SQA

process smooth by writing the test scripts. It can enable faster testing and without compromising performance and quality standards (Shekhar, 2022). QA is essential to achieving the performance of software. As technology advances, AI tools are now used for improving the software quality (Khankhoje, 2023). By using AI tools, SQA team can better focus on the performance of software. Table 3.2 shows the expected benefits. The total manual testing and testing without AI would cost around €132,727.50 average based on (MORS report) .

Table 3.2

<b>Expected benefit</b>	
<b>Benefits</b>	<b>Cost (Euro)</b>
Testing time reduction cost (35%) SDET Technologies (Report)	46454.625
Manual efforts are reduced (40%)	53091.00
Faster bug detection (20%) (Qazi et al. 2023)	26545.50
Total	126091.125

Table 3.2 shows the expected benefits. High-quality data can help firm to innovate the testing methods. It can also accelerate the progress of software testing by reducing the testing time (35%). Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results (Khan et al., 2023) by minimizing manual efforts (40%). In essence, the positive impact of data quality on test methods can also be enhanced. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.

The deep learning models integrating with neural networks can enhance the testing performance of software (Qazi et al. 2023) by faster bug detection (20%). This can reduce the workload of SQA teams and improve testing performance. Further, software developers do not need to create many data rules to identify the bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively (Awad et al., 2024). Generative AI models use test data and analyze the performance of software (Martínez-Fernández et al., 2022). Based on the requirements, Generative AI models generate test data which can be analyzed by SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from

patterns, AI models can predict the behavior of system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically.

$$\text{Net benefit} = 1256486.12 - 126091.125 = \text{Euro } 1130394.995$$

$$\text{Return on investment} = \frac{126091.125}{1256486.12} \times 100 = 10.035 = 10 \text{ years}$$

Therefore, by implementing AI tools in SQA process IT industry can achieve the benefits of Euro 1130394.995 with a payback period of 10 years.

## 4. GENERAL CONCLUSIONS

### Conclusion

1. In conclusion, technical complexity, skill and knowledge gaps, financial constraints, and organizational resistance are the major factors in implementing AI-driven QA automation in IT industries.
2. The proposed framework can be used by the IT industry. The proposed framework has different key outcomes such as automated test results, test automation, automated test data transformation, managed AI testing and validation of test case and test data.
3. These outcomes collectively enhance the outcomes in QA. For instance, managed AI testing help IT industry for delivering end-to-end test model. This will support the testing activities driven by AI framework.
4. Further, IT industry must design a transformation roadmap by identifying the weak areas. Additionally, custom made AI framework must integrate automation test suite with test execution logs which helps in the generation of automated test results.
5. Also, AI engine can also generate bulk test data and custom-made algorithms validate the accuracy of test cases and test data.
6. Moreover, ML driven algorithms can also measure the test coverage, and it can identify the gaps with request to requirements. Ultimately, this framework acts as a strategic tool for IT industry which enable their transition from traditional QA to automate QA.
7. Today, Generative AI models are gaining attention among practitioners. It has a wide range of applications in various sectors. Generative AI models can handle vast and diverse collections of unorganized data and can offer testing automation.
8. The licensing cost will be Euro 88485. The upgradation cost will be Euro 619395. The skill and training development cost will be Euro 13272.75. The cost of high-quality trained dataset will be Euro 79636.50. The one-time implementation cost will be 442425.00 and the maintenance cost will be Euro 13271.87.

### Recommendations

1. IT industry must make an investment of Euro 1256486.12. High-quality data can help firms to innovate the testing methods. It can also accelerate the progress of software testing. Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards. It also forms a strong foundation for providing reliable results. In essence, the positive impact of data quality on test methods can also be enhanced.
2. High-quality data can help firms to innovate the testing methods. It can also accelerate the progress of software testing by reducing the testing time (35%). Therefore, high data quality is needed to ensure data-driven decision-making. This can empower SQA team to explore new ideas and opportunities with certainty which can ensure quality standards.

3. It also forms a strong foundation for providing reliable results by minimizing manual efforts (40%). In essence, the positive impact of data quality on test methods can also be enhanced.
4. The deep learning models integrating with neural networks can enhance the testing performance of software by faster bug detection (20%). This can reduce the workload of SQA teams and improve testing performance.
5. Further, software developers do not need to create many data rules to identify bugs because it can automatically learn data patterns through trained datasets. Hence, Generative AI models are a promising approach for the testing of software effectively.
6. Based on the requirements, Generative AI models generate test data which can be analyzed SQA teams. By analyzing the test data using generative AI models, firm can enhance the performance of SQA teams.
7. Generative AI models can study software code or system behaviors based on gathered data over a time. By learning from patterns, AI models can predict the behavior of the system. Generative AI models are well suited for software testing, because it has the capacity to create test scripts and test code automatically. Therefore, by implementing AI tools in SQA process IT industry can achieve the benefits of Euro 1130394.99 with a payback period of 10 years.

## Literature and sources used

1. Atoum, I., Baklizi, M. K., Alsmadi, I., Otoom, A. A., Alhersh, T., Ababneh, J., ... & Alshahrani, S. M. (2021). Challenges of software requirements quality assurance and validation: A systematic literature review. *IEEE Access*, 9, 137613-137634.
2. Pargaonkar, S. (2023). Synergizing requirements engineering and quality assurance: A comprehensive exploration in software quality engineering. *International Journal of Science and Research*, 12(8), 2003-2007.
3. Shalwa, 2024. How Much Does AI Software Cost? Available at: <https://artsmart.ai/blog/how-much-does-ai-software-cost/>
4. Coherent solutions, 2025. How Much Does It Cost to Develop an AI Solution? Pricing and ROI Explained. Available at: <https://www.coherentsolutions.com/insights/ai-development-cost-estimation-pricing-structure-roi>
5. Future Processing, 2024. AI pricing: how much does Artificial Intelligence cost? Available at: <https://www.future-processing.com/blog/ai-pricing-is-ai-expensive/>
6. Felderer, M., & Ramler, R. (2021). Quality assurance for AI-based systems: Overview and challenges (introduction to interactive session). In *Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference, SWQD 2021, Vienna, Austria, January 19–21, 2021, Proceedings 13* (pp. 33-42). Springer International Publishing.
7. Test Institute, 2024. What is software quality assurance? Available at: [https://www.test-institute.org/What\\_is\\_Software\\_Quality\\_Assurance.php](https://www.test-institute.org/What_is_Software_Quality_Assurance.php)
8. Bizzuka, 2024. How Much Does AI Training for Businesses Cost? Available at: <https://www.bizzuka.com/how-much-does-ai-training-for-businesses-cost/>
9. DDI development, 2024. How Much Does AI Cost in 2025. Available at: <https://ddi-dev.com/blog/programming/how-much-does-ai-cost>
10. Bharadwaj, C. (2025). How much does it cost to develop an AI testing tool. Available at: <https://appinventiv.com/blog/ai-testing-tool-development-cost/>
11. Tian, J. (2005). *Software quality engineering: testing, quality assurance, and quantifiable improvement*. John Wiley & Sons.
12. Mnkandla, E., & Dwolatzky, B. (2006). Defining agile software quality assurance. In *2006 International Conference on Software Engineering Advances (ICSEA'06)* (pp. 36-36). IEEE.

13. Naik, K., & Tripathy, P. (2011). *Software testing and quality assurance: theory and practice*. John Wiley & Sons.
14. Mahfuz, A. S. (2016). *Software Quality Assurance: Integrating Testing, Security, and Audit*. CRC Press.
15. Maxim, B. R., & Kessentini, M. (2016). An introduction to modern software quality assurance. In *Software quality assurance* (pp. 19-46). Morgan Kaufmann.
16. Khanjani, A., & Sulaiman, R. (2011). The process of quality assurance under open source software development. In *2011 IEEE Symposium on Computers & Informatics* (pp. 548-552). IEEE.
17. Kassie, N. B., & Singh, J. (2020). A study on software quality factors and metrics to enhance software quality assurance. *International Journal of Productivity and Quality Management*, 29(1), 24-44.
18. Gao, J., Tsao, H. S., & Wu, Y. (2003). *Testing and quality assurance for component-based software*. Artech House.
19. Bajnaid, N. O. (2013). *An ontological approach to model software quality assurance knowledge domain* (Doctoral dissertation, London Metropolitan University).
20. Mistrík, I., Soley, R. M., Ali, N., Grundy, J., & Tekinerdogan, B. (Eds.). (2015). *Software quality assurance: in large scale and complex software-intensive systems*. Morgan Kaufmann.
21. Ramchand, S., Shaikh, S., & Alam, I. (2021, August). Role of artificial intelligence in software quality assurance. In *Proceedings of SAI Intelligent Systems Conference* (pp. 125-136). Cham: Springer International Publishing.
22. Ouaraous, R., Hilal, I., & Mezrioui, A. (2023). On Using Artificial Intelligence in Software Quality Assurance: A State of the Art.
23. Felderer, M., & Ramler, R. (2021). Quality assurance for AI-based systems: Overview and challenges (introduction to interactive session). In *Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference, SWQD 2021, Vienna, Austria, January 19–21, 2021, Proceedings 13* (pp. 33-42). Springer International Publishing.
24. Farah, J. (2024). AI-Driven Software Testing: Automating Quality Assurance with Machine Learning for Distributed Networks.

25. Natarajan, D. R. (2020). AI-Generated Test Automation for Autonomous Software Verification: Enhancing Quality Assurance Through AI-Driven Testing. *International Journal of HRM and Organizational Behavior*, 8(4), 89-103.
26. Trifunova, A., Jakimovski, B., Chorbev, I., & Lameski, P. (2024, November). AI in Software Testing: Revolutionizing Quality Assurance. In *2024 32nd Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE.
27. Goericke, S. (2020). *The future of software quality assurance*(p. 257). Springer Nature.
28. Fujii, G., Hamada, K., Ishikawa, F., Masuda, S., Matsuya, M., Myojin, T., ... & Ujita, Y. (2020). Guidelines for quality assurance of machine learning-based artificial intelligence. *International journal of software engineering and knowledge engineering*, 30(11n12), 1589-1606.
29. Qazi, S., Memon, I., Memon, S., Hashmi, B. Q., Memon, M. S., & Ali, Q. E. (2023). Software Quality Assurance Using Artificial Intelligence Techniques: A Survey of the Software Industry of Pakistan. *Journal of Hunan University Natural Sciences*, 50(12).
30. Smith, H. K. (2024). Leveraging AI for Proactive Software Quality Assurance: Predictive Analytics, Machine Learning, and Agile Test Automation.
31. Tao, C., Gao, J., & Wang, T. (2019). Testing and quality validation for ai software–perspectives, issues, and practices. *IEEE Access*, 7, 120164-120175.
32. Gezici, B., & Tarhan, A. K. (2022). Systematic literature review on software quality for AI-based software. *Empirical Software Engineering*, 27(3), 66.
33. Fujii, G., Hamada, K., Ishikawa, F., Masuda, S., Matsuya, M., Myojin, T., ... & Ujita, Y. (2020). Guidelines for quality assurance of machine learning-based artificial intelligence. *International journal of software engineering and knowledge engineering*, 30(11n12), 1589-1606.
34. Shekhar, P. C. (2022). Accelerating Agile Quality Assurance with AI-Powered Testing Strategies.
35. Nama, P. (2024). Integrating AI in testing automation: Enhancing test coverage and predictive analysis for improved software quality. *World Journal of Advanced Engineering Technology and Sciences*, 13, 769-782.
36. Khankhoje, R. (2023). Quality Challenges and Imperatives in Smart AI Software. *Computer Science & Information Technology (CS & IT)*, 143-152.

37. Martínez-Fernández, S., Bogner, J., Franch, X., Oriol, M., Siebert, J., Trendowicz, A., ... & Wagner, S. (2022). Software engineering for AI-based systems: a survey. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(2), 1-59.
38. Awad, A., Qutqut, M. H., Ahmed, A., Al-Haj, F., & Almasalha, F. (2024, December). Artificial Intelligence Role in Software Automation Testing. In *2024 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 1-6). IEEE.
39. Khan, S. A., Oshin, N. T., Nizam, M., Ahmed, I., Musfique, M. M., & Hasan, M. (2023, September). AI-Based Software Testing. In *World Conference on Information Systems for Business Management* (pp. 323-334). Singapore: Springer Nature Singapore.
40. Wang, Q., Wang, J., Li, M., Wang, Y., & Liu, Z. (2024). A Roadmap for Software Testing in Open-Collaborative and AI-Powered Era. *ACM Transactions on Software Engineering and Methodology*.
41. Gezici, B., & Tarhan, A. K. (2022). Systematic literature review on software quality for AI-based software. *Empirical Software Engineering*, 27(3), 66.
42. Natarajan, D. R. (2020). AI-Generated Test Automation for Autonomous Software Verification: Enhancing Quality Assurance Through AI-Driven Testing. *International Journal of HRM and Organizational Behavior*, 8(4), 89-103.
43. Lenarduzzi, V., Lomio, F., Moreschini, S., Taibi, D., & Tamburri, D. A. (2021). Software quality for ai: Where we are now?. In *Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference, SWQD 2021, Vienna, Austria, January 19–21, 2021, Proceedings 13*(pp. 43-53). Springer International Publishing.
44. Ali, M. A., Yap, N. K., Ghani, A. A. A., Zulzalil, H., Admodisastro, N. I., & Najafabadi, A. A. (2022). A systematic mapping of quality models for AI systems, software and components. *Applied Sciences*, 12(17), 8700.
45. Singhal, P., Kundu, S., Gupta, H., & Jain, H. (2021, December). Application of artificial intelligence in software testing. In *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 489-492). IEEE.
46. Pham, P., Nguyen, V., & Nguyen, T. (2022, October). A review of ai-augmented end-to-end test automation tools. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering* (pp. 1-4).
47. Job, M. A. (2021). Automating and optimizing software testing using artificial intelligence techniques. *International Journal of Advanced Computer Science and Applications*, 12(5).

48. Krichen, M. (2022). How artificial intelligence can revolutionize software testing techniques. In *International Conference on Innovations in Bio-Inspired Computing and Applications* (pp. 189-198). Cham: Springer Nature Switzerland.
49. Khaliq, Z., Farooq, S. U., & Khan, D. A. (2022). Artificial intelligence in software testing: Impact, problems, challenges and prospect. *arXiv preprint arXiv:2201.05371*.
50. Al Alamin, M. A., & Uddin, G. (2021, August). Quality assurance challenges for machine learning software applications during software development life cycle phases. In *2021 IEEE International Conference on Autonomous Systems (ICAS)* (pp. 1-5). IEEE.
51. Chinamanagonda, S. (2021). AI-driven Performance Testing AI tools enhancing the accuracy and efficiency of performance testing. *Advances in Computer Sciences*, 4(1).
52. Neto, A. V. S., Camargo, J. B., Almeida, J. R., & Cugnasca, P. S. (2022). Safety assurance of artificial intelligence-based systems: A systematic literature review on the state of the art and guidelines for future work. *IEEE Access*, 10, 130733-130770.
53. Poth, A., Meyer, B., Schlicht, P., & Riel, A. (2020, December). Quality Assurance for Machine Learning—an approach to function and system safeguarding. In *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)* (pp. 22-29). IEEE.
54. Veeramachaneni, V. (2021). AI-driven software development: enhancing code quality and maintainability through automated refactoring. *Int J Recent Dev Sci Technol*, 5(6), 94-101.
55. Chinamanagonda, S. (2021). AI-driven Performance Testing AI tools enhancing the accuracy and efficiency of performance testing. *Advances in Computer Sciences*, 4(1).
56. Nigam, A., Pasricha, R., Singh, T., & Churi, P. (2021). A systematic review on AI-based proctoring systems: Past, present and future. *Education and Information Technologies*, 26(5), 6421-6445.
57. Osamy, W., Khedr, A. M., Salim, A., El-Sawy, A. A., Alreshoodi, M., & Alsukayti, I. (2022). Recent advances and future prospects of using AI solutions for security, fault tolerance, and QoS challenges in WSNs. *Electronics*, 11(24), 4122.
58. Barenkamp, M., Rebstadt, J., & Thomas, O. (2020). Applications of AI in classical software engineering. *AI Perspectives*, 2(1), 1.
59. Sofian, H., Yunus, N. A. M., & Ahmad, R. (2022). Systematic mapping: Artificial intelligence techniques in software engineering. *IEEE Access*, 10, 51021-51040.

60. Nguyen, P. H., Sen, S., Jourdan, N., Cassoli, B., Myrseth, P., Armendia, M., & Myklebust, O. (2022). Software engineering and AI for data quality in cyber-physical systems-sea4dq'21 workshop report. *ACM SIGSOFT Software Engineering Notes*, 47(1), 26-29.
61. Tadi, S. R. C. C. T. (2021). The Future of Automated Software Engineering: Enhancing Full-Stack Development with Predictive Modeling. *European Journal of Advances in Engineering and Technology*, 8(6), 147-158.
62. Santhanam, P. (2020). Quality management of machine learning systems. In *International Workshop on Engineering Dependable and Secure Machine Learning Systems* (pp. 1-13). Cham: Springer International Publishing.
63. Song, J., Lyu, D., Zhang, Z., Wang, Z., Zhang, T., & Ma, L. (2022, May). When cyber-physical systems meet AI: a benchmark, an evaluation, and a way forward. In *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice* (pp. 343-352).
64. Ericsson, M., Johansson, D., & Stjern, D. (2021). AI-based quality control of wood surfaces with autonomous material handling. *Applied Sciences*, 11(21), 9965.
65. Dandotiya, S. (2025). Generative AI for software testing: Harnessing large language models for automated and intelligent quality assurance. *International Journal of Science and Research Archive*, 14(1), 1931-1935.
66. Gao, J., Tao, C., Jie, D., & Lu, S. (2019). What is AI software testing? and why. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*(pp. 27-2709). IEEE.
67. Aleti, A. (2023, May). Software testing of generative ai systems: Challenges and opportunities. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)* (pp. 4-14). IEEE.
68. Sugali, K. (2021). Software testing: Issues and challenges of artificial intelligence & machine learning.
69. Shekhar, P. C. (2022). Accelerating Agile Quality Assurance with AI-Powered Testing Strategies.
70. Khan, S. A., Oshin, N. T., Nizam, M., Ahmed, I., Musfique, M. M., & Hasan, M. (2023, September). AI-Based Software Testing. In *World Conference on Information Systems for Business Management* (pp. 323-334). Singapore: Springer Nature Singapore.
71. Devalla, S., & Yogi, M. K. (2023). Building trust in ai-a simplified guide to ensure software quality. *Journal of Soft Computing Paradigm*, 5(3), 218-231.

72. Awad, A., Qutut, M. H., Ahmed, A., Al-Haj, F., & Almasalha, F. (2024). Artificial Intelligence Role in Software Automation Testing. In *2024 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 1-6). IEEE.
73. Global AI market (2024). Software Testing Market Report. Available at: <https://market.us/report/ai-in-software-testing-market/>
74. WisdomSchema (2024). Data Quality and Impact on AI. Available at: <https://wisdomschema.com/blog/data-quality-and-impact-on-ai/>
75. Arramton Infotech (2024). What is the role of AI data quality in achieving AI business objectives? Available at: <https://arramton.com/blogs/role-of-ai-data-quality>
76. Li, Z., Sharma, V., & Mohanty, S. P. (2020). Preserving data privacy via federated learning: Challenges and solutions. *IEEE Consumer Electronics Magazine*, 9(3), 8-16.
77. Ksolves (2024). Data Privacy Challenges and Best Practices in the Era of Generative AI. Available at: <https://www.ksolves.com/blog/artificial-intelligence/data-privacy-challenges-and-best-practices-in-the-era-of-generative-ai>
78. (Sify, 2024). Forecasting the Future: Generative AI in Software Testing for 2024. Available at: <https://www.sify.com/ai-analytics/forecasting-the-future-generative-ai-in-software-testing-for-2024/>.
79. SDET-Technologies are a group of Software Quality Evangelists(2025) Available at: <https://sdettech.com/how-ai-cut-testing-time-by-30-35-a-case-study/>
80. "AI in the Testing Industry Statistics," published on April 29, 2025 Available at: <https://gitnux.org/ai-in-the-testing-industry-statistics/>
81. MORS case study reports (2025) Available at: <https://morsoftware.com/blog/software-testing-costs>

## Appendix

### Questionnaire

Gender

Male       Female

Age

25-30       30-35       More than 35

Job position

Junior level       Middle level       Senior level

Job experience

0-5       5-10       More than 10

For testing, my company has implemented QA automation

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

Our current project needs the implementation of QA automation tools

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

Our team is familiar with AI-driven testing tools and frameworks

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

The efficiency of our software testing has been improved by AI-based testing

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

Our company has integrated both manual and automated testing effectively

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

We are actively exploring or adopting AI solutions in our QA strategy.

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

AI-driven testing has helped us reduce bugs and defects in production.

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

Our organization regularly evaluates new AI tools for potential integration in QA automation.

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

The return on investment (ROI) from QA automation is satisfactory.

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

AI-driven testing aligns well with our agile or DevOps development methodologies.

Strongly disagree    Disagree    Neutral    Agree    Strongly agree

### **Technical Complexity**

Implementing AI-based testing tools requires advanced technical knowledge.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Our existing systems are not compatible with AI testing tools.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Lack of proper documentation makes AI tool integration difficult.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

The setup and maintenance of AI-driven QA solutions are complex.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

### **Skill and Knowledge Gaps**

Our QA team lacks the necessary skills to implement AI automation.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

There is limited training available on AI-driven QA technologies.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Hiring experts in AI testing is a major challenge.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Team members are hesitant to learn or adopt AI testing practices.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

### **Financial Constraints**

The cost of implementing AI in QA is too high.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Licensing AI testing tools exceeds our current budget.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

ROI from AI-driven QA is unclear, making investment risky.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Financial decision-makers are reluctant to allocate funds to AI QA tools.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

### **Organizational Resistance**

There is resistance to change from traditional QA practices.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Management lacks confidence in AI-driven QA solutions.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Team members fear that AI automation might replace human roles.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

Organizational culture does not support AI innovation in QA.

Strongly disagree  Disagree  Neutral  Agree  Strongly agree

THE DATE THE DOCUMENT WAS SIGNED IS THE LAST ADDED SECURE ELECTRONIC  
DATE OF SIGNATURE AND ITS TIME STAMP

Master thesis “Automation in QA: A Project Management Framework for Delivering AI-Driven Testing Solutions” elaborated in the Faculty of Economics and Social Science of the University of Latvia.

Herewith I confirm by putting my e-signature that the research has been conducted individually, as well as the fact that only the indicated sources of literature have been used in the paper and that the electronic copy of the paper corresponds to the printout.

Supervisor: prof. Andrejs Cekuls

Reviewer: pasn. Ilze Romane

The Thesis was uploaded to e-studies with an e-signature  
Checked by the Dean's authorized person:

Study Information Center methodologist Inese Ranka