



Vaasan yliopisto
UNIVERSITY OF VAASA

Shalini Veronika

**Optimizing Software Development and Quality in
the Finnish Software Industry: Agile Practices and
Quality Management**

School of Technology and Innovations
Master's thesis in Industrial Management
Master of Science in Economics and Business Administration

Vaasa 2025

UNIVERSITY OF VAASA**School of Technology and Innovations**

Author:	Shalini Veronika	
Title of the thesis:	Optimizing Software Development and Quality in the Finnish Software Industry: Agile Practices and Quality Management	
Degree:	Master of Science in Economics and Business Administration	
Discipline:	Industrial Management	
Supervisor:	Ville Tuomi	
Year:	2025	Pages: 112

ABSTRACT:

The Finnish software industry is a dynamic and technologically advanced sector, comprising a diverse mix of startups and multinational firms engaged in software development, consulting, and IT services. Agile methodologies have become foundational to this ecosystem, widely adopted to enhance development speed, adaptability, and customer satisfaction. However, optimizing Agile implementation to consistently improve software quality remains a central challenge for practitioners and researchers alike.

This thesis investigates the impact of Agile practices on software quality and customer satisfaction within the Finnish software sector. A conceptual framework was developed around five core dimensions: Agile testing practices, adherence to Agile principles, iteration frequency, test automation maturity, and customer feedback integration. Employing a quantitative, cross-sectional research design, the study utilized a structured survey instrument targeting software developers, QA engineers, project managers, and Agile coaches. A total of 200 valid responses were collected and statistically analyzed using Pearson correlation and multiple linear regression to evaluate the relationships between Agile practices and key software outcomes.

The thesis is organized into five chapters. Chapter one establishes the research background and objectives. Chapter two reviews key theoretical foundations in Agile methodology, software quality engineering, and customer-centric development. Chapter three details the methodological approach, including sampling, instrument design, and validation techniques. Chapter four presents the empirical results, while chapter five discusses conclusions, practical implications, limitations, and future research directions.

Findings demonstrate that Agile testing practices and high iteration frequency exert the strongest positive influence on software quality and customer satisfaction. Automation practices and structured feedback loops also contribute meaningfully, though their impact may vary across organizational contexts. The reliability of the instrument, tested through Cronbach's alpha ($\alpha = 0.700$), supports the internal consistency of key measurement scales, while statistical diagnostics confirm the robustness of the regression model.

Despite limitations such as potential self-reporting bias and the regional focus, this study provides actionable insights for software teams seeking to align Agile practices with quality assurance goals. It highlights the strategic importance of continuous integration, iterative development, and collaborative feedback mechanisms in optimizing software performance and stakeholder satisfaction.

KEYWORDS: Agile practices, software quality, Agile testing, customer satisfaction, continuous integration, test automation, TDD, BDD, regression analysis, quantitative research

Contents

1 Introduction	9
1.1 Background and the aim of the study	9
1.2 Research Question and Objectives	10
1.4 Formulation of Research Hypotheses	10
1.4 Scope of the Study	11
2 Literature Review	13
2.1 Key Concepts and Overview of Agile Methodologies	14
2.2 Software Quality Engineering	17
2.3 Quality Management and Agile Software Practices	18
2.4 Core Principles and Practices of Agile Methodologies	22
2.5 Integration of Quality Management Practices within Agile Environments	23
2.6 Analysis of Development Strategies that Support Agile Practices	24
Continuous Integration and Continuous Delivery (CI/CD)	25
2.7 Cross-Functional Teams	26
2.8 Case Studies and Empirical Research on Effective Development Strategies in Agile Projects	26
2.9. Conceptual Framework Justification for Combining Dependent Variables	27
Justification for Combining Dependent Variables	28
Conceptual Framework Diagram	29
3 Methodology	30
3.1 Population and sample size	30
3.3 Quantitative Data Analysis	39
3.4 Study Plan and Justification	42
4 Results	43
4.2 Customer Collaboration in Agile	56

4.3 Test Automation Coverage	69
4.4 Feedback Loops and Satisfaction Analysis	82
4.5 Quantitative Analysis: Testing Hypotheses and Exploring Relationships in Agile Practices and Software Quality Outcomes	84
5 Conclusion	91
5.1 Summary of Findings	91
5.2 Evaluation of Hypotheses	92
5.3 Limitations and Future Research	93
5.4 Implications for Practice and conclusive thoughts	95
References	96
Appendices	105
Appendix 1. Structured questionnaire questions	105

Figures

Figure 1. Conceptual Framework designed by authour	29
Figure 2. Distribution of Responses on Job Role Distribution of Respondents	44
Figure 3. Distribution of Responses on Organization Size Distribution	46
Figure 4. Distribution of Responses on Duration of Agile Methodology Usage	47
Figure 5. Distribution of Responses on Primary Agile Framework Used	49
Figure 6. Distribution of Responses on Test-Driven Development (TDD) Usage	50
Figure 7. Distribution of Responses on Continuous Integration (CI) adoption	52
Figure 8. Distribution of Adoption levels Behavior-Driven Development participants	53
Figure 9. Distribution of the extent of automation in Agile practices	55
Figure 10. Distribution of the frequency of Agile practice reviews	56
Figure 11. Distribution of the Exploring the principle of customer collaboration	57
Figure 12. Distribution of the adoption of iterative feedback principles	58
Figure 13. Distribution of the team self-organization principles	60
Figure 14. Distribution of adoption of team self-organization principles	61
Figure 16. Distribution of frequency of biweekly iterations	63
Figure 17. Distribution of adjustment within Agile iterations	65
Figure 18. Distribution of planning meetings in Agile Iterations	66
Figure 19. Distribution of continuous improvement retrospectives	67
Figure 20. Distribution of the effectiveness of agile iterations in defect identification	68
Figure 21. Distribution of continuous automation test coverage	70
Figure 22. Distribution of automation CI integration	71
Figure 23. Distribution of automation effectiveness of CI integration	73

Figure 24. Distribution of impact of automation in reducing manual effort in Agile settings	74
Figure 25. Distribution of impact of responses on the effectiveness of automation	76
Figure 26. Distribution of the integration of customer feedback development process	78
Figure 27. Distribution of the Feature prioritization based on customer prioritize features	79
Figure 28. Distribution responses regarding the effectiveness of customer communication	81
Figure 29. Distribution of the established effective feedback mechanisms	82
Figure 30. Distribution of the analysis of customer satisfaction	83

Tables

Table 1. Mapping Research Objectives, Questions and Hypotheses	35
Table 2. Survey Results on Job Role Distribution of Respondents	44
Table 3. Survey Results on Organization Size Distribution	45
Table 4. Survey Results on Duration of Agile Methodology Usage	47
Table 5. Survey Results on Primary Agile Framework Used	48
Table 6. Survey Results on Test-Driven Development (TDD) Usage	50
Table 7. Survey Results on Continuous Integration (CI) adoption	51
Table 8. Survey Results on Adoption levels of Behavior-Driven Development among participants	53
Table 9. Survey Results on the extent of automation in Agile practices	54
Table 10. Survey Results on the frequency of Agile practice reviews	56
Table 11. Survey Results on the Exploring the principle of customer collaboration	57
Table 12. Survey Results on the adoption of iterative feedback principles	58
Table 13. Survey Results on data on the adoption of team self-organization principles	59
Table 14. Survey Results on data on the adoption of team self-organization principles	61
Table 15. Survey Results on Continuous improvement	62
Table 16. Survey Results on frequency of biweekly iterations	63
Table 17. Survey Results on scope adjustment within Agile iterations	64
Table 18. Survey Results on planning meetings in Agile Iterations	66
Table 19. Survey Results on continuous improvement retrospectives	67
Table 20. Survey Results on the effectiveness of agile iterations in defect identification	68
Table 22. Survey results automation CI integration	71
Table 23. Survey results in automation effectiveness of CI integration	72
Table 24. Survey results impact of automation in reducing manual effort in Agile settings	74

Table 25. Survey results impact of responses on the effectiveness monitoring automation	76
Table 26. Survey results of the integration of customer feedback development process	77
Table 27. Survey results of the Feature prioritization based on customer input and prioritize features	79
Table 29. Survey results of the established effective feedback mechanisms	82
Table 30. Survey results of the analysis of customer satisfaction	83
Table 31. reliability of the survey instrument through the use of Cronbach's alpha	85
Table 32. Residuals Statistics	86
Table 33. Residuals Statistics Coefficients	87
Table 34. Residuals Statistics Collinearity Diagnostics	88

Introduction

This research paper presents a comprehensive quantitative study on optimizing Agile testing practices and development strategies to enhance software quality and customer satisfaction in the Finnish software industry. The study examines the impact of various Agile practices on software quality outcomes and provides data-driven recommendations for improving these practices. This chapter covers the background of the study, research problem, scope and limitations, significance of the study, research objectives, and hypotheses.

1.1 Background and the aim of the study

As the software industry continues to evolve, Agile practices have gained prominence due to their flexibility and iterative approach, which are particularly suitable for environments characterized by rapid change and high complexity (Highsmith, 2009, p. 27). This research aims to provide actionable insights into how these agile practices can be optimized in the context of Finland's software industry. Given the quantitative nature of this research, the methodology is designed to collect and analyze data systematically, ensuring the validity and reliability of the findings (Kettunen et al., 2019, p. 30).

The quantitative approach is well-suited for this study because it allows for the testing of hypotheses and the identification of statistically significant relationships between key variables (Creswell, 2014, p. 155). The methodology encompasses a conceptual framework that outlines the relationships between independent and dependent variables (Kettunen et al., 2019), the formulation of specific research hypotheses, and the design of a survey instrument to gather data from a representative sample of IT professionals in Finland (Edmonds & Kennedy, 2017, p. 45).

1.2 Research Question and Objectives

The aim of this thesis work is presented in the following research question:

How can Agile testing practices and development strategies be optimized to enhance software quality engineering in the Finnish software industry?

To address this central research question, the study pursues three specific objectives.

The objectives of this research are the following:

1. To empirically examine Agile software testing practices and development strategies prevalent in the Finnish software industry.
2. To identify key factors influencing software quality engineering within Agile environments.
3. To provide recommendations for optimizing Agile testing practices and development strategies to enhance software quality in the Finnish software industry based on empirical findings.

1.4 Formulation of Research Hypotheses

The research hypotheses are formulated based on the conceptual framework and research objectives. Each hypothesis has a corresponding null hypothesis (H₀), which will be tested using the data collected (Yang et al., 2023, p. 12).

A set of research hypotheses has been formulated based on the conceptual framework and research objectives to guide the empirical analysis. These hypotheses reflect the expected relationships between key Agile practices such as Agile testing, iteration frequency, test automation, adherence to Agile principles, and customer feedback integration and the outcomes of software quality and customer satisfaction. Each hypothesis is designed to be tested using quantitative methods. A detailed

presentation of these hypotheses, including their corresponding null formulations and statistical testing approach, is provided in the methodological section of this thesis. These hypotheses are designed to test the relationships between Agile practices and the outcomes of software quality and customer satisfaction, aligning with the research objectives. These hypotheses are designed to test the relationships between Agile practices and the outcomes of software quality and customer satisfaction, aligning with the research objectives.

1.4 Scope of the Study

This study is centered on optimizing Agile testing practices and development strategies to enhance software quality and customer satisfaction within the Finnish software industry. The research primarily aims to empirically assess the current Agile practices, identify key factors influencing software quality engineering, and provide data-driven recommendations based on empirical findings (Khan et al., 2022, p. 331). The study targets IT professionals from a variety of companies within Finland, including small startups and large enterprises. These professionals, such as software developers, quality assurance engineers, and project managers, are directly involved in Agile methodologies.

The research specifically addresses key components of Agile practices, including the adoption of Agile testing techniques, adherence to Agile principles, frequency of Agile iterations, test automation, and customer feedback integration, evaluating their combined effect on software quality and customer satisfaction (Khan et al., 2022, p. 330).

1.5 Limitations of the Study

The study is confined to the Finnish software industry, which may limit the generalizability of the findings to other countries or regions with different software development cultures, industry norms, and levels of Agile adoption (Eriksson & Westerlund, 2019, p. 330). This geographic focus restricts the applicability of the results to environments outside Finland.

Although the study aims for a robust sample size, the actual response rate may be lower than anticipated, potentially affecting the statistical power and reliability of the findings (Saunders, Lewis, & Thornhill, 2016, pp. 25). Furthermore, reliance on voluntary participation could introduce self-selection bias, as those with strong opinions or significant experiences in Agile practices may be more likely to respond, skewing the data (Robson, 2023, p. 75).

The study employs a cross-sectional research design, which gathers data at a single point in time. This approach limits the ability to observe changes and trends in Agile practices and their effects on software quality and customer satisfaction over time (Creswell & Creswell, 2017, pp.12). A longitudinal approach would offer deeper insights into how these practices evolve and their long-term impacts.

Data collection is based on self-reported questionnaires, which can introduce biases such as social desirability bias or recall bias, potentially affecting the validity of the results (Podsakoff et al., 2003, p. 882). Participants may provide responses that they believe are socially acceptable or fail to accurately remember their experiences, which could lead to skewed findings (Meckenstock, 2024, p. 15).

Agile practices are implemented differently across organizations, and the study may not fully capture the nuances and variations in these practices (Dingsøy, Nerur, Balijepally, & Moe, 2012, p. 12). This could limit the generalizability of the findings to other contexts where Agile is practiced differently.

These limitations are important considerations when interpreting the results of the study (Kettunen et al., 2019, p. 75). The findings and their implications will be discussed with these constraints in mind, and recommendations for future research will be provided to address these limitations, building on the insights gained from this study. Despite these limitations, this research is expected to make a significant contribution to understanding how to optimize Agile practices for improved software quality and customer satisfaction within the Finnish software industry (Blom et al., 2017, p. 12).

2 Literature Review

In recent years, the software industry has increasingly adopted Agile methodologies to enhance the efficiency, flexibility, and overall quality of software development processes (Glowbl, 2023, p.4). Agile practices, known for their iterative and incremental approaches, have significantly transformed traditional software engineering by emphasizing continuous improvement, collaboration, and customer satisfaction (Kent et al., 2001, p.3). The adoption of Agile methods in the European software industry, characterized by its diverse market and regulatory environments, presents unique challenges and opportunities for quality management and development strategies (Dingsøy et al., 2012, p. 1213).

Agile methodologies have revolutionized the software development landscape by promoting flexibility, iterative progress, and close collaboration between cross-functional teams. Originating from the principles laid out in the Agile Manifesto (Beck et al., 2001, p.2), these practices have been adopted globally, including in the diverse and dynamic European software industry. The emphasis on rapid delivery, customer feedback, and continuous improvement has rendered Agile practices particularly effective in addressing the challenges posed by fast-paced technological advancements and evolving market demands (Dingsøy et al., 2012, pp. 1215–1216).

Quality engineering, which encompasses a range of activities aimed at ensuring software products meet specified requirements and standards, is integral to the success of Agile methodologies (Al-Saqqa, Sawalha, & AbdelNabi, 2020, p. 99). Prior work has emphasized the importance of scalable digital infrastructure, such as cloud computing, in enhancing system responsiveness and operational efficiency an aspect closely aligned with Agile implementation in high-demand environments (Veronika & Jayathilaka, 2020). Agile testing practices such as Test-Driven Development (TDD), Continuous Integration (CI), and Automated Testing are pivotal in maintaining high-quality standards

while accommodating frequent changes and incremental development (Hafiz & Johnson, 2008, p. 1570).

However, the implementation of Agile practices within the European context—characterized by unique regulatory frameworks and market conditions—requires a nuanced understanding of their effectiveness and optimization strategies (Daraojimba et al., 2024, p. 32). While the existing literature predominantly offers theoretical insights and case studies, there is a notable lack of comprehensive empirical analyses that evaluate the actual impact of these practices on software quality engineering in the European software industry (Merino et al., 2018, p. 166).

This literature review addresses an ongoing need to synthesize empirical evidence on optimizing Agile testing practices and quality management strategies an area where earlier studies, such as Gil et al. (2016), provided valuable foundational insights but did not fully capture the complexities and advancements in modern Agile implementation. By focusing on the European context, particularly within the Finnish software industry, this review contributes to the broader discourse on enhancing software quality and development efficiency through the strategic implementation of Agile methodologies (Daraojimba et al., 2024, p. 35).

2.1 Key Concepts and Overview of Agile Methodologies

The Agile movement officially began with the publication of the Agile Manifesto in 2001 by a group of seventeen software developers. This manifesto outlined four core values and twelve principles aimed at improving the process of software development by focusing on individuals and interactions, working software, customer collaboration, and responding to change (Beck & Andres, 2012, p. 4).

Before Agile, software development was dominated by the waterfall model, which is a linear and sequential approach. The waterfall model often led to issues such as inflexibility, late-stage defect discovery, and poor alignment with customer needs due to its rigid structure and lack of iterative feedback loops (Royce, 1970, p. 329).

The dissatisfaction with traditional methodologies led to the exploration of more flexible approaches. The 1990s saw the rise of various iterative and incremental development methods, such as Rapid Application Development (RAD) and Scrum. The formalization of these methods into the Agile Manifesto marked a significant shift in software development philosophy, emphasizing adaptability, collaboration, and customer satisfaction (Highsmith, 2000, p. 3).

Since its inception, Agile has been widely adopted across various industries and has evolved to include a range of frameworks and practices tailored to different project needs. The adoption of Agile has also been facilitated by the rise of DevOps, which integrates development and operations to enhance collaboration and streamline workflows. Agile practices continue to evolve, incorporating new tools and techniques to address emerging challenges in software development (Jabbari et al., 2016, p. 130).

Agile practices refer to a set of methodologies and principles that emphasize flexibility, collaboration, and customer-centricity in software development (Raheem & Osilaja, 2024, p. 12). Originating from the Agile Manifesto, Agile practices focus on iterative development, where requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams (Beck et al., 2001, p. 12). Key Agile methodologies include Scrum, Kanban, Extreme Programming (XP), and Lean Software Development. Agile methodologies represent a paradigm shift in software development, moving away from traditional, sequential (waterfall) models to more flexible, iterative approaches (Al-Saqqa et al., 2020, p. 174). The most widely adopted Agile methodolo-

gies include Scrum, Kanban, and Extreme Programming (XP). Each of these methodologies offers unique frameworks and practices tailored to enhance collaboration, responsiveness, and efficiency in software development (Al-Saqqa et al., 2020, p. 175).

Scrum is an Agile framework that facilitates teamwork and iterative progress towards a well-defined goal. It employs fixed-length iterations called sprints, typically lasting two to four weeks, during which a potentially shippable product increment is developed (Schwaber & Sutherland, 2020, p. 6). Scrum is one of the most popular Agile methodologies, characterized by its focus on iterative development, self-organizing teams, and regular feedback loops (Mahnic, 2017, p. 193). Scrum structures work into sprints, typically lasting two to four weeks, during which specific tasks are completed and reviewed (Schwaber & Sutherland, 2020, p. 9; Mahnic, 2017, p. 193). Key roles in Scrum include the Product Owner, Scrum Master, and Development Team (Schwaber & Sutherland, 2020, p. 5; livari & livari, 2022, p. 5). The Scrum framework emphasizes the importance of daily stand-up meetings, sprint planning sessions, sprint reviews, and retrospectives to ensure continuous improvement and alignment with project goals (Schwaber & Sutherland, 2020, pp. 11–13).

Kanban emphasizes continuous delivery without overburdening the development team. Work items are visualized on a Kanban board, allowing team members to track progress and manage workflow (Anderson, 2010, pp. 15–17). Kanban, originating from lean manufacturing principles, is a visual system for managing work as it moves through a process (Ahmad et al., 2018, p. 98). Kanban focuses on continuous delivery without overburdening the development team. It uses visual boards to represent the workflow, with columns indicating different stages of the process, such as "To Do," "In Progress," and "Done" (Ahmad et al., 2018, p. 101). Work items are represented as cards that move through these columns. Kanban promotes transparency, limits work in progress (WIP), and aims to optimize workflow efficiency by identifying and addressing bottlenecks (Anderson, 2010, pp. 22–25).

Extreme Programming (XP) enhances software quality and responsiveness to changing customer requirements through practices such as pair programming, test-driven development (TDD), and continuous integration (Beck & Andres, 2012, pp. 20–21). XP is an Agile methodology that emphasizes technical excellence and customer satisfaction. It includes practices such as TDD, continuous integration (CI), pair programming, and frequent releases in short development cycles (Williams & Cockburn, 2018, pp. 38–39). These practices aim to improve software quality and adaptability to evolving customer needs. XP promotes a high level of discipline and collaboration among team members, ensuring that the software product evolves in a robust and adaptive manner (Meckenstock, 2024, p. 17).

Lean Software Development aims to optimize efficiency by eliminating waste, amplifying learning, and delivering value quickly (Poppendieck & Poppendieck, 2010, pp. 24). Agile practices are grounded in four foundational principles, as articulated in the Agile Manifesto (Beck et al., 2001, p.2):

1. Customer collaboration over contract negotiation.
2. Responding to change over following a plan.
3. Individuals and interactions over processes and tools.
4. Working software over comprehensive documentation.

2.2 Software Quality Engineering

Software quality engineering encompasses a set of practices and methodologies aimed at ensuring that software products meet specified requirements and standards (Torkar et al., 2019, p. 13). It involves the systematic application of engineering principles to the design, development, testing, and maintenance of software, with a focus on quality assurance (QA) and quality control (QC) (Kundu et al., 2021, p. 55).

Quality assurance involves the systematic monitoring and evaluation of various aspects of a project to ensure that standards of quality are being met. QA activities include process audits, reviews, and validations (Juran & Godfrey, 1999, pp. 12).

Quality control refers to the activities and techniques used to achieve and maintain product quality, such as testing, inspections, and corrective actions. QC is typically performed at the end of the development cycle to identify and fix defects (Tariq et al., 2022, p. 2).

Test-Driven Development (TDD) is a software development process in which test cases are created to define and validate what the code should do. In TDD, tests are written before the code and guide the development process (Beck, 2015, p. 56).

Continuous Integration (CI) is a practice where developers frequently commit code to a shared repository, triggering automated builds and tests. CI aims to detect and resolve integration issues early in the development cycle (Fowler, 2024, p. 3). The overarching goal of software quality engineering is to deliver reliable, functional, and efficient software products that meet or exceed customer expectations and regulatory requirements.

2.3 Quality Management and Agile Software Practices

Quality management in software engineering is a critical aspect that ensures software products meet or exceed customer expectations. It encompasses a variety of principles, frameworks, and practices aimed at enhancing product quality and process efficiency (ISO, 2015, p. 1; Torkar, Feldt, & Awan, 2019, p. 14). Specially, Quality management in software development involves a set of practices and processes aimed at ensuring that the software products meet customer requirements and are free of defects. It includes quality planning, quality assurance, quality control, and quality improvement (ISO 9001:2015, 2015, p. 2).

Total Quality Management (TQM) is a management approach centered on quality, based on the participation of all members of an organization and aiming at long-term success through customer satisfaction (Deming, 2000, p. 23). This approach has evolved over the years, integrating new methodologies and technologies to enhance quality outcomes. For instance, recent studies highlight the integration of digital tools and continuous improvement processes as critical components of modern TQM practices (Oakland, 2012, p. 45).

ISO Standards the International Organization for Standardization (ISO) provides frameworks and standards for quality management systems, such as ISO 9001, which specify requirements for quality management systems (ISO 9001:2015, 2015, p.2). Quality management integrates various processes and methodologies to ensure the software meets the desired quality standards. It encompasses planning, control, assurance, and improvement of quality throughout the software development lifecycle (Project Management Institute, 2017, p. 289).

Six Sigma is a set of techniques and tools for process improvement. It seeks to improve the quality of the output by identifying and removing the causes of defects and minimizing variability in manufacturing and business processes (Pande et al., 2000, p. 12).

CMMI is a process-level improvement training and appraisal program. Administered by the CMMI Institute, it is used to guide process improvement across a project, division, or an entire organization (CMMI Institute, 2010, p. 2). CMMI defines the key elements of an effective process, provides a framework for improvement, and is used to develop and refine processes that improve performance. It includes best practices that address development and maintenance activities across the software product lifecycle (CMMI Institute, 2010, p. 2).

Quality management in software engineering is a critical aspect that ensures software products meet or exceed customer expectations. It encompasses a variety of principles, frameworks, and practices aimed at enhancing product quality and process efficiency (ISO, 2015, p. 1; Torkar, Feldt, & Awan, 2019, p. 14).

Quality assurance (QA) and quality control (QC) play vital roles in software development, ensuring that the final product meets predefined standards and satisfies user requirements (AltexSoft, 2023). As integral components of a broader quality management strategy, both QA and QC contribute to the delivery of high-quality software in Agile and traditional development environments.

Quality assurance refers to the systematic activities implemented within a quality system to ensure that quality requirements for a product or service will be fulfilled. It is a process-oriented approach that spans the entire software development lifecycle (SDLC), focusing on the methodologies, standards, and processes used during development. QA activities may include process checklists, audits, development of standardized methods, and staff training initiatives (Juran & Godfrey, 1999, p. 2). The principal aim of QA is to proactively prevent defects by improving the underlying development and testing processes before errors occur (Pressman, 2005, p. 795).

In contrast, quality control is product-oriented and consists of specific activities and techniques used to detect and correct defects after they have occurred. QC typically takes place toward the end of the development cycle and includes testing, inspections, and other verification activities intended to ensure that the software meets the required specifications. It involves executing the software with the intent of identifying faults and validating the final product (Feigenbaum, 2001, p. 95; Juran & Godfrey, 1999, p. 2.20).

Together, QA and QC form the foundation of software quality management, particularly in Agile contexts where continuous integration and frequent deliveries require both proactive and reactive quality measures.

Agile practices are closely linked to quality management as they incorporate continuous feedback loops, iterative development, and customer collaboration, all of which are vital for maintaining high-quality standards in software development (Mendix, 2024; TestRail, 2024). The integration of Agile practices and quality management strategies ensures that software products are not only functional and reliable but also meet customer needs and regulatory requirements (López et al., 2021, p. 2; AgileVelocity, 2023).

Agile methodologies promote iterative development, where each iteration includes phases of planning, development, testing, and review. Continuous feedback from customers and stakeholders helps in identifying defects early and incorporating changes efficiently (Schwaber & Sutherland, 2020, p. 8).

Agile emphasizes collaboration among team members and stakeholders, enhancing communication and ensuring that everyone is aligned with the project goals. This collaborative approach helps in identifying potential quality issues early and addressing them promptly (Beck et al., 2001, p. 4).

Agile practices allow teams to respond to changes quickly, whether they are changes in requirements, market conditions, or technological advancements. This flexibility ensures that the final product remains relevant and meets quality standards (Dingsøyr et al., 2012, pp. 1215–1216).

By embedding quality management principles into Agile practices, software development teams can achieve a balance between speed and quality, ensuring the delivery of

high-quality software products that satisfy customer expectations and adhere to industry standards.

2.4 Core Principles and Practices of Agile Methodologies

The core principles and practices of Agile methodologies are rooted in the Agile Manifesto and its twelve supporting principles, which collectively guide Agile teams toward delivering high-quality software efficiently and effectively. Central to these principles is customer satisfaction, achieved by delivering valuable software early and continuously. Agile embraces welcoming change, allowing for evolving requirements even late in the development cycle to maintain a competitive edge. Frequent delivery of working software is encouraged, with a preference for shorter timescales to promote responsiveness. Collaboration between business stakeholders and developers is essential and should occur daily throughout the project (Williams & Cockburn, 2018, p. 38).

Agile methodologies emphasize building projects around motivated individuals, offering them the support and trust needed to succeed. Face-to-face conversation is considered the most effective way to communicate within development teams. Progress is primarily measured through working software, reinforcing the importance of tangible results. Sustainable development is promoted by maintaining a consistent work pace indefinitely (Williams & Cockburn, 2018, p. 39).

Continuous attention to technical excellence and good design enhances agility. Simplicity, defined as maximizing the amount of work not done, is a core tenet. Agile also values self-organizing teams, which are believed to produce the best architectures, requirements, and designs. Finally, reflection and adjustment are encouraged through regular intervals of team introspection, aiming to improve processes and performance continuously (Beck et al., 2001, pp. 1–2; Williams & Cockburn, 2018, p. 39).

2.5 Integration of Quality Management Practices within Agile Environments

Integrating quality management practices within Agile environments presents unique challenges and opportunities (Sharma, 2023, p. 14). Agile methodologies, known for their flexibility and iterative nature, align well with the principles of continuous improvement and customer focus emphasized in quality management frameworks like TQM and Six Sigma (Sharma, 2023, p. 14).

Agile methodologies emphasize frequent delivery of working software, customer collaboration, and responsiveness to change (Salah, Paige, & Cairns, 2019, p. 3). Quality management in Agile environments involves integrating QA and QC activities throughout the development process rather than at the end (Brosseau, 2019, p. 2). This integration ensures continuous feedback and improvement, aligning with Agile principles (Beck et al., 2001, p. 1).

Continuous Integration (CI) involves integrating code into a shared repository several times a day. Each integration is verified by an automated build and automated tests, allowing teams to detect problems early (Fowler, 2006, p. 15). CI ensures that the codebase remains in a deployable state, facilitating frequent releases and high-quality standards (Fowler, 2020, p. 16).

In TDD, developers write tests before writing the code. This practice ensures that the code meets the required specifications and reduces the number of defects. TDD fosters better design, improved code quality, and continuous integration, which are crucial for maintaining quality in Agile environments (Kundu, Patel, & Bhatt, 2021, p. 3).

Pair programming involves two developers working together at one workstation. One writes the code while the other reviews it. This practice improves code quality through

continuous review and feedback, ensuring that defects are identified and fixed early (Williams & Kessler, 2003, p. 20).

Continuous delivery extends Continuous implementation by ensuring that code changes are automatically prepared for release to production. CD ensures that the software can be reliably released at any time, providing a faster and more efficient deployment process, which is essential for maintaining high-quality standards in Agile environments (Humble & Farley, 2010, p. 6).

By incorporating these Agile practices, organizations can ensure that quality management is an integral part of the development process. This approach not only enhances product quality but also aligns with the Agile principles of delivering value to customers, fostering collaboration, and responding to change.

2.6 Analysis of Development Strategies that Support Agile Practices

Iterative and incremental development is a cornerstone strategy in Agile environments, emphasizing the delivery of small, workable segments of software at regular intervals. This approach contrasts with the traditional waterfall model, where the software is developed as a whole in sequential stages (Wrike, 2024, p. 2).

In Agile, each iteration involves a complete cycle of planning, development, testing, and review, enabling teams to integrate feedback and make necessary adjustments promptly (Scaled Agile Framework, 2023, p. 1). This strategy minimizes risks, as potential issues can be identified and resolved early in the development process. Iterative development also enhances the ability to respond to changing requirements, ensuring that the final product is more closely aligned with user needs (Larman & Basili, 2003, p. 48).

Adaptive planning is another critical strategy in Agile environments. Unlike traditional project management, which relies on static, long-term plans, Agile methodologies employ dynamic planning processes that are continuously updated based on new insights

and feedback (Project Management Institute, 2013, p. 152). This approach, often referred to as rolling wave planning, allows teams to remain flexible and responsive to changes.

Adaptive planning involves setting short-term goals that are revisited and revised at the end of each iteration or sprint, facilitating ongoing alignment with project objectives and stakeholder expectations (Highsmith, 2009, p. 58). The iterative nature of Agile planning helps manage uncertainty and reduces the impact of unforeseen challenges.

Evolutionary design supports the Agile principle of accommodating change by allowing the software design to evolve over time (Fowler, 2005, para. 2). This approach differs from the traditional upfront design, which often proves inflexible in the face of changing requirements (Casey, 2017, p. 11). Evolutionary design involves continuously improving and adapting the software architecture and codebase as new requirements emerge and as the understanding of the problem domain deepens.

Refactoring, or the process of restructuring existing code without changing its external behavior, is a key practice in evolutionary design (ThoughtWorks, 2012, p. 1). It ensures that the code remains clean, efficient, and maintainable, which is crucial for accommodating future changes (Fowler, 2018, p. 109).

Continuous Integration and Continuous Delivery (CI/CD)

Continuous Integration (CI) and Continuous Delivery (CD) are essential strategies in Agile environments that facilitate frequent and reliable software releases (Humble & Farley, 2010, pp. 4–6). CI involves integrating code changes into a shared repository several times a day, followed by automated builds and tests to detect integration issues early. This practice helps ensure that the codebase remains stable and that defects are identified and addressed promptly (Red Hat, 2023, para. 3). Continuous Delivery extends CI by automating the deployment process, ensuring that the software can be released to

production at any time. These practices reduce the risks associated with large, infrequent releases and enable teams to deliver value to customers more quickly and reliably (Humble & Farley, 2010, p. 9).

2.7 Cross-Functional Teams

Agile methodologies advocate for cross-functional teams that encompass all the skills necessary to deliver a complete product increment (Scaled Agile, Inc., 2023, p. 8). Cross-functional teams enhance collaboration, streamline communication, and reduce dependencies on external entities. This integrated approach allows team members to work closely together, share knowledge, and collectively address challenges, leading to more efficient and effective development processes (ProofHub, 2023, para. 5).

Cross-functional teams are also better positioned to understand and meet customer needs, as they bring together diverse perspectives and expertise, which contributes to faster decision-making and improved product quality (Jalali & Wohlin, 2012, p. 90).

2.8 Case Studies and Empirical Research on Effective Development Strategies in Agile Projects

Spotify's implementation of Agile methodologies provides a compelling case study of effective development strategies. Spotify employs a unique model comprising autonomous squads, tribes, chapters, and guilds, each with specific roles and responsibilities. Squads operate like mini-startups within the organization, focusing on delivering specific features or services (Scaled Agile, Inc., 2023, p. 14). Tribes consist of multiple squads working toward a common goal, while chapters and guilds facilitate cross-squad collaboration and knowledge sharing. This model promotes iterative development, adaptive planning, and continuous feedback, resulting in faster delivery cycles, improved product quality, and higher employee satisfaction (Kniberg & Ivarsson, 2012, pp. 2–3).

Empirical research by Dingsøy et al. (2012, p. 1215) examines the impact of Agile practices on project performance in the software industry. The study reveals that Agile practices such as iterative development, continuous integration, and regular feedback loops significantly enhance project outcomes, including product quality, customer satisfaction, and team productivity. The research underscores the importance of aligning development strategies with Agile principles to achieve optimal results (Boehm & Turner, 2005, p. 33). It also highlights the need for ongoing empirical studies to refine and adapt Agile practices to different contexts and challenges (Dingsøy et al., 2012, p. 1220).

2.9. Conceptual Framework and Justification for Combining Dependent Variables

The conceptual framework for this study is designed to investigate the relationships between various Agile practices and the combined dependent variable of Software Quality and Customer Satisfaction within the Finnish software industry (Gil et al., 2016). The study identifies five key independent variables that are hypothesized to influence this combined outcome. These variables include the adoption of Agile testing practices, adherence to Agile principles, frequency of Agile iterations, level of test automation, and the integration of customer feedback (Humble & Farley, 2010). Each of these variables represents a critical aspect of Agile methodologies that can impact the success of software development projects.

Adoption of Agile Testing Practices ensures that software is continuously verified and improved. Practices like Test-Driven Development (TDD), Continuous Integration (CI), and Behavior-Driven Development (BDD) are associated with reduced defect rates and higher reliability (Gil et al., 2016, p. 28).

Adherence to Agile Principles, such as iterative development, collaboration, and responsiveness to change, contributes positively to quality outcomes. High alignment

with these principles has been shown to enhance project success rates (Chow & Cao, 2008, p. 964).

Frequency of Agile Iterations refers to the regularity with which development cycles (sprints) occur. Frequent iterations are associated with quicker identification and resolution of defects, which can significantly enhance software quality (Petersen & Wohlin, 2010, p. 686). Furthermore, frequent releases allow for continuous customer feedback, which is essential for maintaining high levels of customer satisfaction.

Test Automation plays a vital role in ensuring that software is both reliable and scalable. Automated tests can be run frequently and consistently, providing immediate feedback on the quality of the software. Studies have shown that companies with high levels of test automation report significantly better software quality outcomes (Fowler & Highsmith, 2001, p. 5).

Customer Feedback Integration is the process of incorporating customer input into the development process. Agile methodologies emphasize the importance of this feedback loop, as it ensures that the software evolves to meet user needs. This integration directly contributes to both software quality and customer satisfaction, as products are more likely to align with customer expectations (López et al., 2021, p. 3).

Justification for Combining Dependent Variables

Combining software quality and customer satisfaction into a single dependent variable is justified due to the interconnected nature of these two concepts, especially within Agile environments. In Agile development, software quality is not just a technical measure but also a reflection of how well the software meets customer needs. High-quality software that is free from defects and is user-friendly directly impacts customer satisfaction, as users are more likely to be pleased with a product that works well and meets their expectations (DeLone & McLean, 2003, p. 16). Moreover, Agile practices inher-

ently tie together the development process with customer feedback, making it practical to assess software success through a unified lens of quality and satisfaction (Seddon, 1997, p. 241; Petter, DeLone, & McLean, 2008, p. 244).

This combined dependent variable aligns with the practical goals of Agile development, where the end goal is to deliver software that is both technically sound and well-received by users. By using a combined measure, the study captures the holistic impact of Agile practices on the overall success of software projects (Khalifa & Liu, 2003, p. 5).

Conceptual Framework Diagram

As a conclusion of the literature review, the author developed the conceptual framework presented in Figure 1. This framework will guide the research by systematically examining how each Agile practice contributes to the combined outcome of Software Quality and Customer Satisfaction in the Finnish software industry.

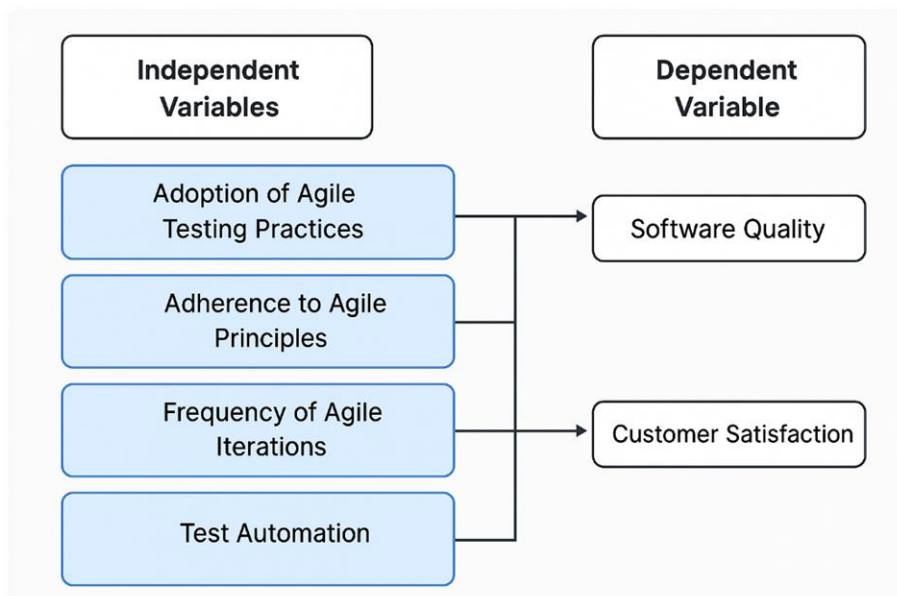


Figure 1. Conceptual Framework designed by author

3 Methodology

This chapter details the methodology employed to investigate the optimization of Agile testing practices and development strategies within the Finnish software industry. The study is driven by the need to empirically validate the effectiveness of various Agile methodologies and their impact on software quality and customer satisfaction.

3.1 Population and sample size

The population for this study consists of IT professionals working in companies within the Finnish software industry. According to Statistics Finland (2023), there are approximately 116,630 IT companies operating in Finland, making it a significant sector within the country's economy. The Finnish software industry is highly diverse, encompassing a wide range of companies, from small startups to large multinational corporations (Nylund, Kontio, & Harkke, 2023, p. 22). These companies are engaged in various aspects of software development, including product development, consulting, and services. This diversity provides a rich context for studying Agile practices, as different types of organizations may adopt and implement Agile methodologies in unique ways.

The decision to focus on IT professionals, particularly those involved in software development, quality assurance, and project management, is driven by the need to obtain insights from individuals who are directly engaged with Agile methodologies (Kalenda, Hyna, & Rossi, 2018, p. 4). These professionals play crucial roles in the software development lifecycle, making their experiences and perspectives invaluable for understanding how Agile practices influence software quality. Additionally, targeting this population allows the study to capture a comprehensive view of the software development process, from initial planning and design to testing and deployment (Schwaber & Sutherland, 2020, p. 5).

This population is particularly relevant to the research objectives, which aim to optimize Agile testing practices and development strategies to enhance software quality

in the Finnish software industry. By focusing on professionals who actively participate in these processes, the study ensures that the data collected is both relevant and actionable. Moreover, the insights gained from this population can be used to develop practical recommendations that are grounded in the real-world experiences of those working in the industry (Boehm & Turner, 2005, p. 32).

Furthermore, the Finnish software industry is known for its strong emphasis on innovation and technology adoption, making it an ideal context for studying Agile methodologies. Finland's technology sector is characterized by a high level of digital maturity, with companies frequently adopting cutting-edge practices to maintain competitive advantages (European Commission, 2020, p. 9). This environment is conducive to the study's goals, as it is likely to yield rich data on the various ways Agile practices are implemented and their impact on software quality.

Finally, focusing on this specific population allows for a more targeted analysis of Agile practices within a defined cultural and organizational context. By examining the experiences of IT professionals in Finland, the study can provide insights that are directly applicable to the local industry, while also contributing to the broader body of knowledge on Agile methodologies and software quality management (Dingsøy, Nerur, Balijepally, & Moe, 2012, p. 1216). This approach not only enhances the relevance of the findings for Finnish companies but also adds value to the global understanding of how Agile practices can be optimized in different environments.

To determine an appropriate sample size, the study follows established guidelines for ensuring adequate power in statistical analyses, particularly for regression analysis. Regression analysis requires a reasonably large sample size to detect significant relationships between variables, especially when multiple independent variables are involved (Field, 2018, p. 72).

Given that the research aims to achieve a broad representation of the Finnish software industry, the study targets approximately 500 IT professionals. Assuming a response rate of about 30%, which is typical for online surveys in professional contexts (Fulton, 2018, p. 14), around 650 survey invitations will be distributed. This approach is expected to yield at least 195 valid responses, which is considered adequate for conducting regression analysis and other statistical tests required to evaluate the research hypotheses.

A non-probability convenience sampling method is employed, where survey invitations are sent to IT professionals working in companies known to use Agile practices. This method is suitable given the specific expertise required from respondents, as they need to have experience with Agile methodologies to provide relevant insights. While convenience sampling may introduce some bias due to the non-random selection of participants, it is a pragmatic choice given the challenges of accessing a large and varied population within a limited time frame and resource constraints (Etikan, Musa, & Alkassim, 2016, p. 2).

The chosen sampling strategy aligns well with the research objectives. By focusing on IT professionals with relevant experience in Agile practices, the study ensures that the responses are informed by practical, real-world knowledge of the factors influencing software quality in Agile environments. This targeted approach enhances the validity of the findings, as it gathers data from individuals who are directly involved in the practices under investigation (Saunders, Lewis, & Thornhill, 2019, p. 296).

Furthermore, the decision to target a larger sample size, beyond the minimum required, strengthens the generalizability of the results. A larger sample increases the likelihood of capturing the diversity within the Finnish software industry, thus providing a more comprehensive understanding of the impact of Agile practices on software quality (Cochran, 1977, p. 75).

The sampling strategy employed in this study is well-suited to the research objectives and hypotheses. It ensures that the data collected will be robust enough to support the statistical analyses required, ultimately leading to meaningful and actionable insights into optimizing Agile practices in the Finnish software industry.

3.2 Data Collection and Hypotheses

The questionnaire for this study was meticulously designed to align with the research objectives and hypotheses, as outlined in the conceptual framework. The primary aim was to develop a tool that could empirically assess the relationships between various Agile practices and the combined dependent variable of software quality and customer satisfaction within the Finnish software industry.

Research hypotheses have been formulated based on the conceptual framework and research objectives.

Hypothesis 1 (H1): There is a statistically significant positive relationship between the adoption of Agile testing practices and the level of software quality in the Finnish software industry.

Null Hypothesis (H0): There is no statistically significant relationship between the adoption of Agile testing practices and the level of software quality in the Finnish software industry.

Hypothesis 2 (H2): The extent of Agile principles adherence in development teams positively influences the effectiveness of software quality management processes in Finland.

Null Hypothesis (H0): The extent of Agile principles adherence in development teams does not significantly influence the effectiveness of software quality management processes in Finland.

Hypothesis 3 (H3): Increased frequency of Agile iterations is associated with a significant reduction in software defects within Finnish software projects.

Null Hypothesis (H0): Increased frequency of Agile iterations is not associated with a significant reduction in software defects within Finnish software projects.

Hypothesis 4 (H4): Companies that implement high levels of test automation in Agile environments report significantly higher software quality in Finnish software projects.

Null Hypothesis (H0): Companies that implement high levels of test automation in Agile environments do not report significantly higher software quality in Finnish software projects.

Hypothesis 5 (H5): Agile development strategies that prioritize customer feedback loops are positively correlated with higher customer satisfaction and software quality outcomes in the Finnish software industry.

Null Hypothesis (H0): Agile development strategies that prioritize customer feedback loops are not positively correlated with higher customer satisfaction and software quality outcomes in the Finnish software industry.

The following table summarizes how the questionnaire was structured to address each hypothesis, research objective, and associated variables.

Mapping the Sectors to Research Objectives, Questions, and Hypotheses

Hypothesis	Research Objective	Independent Variable	Dependent Variable	Survey Questions
H1	Objective 1	Adoption of Agile Testing Practices	Software Quality and Customer Satisfaction	Q1-Q5
H2	Objective 1	Adherence to Agile Principles	Software Quality and Customer Satisfaction	Q6-Q10
H3	Objective 2	Frequency of Agile Iterations	Software Quality and Customer Satisfaction	Q11-Q15
H4	Objective 2	Test Automation	Software Quality and Customer Satisfaction	Q16-Q20
H5	Objective 3	Customer Feedback Integration	Software Quality and Customer Satisfaction	Q21-Q25

Table 1. Mapping Research Objectives, Questions and Hypotheses

The design process began by mapping each hypothesis to specific sections of the questionnaire, ensuring that all aspects of the research were comprehensively addressed. For example, Hypothesis One, which investigates the relationship between the adoption of Agile testing practices and software quality, was aligned with targeted questions measuring relevant practices and outcomes. Each of the remaining hypotheses was similarly linked to thematically appropriate items, allowing the questionnaire to generate data directly related to the core research questions (Fowler, 2014, p. 105). The explanation and justification of each hypothesis are presented below, reflecting their alignment with the conceptual framework and research objectives.

Hypothesis one (H1) maps directly to questions regarding Agile testing practices, focusing on how these practices influence software quality. This aligns with Research Objective one, which aims to empirically examine these practices.

Hypothesis two (H2) explores the impact of adhering to Agile principles on software quality management. The corresponding questions assess how closely teams follow

these principles and their effects on software quality, also contributing to Research Objective one.

Hypothesis three (H3) examines the frequency of Agile iterations and their impact on reducing software defects, which is crucial for Research Objective two that identifies key factors influencing software quality engineering.

Hypothesis four (H4) focuses on the role of test automation in improving software quality, with questions designed to understand the effectiveness and impact of automation, directly supporting Research Objective two.

Hypothesis five (H5) investigates how integrating customer feedback influences software quality and customer satisfaction, supporting Research Objective three by providing recommendations for optimizing Agile practices.

The questionnaire was structured using a five-point Likert scale, ranging from "Strongly Disagree" to "Strongly Agree," allowing respondents to express the extent of their agreement or disagreement with each statement. This scale is widely recognized for its effectiveness in capturing attitudes and perceptions in quantitative research (Joshi et al., 2015, p. 397). The use of a five-point scale was particularly chosen for its ability to balance simplicity with the richness of data, offering respondents enough options to express their views while maintaining clarity and ease of response (DeVellis, 2017, p. 110).

In designing the questionnaire, clarity, relevance, and neutrality were prioritized to avoid leading or biased questions. This approach is consistent with best practices in survey design, which emphasize clear and unbiased questions to ensure data validity and reliability (Brace, 2018, p. 72). Each question was tailored to the context of Agile practices within the Finnish software industry, ensuring relevance and applicability of the responses.

To further validate the questionnaire, it was reviewed by industry experts and academics familiar with Agile methodologies and software quality management. Their feedback was used to refine the questions, ensuring they were both understandable and capable of capturing the intended data. Pre-testing or pilot testing of the questionnaire was also conducted with a small group of IT professionals to identify any ambiguities or issues with the survey instrument, following the recommendation of Saunders et al. (2019, p. 436) for enhancing the robustness of survey tools.

This rigorous design process aimed to ensure that the questionnaire would effectively capture the key variables of interest, providing the data necessary to test the hypotheses and achieve the research objectives.

Data Collection

The data collection for this study was conducted through a multi-channel approach, leveraging both digital and personal networks to maximize reach and response rates. The questionnaire was distributed primarily via online platforms, which is a widely accepted method in contemporary empirical research due to its efficiency and broad reach (Dillman, Smyth, & Christian, 2014, p. 7).

The questionnaire was first distributed via email to IT professionals working in Finnish software companies. Email lists were compiled from professional networks, including LinkedIn groups focused on Agile development and software engineering. This method was chosen for its ability to directly target individuals with relevant expertise, ensuring that the respondents had the necessary background to provide informed answers (Evans & Mathur, 2018, p. 117).

LinkedIn was used as a key platform for distributing the survey, especially within groups dedicated to Agile methodologies, software development, and IT management in Finland. LinkedIn offers a highly professional environment where industry-specific groups can be leveraged to reach a large number of relevant participants efficiently (Baltar & Brunet, 2012, p. 57). Posts were made in these groups, accompanied by personalized messages to encourage participation.

The questionnaire was also sent directly to companies known to adopt Agile practices. This approach was intended to secure responses from professionals actively engaged in Agile software development. Direct company emails were particularly effective in reaching larger organizations where internal distribution could lead to higher participation rates (Groves et al., 2009, p. 140).

Personal contacts within the industry were leveraged to distribute the survey, with respondents encouraged to share the questionnaire with their colleagues. This snowball sampling technique is commonly used in empirical research to expand the participant base beyond initial contacts (Goodman, 2011, p. 22). By using this approach, the study was able to tap into broader professional networks, thereby increasing the diversity and representativeness of the sample.

In addition to digital methods, the questionnaire was distributed through professional associations and at industry conferences, either virtually or in person. These channels were used to target respondents who are actively engaged in the Finnish software industry, further enhancing the reliability and relevance of the data collected.

This combination of distribution methods ensured a wide reach across various segments of the Finnish software industry, thereby increasing the likelihood of obtaining a representative sample. Moreover, the use of multiple channels helped to mitigate potential biases associated with any single method, contributing to the overall validity and reliability of the study (Bryman, 2016, p. 214).

Through this comprehensive and strategically planned data collection process, the study was able to gather rich and diverse data, which is crucial for testing the research hypotheses and achieving the study's objectives.

3.3 Quantitative Data Analysis

The data analysis conducted in this study was structured to rigorously test the research hypotheses and to examine the relationships between Agile practices and software quality outcomes within the Finnish software industry. The methodological design also reflects the increasing relevance of emerging technologies such as integrated digital platforms and virtualized environments which support the iterative, flexible, and collaborative nature of Agile practices in dynamic and performance-critical contexts (Veronika & Jayathilaka, 2020). In line with the quantitative research design, a combination of statistical techniques including correlation and regression analysis was applied to ensure that the results are both robust and generalizable. These methods also support the reliability and validity of the findings by providing empirical evidence for the hypothesized associations.

This study employs a cross-sectional quantitative research design to investigate the relationships between Agile practices and software quality in the Finnish software industry. The data collection method involves a structured survey targeting IT professionals in Finland, focusing on those involved in software development, quality assurance, and project management (Lavrakas et al., 2019, p. 342).

To analyse the collected data, statistical techniques such as regression analysis and correlation analysis will be used. Regression analysis is particularly suitable for this study as it allows for the examination of the relationships between multiple independent variables and a dependent variable (Montgomery, Peck, & Vining, 2012, pp. 45–47). In this

case, regression analysis will help identify which Agile practices most significantly impact software quality and customer satisfaction (Yang et al., 2023, p. 301). Additionally, correlation analysis will be employed to explore the strength and direction of the relationships between variables (West et al., 2017, p. 89).

The use of these statistical approaches ensures that the study can provide robust, evidence-based recommendations for optimizing Agile practices within the Finnish software industry (Kettunen et al., 2019, p. 14). This design is in line with best practices for quantitative research, where the focus is on identifying and measuring relationships between variables to draw meaningful conclusions (Field, 2018, pp. 216–218).

The initial phase of the data analysis involves descriptive statistics, which are used to summarize the survey data and provide an overview of the sample characteristics. This includes calculating measures such as the mean, median, standard deviation, and frequency distributions for each variable. Descriptive statistics help to identify patterns and trends within the data, offering a preliminary understanding of the respondents' practices and perceptions regarding Agile methodologies and software quality (Pallant, 2020, p. 53).

To ensure the reliability of the questionnaire and assess the internal consistency of the measurement scales, Cronbach's alpha was calculated for each set of items representing key dimensions of Agile practices, such as the adoption of Agile testing methods and adherence to Agile principles. A Cronbach's alpha value exceeding 0.70 was considered acceptable, indicating that the items within each scale consistently measure the same underlying construct (Tavakol & Dennick, 2011, p. 54). This statistical approach strengthens the trustworthiness of the survey instrument used in the study.

Following the descriptive analysis, Pearson's correlation coefficient was employed to assess the strength and direction of the relationships between the independent variables such as Agile testing practices and test automation and the dependent variables,

namely software quality and customer satisfaction. This correlation analysis provided insights into which Agile practices were most strongly associated with improved software outcomes, thereby identifying potential areas for strategic focus and optimization (Dancey & Reidy, 2014, p. 191).

The primary method of hypothesis testing in this study is multiple regression analysis. This statistical technique is used to determine the extent to which the independent variables predict the dependent variables. In this context, multiple regression analysis allows for the examination of the combined effect of various Agile practices on software quality and customer satisfaction. By analysing the beta coefficients, the study can identify which practices have the most significant impact, thereby validating or refuting the proposed hypotheses (Montgomery, Peck, & Vining, 2012, p. 56).

Regression analysis is particularly suited to this study because it allows for the control of potential confounding variables, ensuring that the relationships identified are not spurious but reflective of true underlying associations (Field, 2018, p. 319). For instance, the analysis can control for variables such as company size or industry sector, which might otherwise influence the results.

Hypothesis Testing: Each hypothesis is tested using the results from the regression analysis. For each independent variable, the study examines the p-value to determine statistical significance, typically using a threshold of 0.05. If the p-value is less than 0.05, the null hypothesis (H_0) is rejected, supporting the alternative hypothesis (H_1) that suggests a significant relationship exists (Cohen, Cohen, West, & Aiken, 2003, p. 82). This approach ensures that the study's findings are not only statistically significant but also practically relevant for improving Agile practices in the software industry.

Model Diagnostics: To ensure the validity of the regression models, several diagnostic tests are conducted. These include tests for multicollinearity (using Variance Inflation Factor, VIF), heteroscedasticity (using Breusch-Pagan test), and normality of residuals

(using the Shapiro-Wilk test). These diagnostics help to verify that the assumptions of regression analysis are met, which is crucial for the reliability of the results (Hair, Black, Babin, & Anderson, 2010, pp. 110–112).

3.4 Study Plan and Justification

The analysis begins with a thorough exploratory data analysis (EDA), which includes the computation of descriptive statistics and the assessment of data quality. This step is critical to ensure that the data is clean and that any outliers or missing values are addressed before proceeding to more complex analyses (Tabachnick & Fidell, 2013, pp. 123–125). Once the data is prepared, the correlation and regression analyses are conducted to explore the relationships between the various Agile practices and software quality outcomes. These methods are chosen for their ability to quantify the strength of associations and predict the impact of multiple factors simultaneously, making them well-suited to the study's objective of optimizing Agile practices in the Finnish software industry (Montgomery, Peck, & Vining, 2012, pp. 27–30).

The use of reliability analysis ensures that the questionnaire is measuring the intended constructs consistently, thereby enhancing the validity of the findings. By integrating these statistical techniques, the study can draw robust conclusions about which Agile practices are most effective for improving software quality and customer satisfaction, ultimately guiding recommendations for industry best practices (Tavakol & Dennick, 2011, p. 53; Pallant, 2020, p. 89).for industry best practices (Tavakol & Dennick, 2011, Pallant, 2020).

4 Results

This chapter presents the findings derived from the survey responses, which provide insights into the demographic composition of respondents and the contextual characteristics of their organizations. The survey was conducted to explore the relationships between Agile practices and their impact on software quality and customer satisfaction within the Finnish software industry. A total of 670 invitations were sent via email and other professional platforms to IT professionals across Finland. Out of the 205 responses received, 200 responses were submitted within the designated timeframe and were included in the analysis. The remaining 5 responses were excluded due to late submissions. This resulted in a response rate of 30%, which is consistent with expectations for professional surveys of this nature (Fulton, 2018, p. 52).

The respondents represent a diverse cross-section of roles within the software development lifecycle, spanning technical, managerial, and strategic positions. This diversity ensures that the data collected provides a comprehensive view of Agile practices as implemented in the Finnish software industry. The analysis begins with an overview of general questions that outline the respondents' job roles, organizational size, Agile methodology usage duration, and primary Agile frameworks used. These findings establish a foundation for the subsequent examination of Agile practices and their effects on software quality outcomes.

The graphical analysis of these aspects, represented through various bar charts, provides a visual affirmation of the quantitative data, offering clear insights into the prevailing Agile practices among Finnish IT professionals. The charts not only delineate the levels of adoption and adherence to these practices but also illustrate the commitment to integrating Agile principles deeply within organizational processes. This comprehensive analysis helps in understanding the current landscape of Agile methodologies in Finland and guides future strategies for enhanced software quality and customer satisfaction.

Job Role of the Respondent

The survey respondents represent a wide range of roles critical to Agile implementation.

As illustrated in Table 2 and Figure 2, the distribution of job roles is as follows:

		Role of the organization			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Software Developer	92	46.0	46.0	46.0
	QA Engineer	42	21.0	21.0	67.0
	Project Manager	25	12.5	12.5	79.5
	Agile Coach	15	7.5	7.5	87.0
	Other	26	13.0	13.0	100.0
	Total	200	100.0	100.0	

Table 2. Survey Results on Job Role Distribution of Respondents

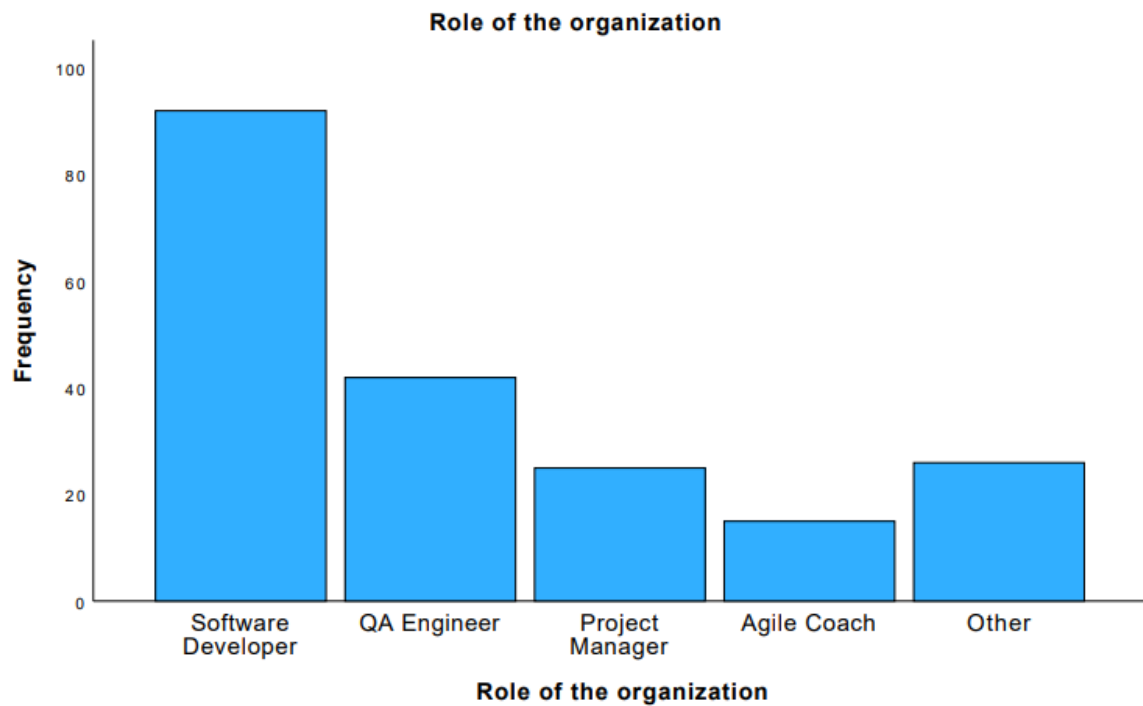


Figure 2. Distribution of Responses on Job Role Distribution of Respondents

In the surveyed group, Software Developers constitute the largest segment, representing 46.0% (n = 92) of participants, which emphasizes their pivotal role in Agile environments. They are followed by Quality Assurance Engineers who make up 21.0% (n = 42), underscoring the importance of quality assurance in Agile methodologies. Project Managers, crucial for orchestrating Agile project coordination, account for 12.5% (n = 25) of the total. Agile Coaches, who facilitate Agile transitions, comprise 7.5% (n = 15). The "Other" category, which includes DevOps Engineers, architects, and other specialized roles, constitutes 13.0% (n = 26) of the participants. This distribution highlights the collaborative nature of Agile methodologies that benefit from the contributions of a diverse mix of technical and managerial roles, reflecting the methodology's focus on quality and iterative development (Highsmith, 2009, pp. 45–46).

Organization Size

The survey responses also provide insights into the size of the organizations represented, as shown in Table 3 and Figure 3:

		organization_size			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Small 1-50	80	40.0	40.0	40.0
	Medium 51-200	79	39.5	39.5	79.5
	Large 201-500	35	17.5	17.5	97.0
	Very Large 500 more	5	2.5	2.5	99.5
	Other	1	.5	.5	100.0
	Total	200	100.0	100.0	

Table 3. Survey Results on Organization Size Distribution

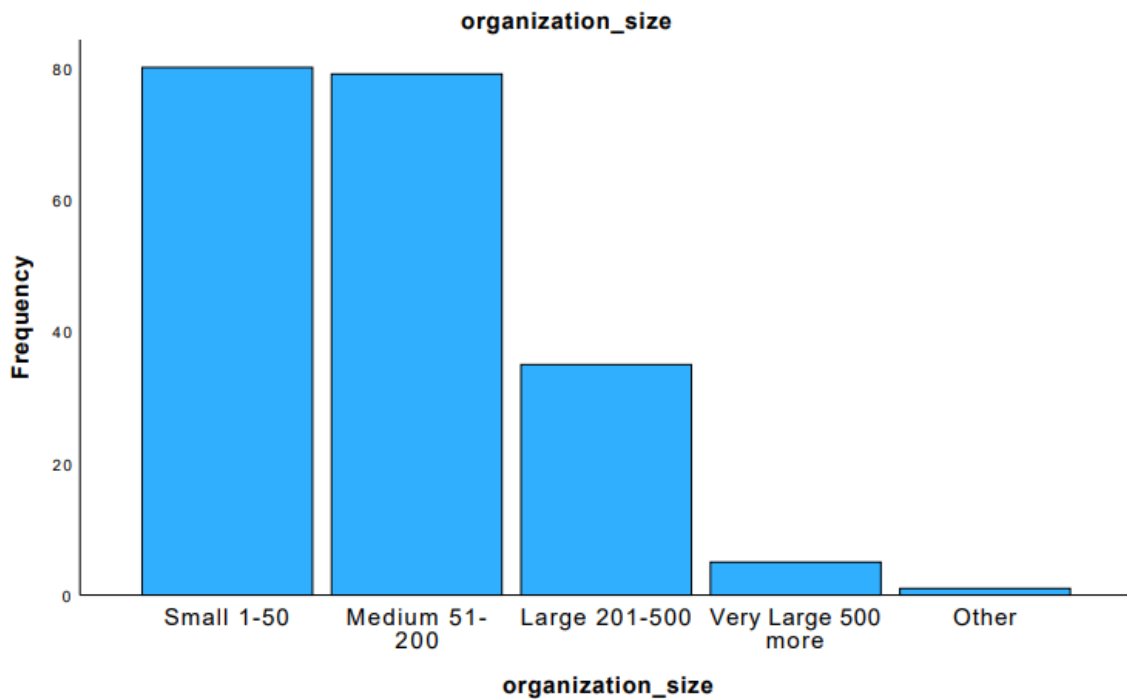


Figure 3. Distribution of Responses on Organization Size Distribution

Small organizations with 1 to 50 employees comprise the largest portion, accounting for 40.0% (n = 80) of the respondents. They are closely followed by medium-sized organizations with 51 to 200 employees, which make up 39.5% (n = 79). Larger organizations with 201 to 500 employees represent 17.5% (n = 35) of the total, and very large organizations with more than 500 employees constitute 2.5% (n = 5). An additional 0.5% (n = 1) of the sample falls into the "Other" category. This distribution demonstrates Agile methodologies' adaptability across various organizational sizes. The prevalent representation of small to medium-sized organizations might suggest their flexibility and quicker adaptability to Agile practices, consistent with findings from literature that emphasize Agile's scalability across different organizational contexts (Chow & Cao, 2008, p. 964).

Agile Methodology Usage Duration

The duration of Agile methodology usage varied across organizations, reflecting differing levels of maturity in implementation. As depicted in Table 4 and Figure 4:

		experience_years			Cumulative Percent
		Frequency	Percent	Valid Percent	
Valid	Less than 1 year	21	10.5	10.5	10.5
	1-3 years	35	17.5	17.5	28.0
	4-6 years	64	32.0	32.0	60.0
	7-10 years	35	17.5	17.5	77.5
	More than 10 years	45	22.5	22.5	100.0
Total		200	100.0	100.0	

Table 4. Survey Results on Duration of Agile Methodology Usage

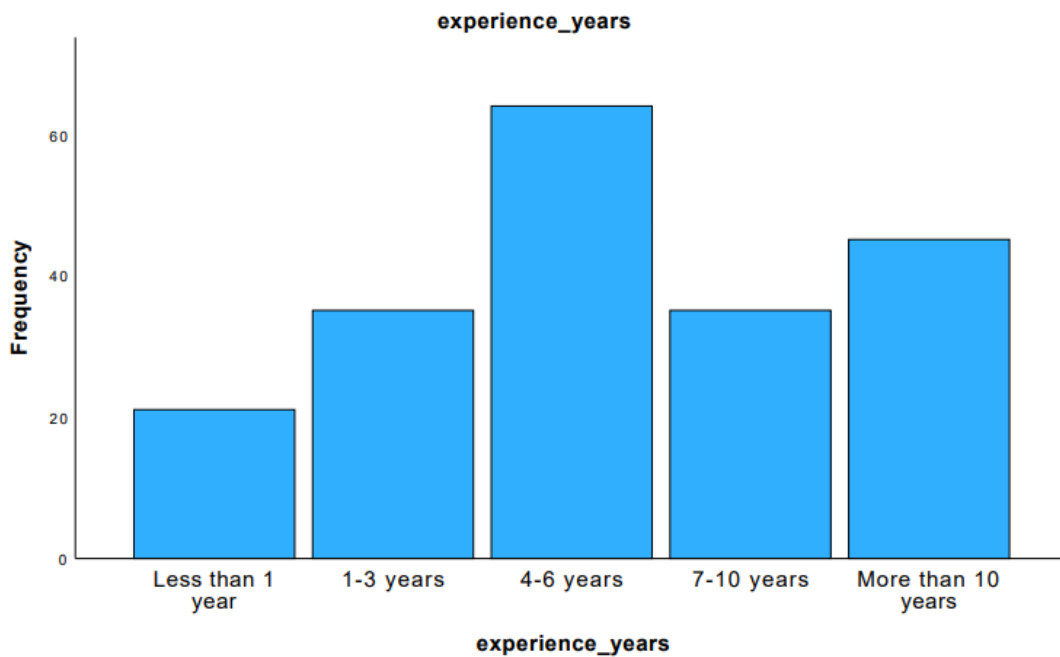


Figure 4. Distribution of Responses on Duration of Agile Methodology Usage

The distribution of experience levels among participants is as follows: individuals with less than one year of experience account for 10.5% (n = 21) of the respondents. Those

with 1-3 years of experience make up 17.5% (n = 35), and the largest group, comprising those with 4-6 years of experience, represents 32.0% (n = 64). Respondents with 7-10 years of experience also make up 17.5% (n = 35), while those with more than 10 years of experience constitute 22.5% (n = 45). This distribution indicates a healthy mix of both relatively newer and more seasoned professionals in the field, reflecting the diverse levels of expertise within the Agile community. The significant number of participants with more than four years of experience underlines the depth of practical knowledge and the established presence of Agile methodologies in the industry.

Primary Agile Framework Used

The respondents reported the use of various Agile frameworks, as shown in Table 5 and Figure 5:

		primary_agile_framework			Cumulative Percent
		Frequency	Percent	Valid Percent	
Valid	Scrum	59	29.5	29.5	29.5
	Kanban	20	10.0	10.0	39.5
	Extreme Programming xp	22	11.0	11.0	50.5
	Lean	65	32.5	32.5	83.0
	Other	34	17.0	17.0	100.0
	Total	200	100.0	100.0	

Table 5. Survey Results on Primary Agile Framework Used

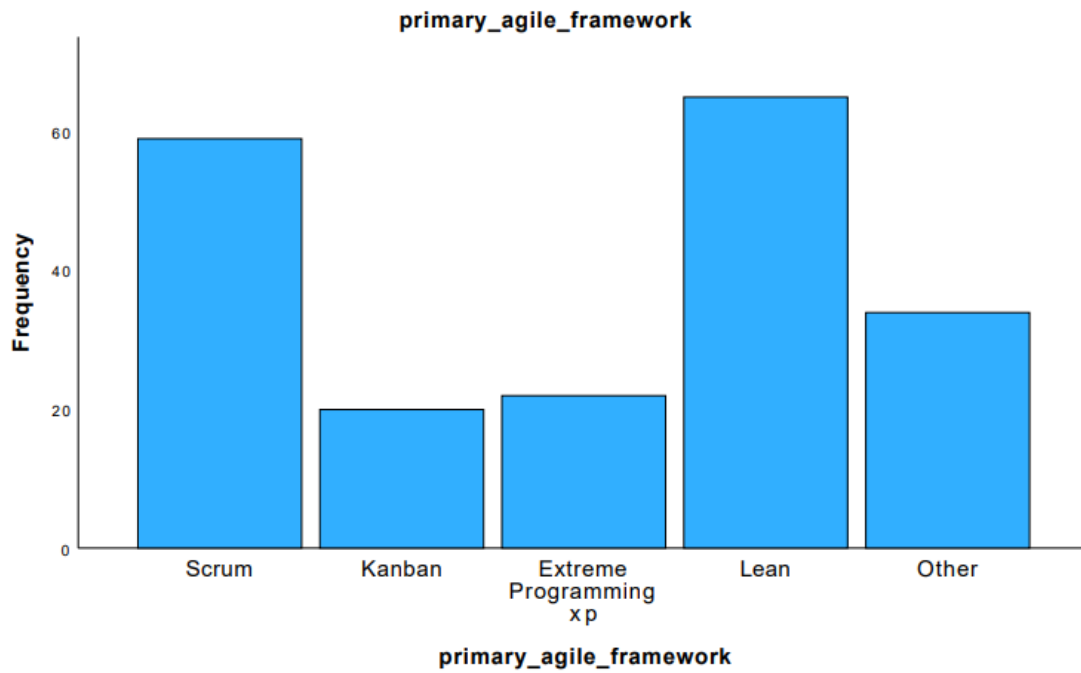


Figure 5. Distribution of Responses on Primary Agile Framework Used

Among the participants, the most commonly used Agile framework is Lean, utilized by 32.5% ($n = 65$) of respondents, reflecting its widespread adoption and versatility in various organizational settings. Scrum follows closely, with 29.5% ($n = 59$) of participants reporting it as their primary Agile framework, indicating its popularity in managing complex software development projects.

Extreme Programming (XP) is employed by 11.0% ($n = 22$) of the sample, showcasing its continued relevance, particularly in enhancing software quality and responsiveness to changing customer requirements. Kanban is used by 10.0% ($n = 20$), favoured for its simplicity and efficiency in workflow management. The 'Other' category, which includes various other Agile methodologies, constitutes 17.0% ($n = 34$) of the responses, highlighting the diversity of Agile practices tailored to specific organizational needs.

Adoption of Agile Testing Practices

The survey explored two key areas of Agile testing practices, Test-Driven Development (TDD) and Continuous Integration (CI), as illustrated in Table 6 and Figure 6:

Regular use of Test-Driven Development (TDD)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	35	17.5	17.5	17.5
	Disagree	31	15.5	15.5	33.0
	Neutral	56	28.0	28.0	61.0
	Agree	54	27.0	27.0	88.0
	Strongly Agree	24	12.0	12.0	100.0
	Total	200	100.0	100.0	

Table 6. Survey Results on Test-Driven Development (TDD) Usage

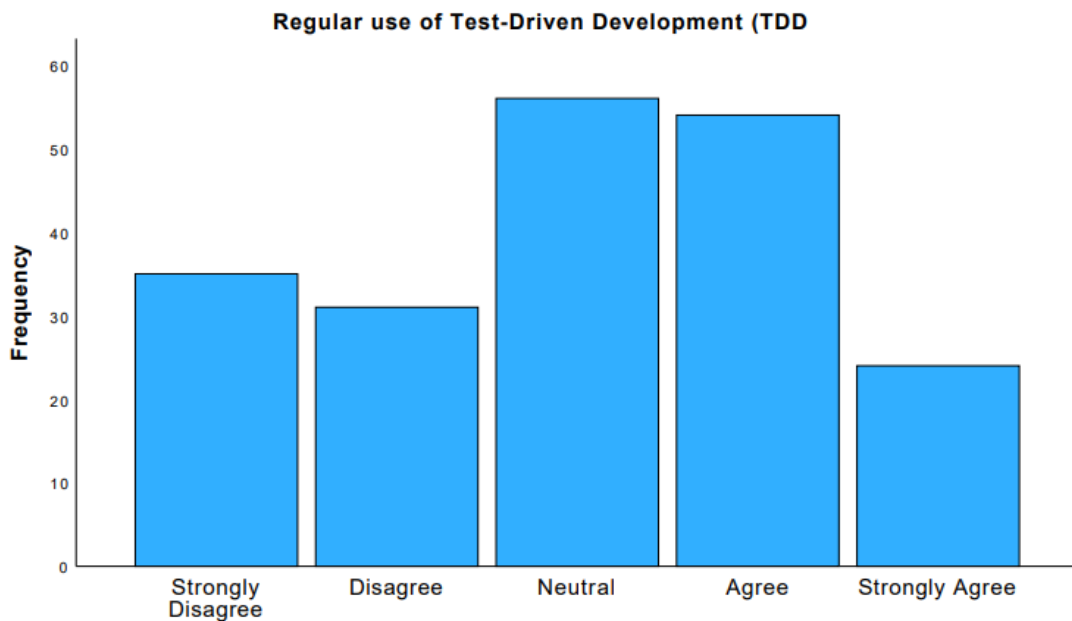


Figure 6. Distribution of Responses on Test-Driven Development (TDD) Usage

The survey results depict a balanced distribution in the adoption of Test-Driven Development among the respondents. Approximately 17.5% of the participants indicated a

low adoption, rarely utilizing TDD in their projects, while 15.5% occasionally employ this methodology. The largest proportion, comprising 28% of the responses, remained neutral. Conversely, a significant 27% of respondents agree they regularly use TDD, and 12% strongly advocate for its consistent application in their projects, highlighting a moderate yet significant embrace of TDD in aligning with contemporary Agile testing practices aimed at enhancing early defect detection and iterative development.

Continuous Integration (CI) Adoption

In the assessment of Continuous Integration (CI) adoption, the distribution of responses highlights diverse levels of agreement among the survey participants. A substantial portion, 28.0% (n = 56), remains neutral, reflecting a significant ambiguity or varied experience with CI practices (refer the Table 7 and Figure 7).

		testing_ci_adoption			Cumulative Percent
		Frequency	Percent	Valid Percent	
Valid	Strongly Disagree	19	9.5	9.5	9.5
	Disagree	32	16.0	16.0	25.5
	Neutral	56	28.0	28.0	53.5
	Agree	55	27.5	27.5	81.0
	Strongly Agree	38	19.0	19.0	100.0
	Total	200	100.0	100.0	

Table 7. Survey Results on Continuous Integration (CI) adoption

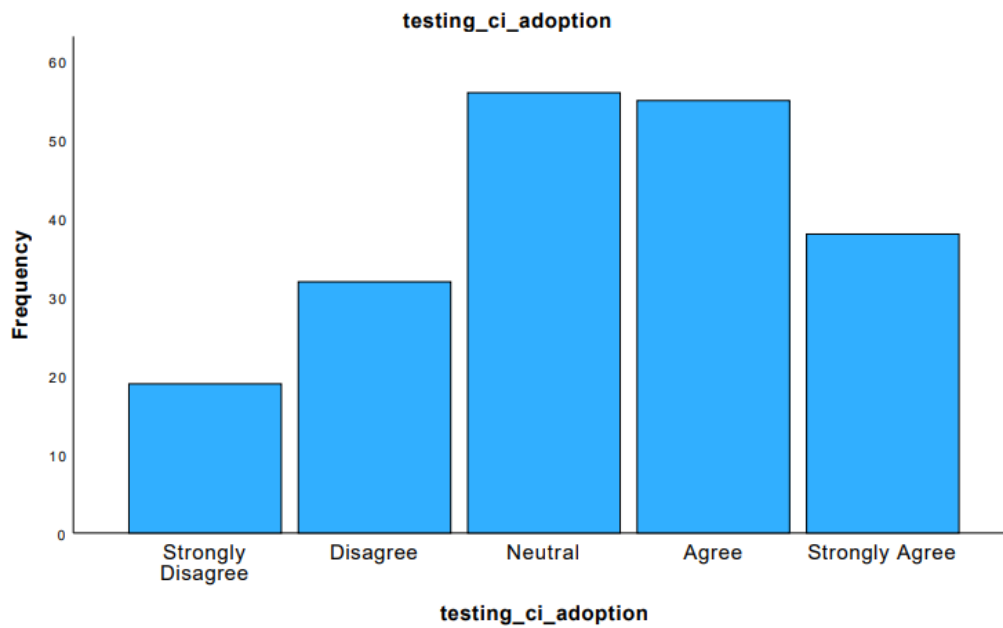


Figure 7. Distribution of Responses on Continuous Integration (CI) adoption

Those in agreement that CI is beneficial comprise 46.5% of the sample, split between 27.5% (n = 55) agreeing and 19.0% (n = 38) strongly agreeing, indicating a robust recognition of CI's importance in enhancing development workflows. Conversely, 25.5% of respondents express reservations about CI, with 16.0% (n = 32) disagreeing and 9.5% (n = 19) strongly disagreeing, which might suggest challenges in implementation or a lack of perceived benefits from CI within their organizational context. This distribution underscores the mixed perceptions of CI's effectiveness in the Agile methodologies adopted by the surveyed organizations.

Behavior-Driven Development (BDD) Usage

The survey results indicate a notable variance in the adoption levels of Behavior-Driven Development among participants. A moderate number of respondents, amounting to 26.5% and 34.5%, respectively, indicated lower engagement levels (rated as 1 and 2 on the scale), suggesting a hesitancy or contextual inapplicability in their current workflows (refer the Table 8 and Figure 8).

testing_bdd_practice					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	53	26.5	26.5	26.5
	Disagree	69	34.5	34.5	61.0
	Neutral	45	22.5	22.5	83.5
	Agree	21	10.5	10.5	94.0
	Strongly Agree	12	6.0	6.0	100.0
	Total	200	100.0	100.0	

Table 8. Survey Results on Adoption levels of Behavior-Driven Development among participants

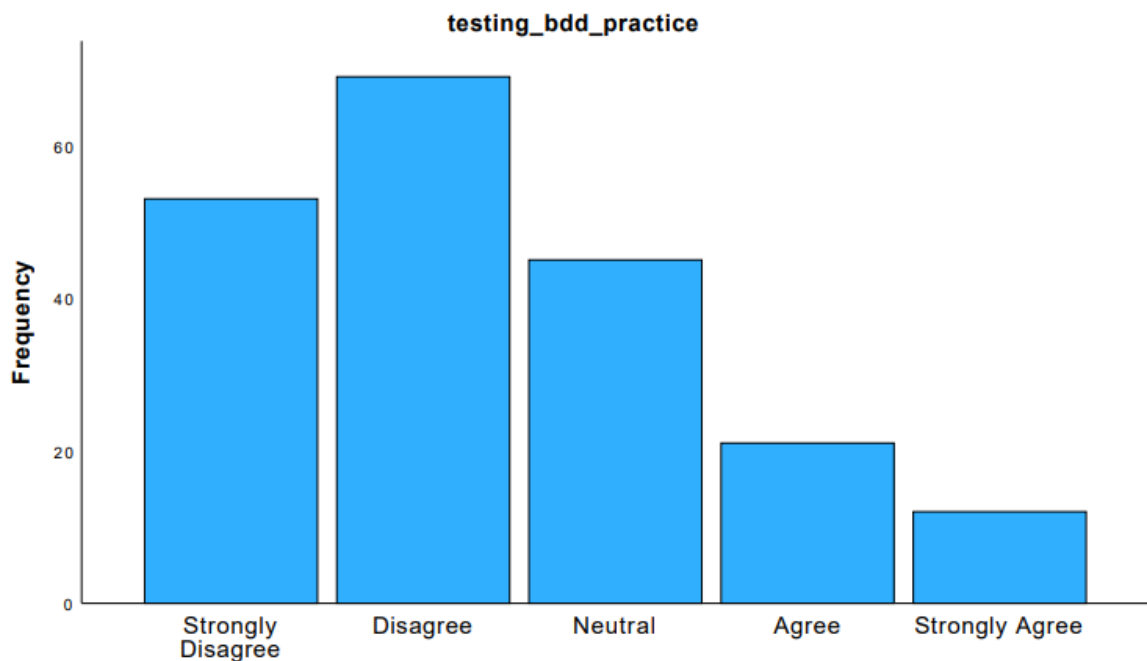


Figure 8. Distribution of Adoption levels of Behavior-Driven Development among participants

Conversely, the largest group, representing 22.5% of participants, strongly embraced BDD practices (rated as 4), aligning with Agile's emphasis on communication and collaboration between developers, testers, and business professionals.

The adoption of BDD, as depicted in the graphical representation, underscores a significant inclination towards integrating these practices in varying degrees within the respondents' organizations (refer the Table 8 and Figure 8).

Agile Automation Extent

The extent of automation in Agile practices presented a more definitive trend, with the majority of respondents reporting moderate to high levels of automation. Specifically, 29.5% of the participants indicated a high adoption rate (rated as 4), while 34.0% confirmed they are fully integrating automation into their Agile processes (rated as 5). These findings suggest a robust inclination towards leveraging automation tools to enhance efficiency and effectiveness in Agile environments, reflecting the increasing reliance on technology to streamline development processes and improve delivery speeds (refer the Table 9 and Figure 9).

		testing_automation_extent			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	25	12.5	12.5	12.5
	Disagree	27	13.5	13.5	26.0
	Neutral	59	29.5	29.5	55.5
	Agree	68	34.0	34.0	89.5
	Strongly Agree	21	10.5	10.5	100.0
	Total	200	100.0	100.0	

Table 9. Survey Results on the extent of automation in Agile practices

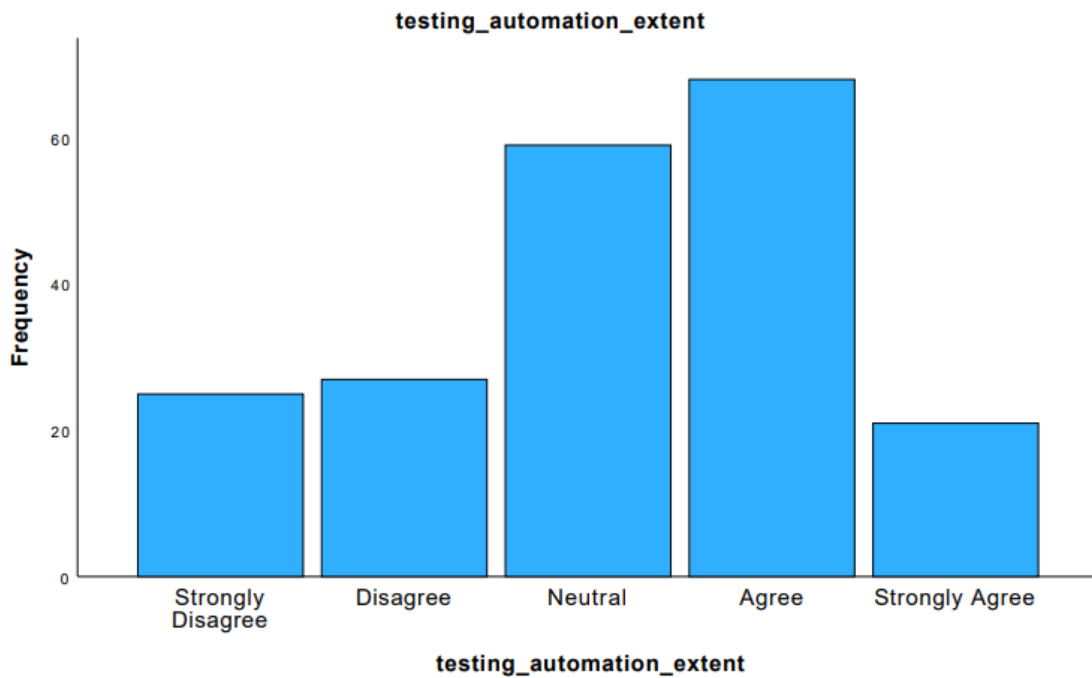


Figure 9. Distribution of the extent of automation in Agile practices

Agile Practice Review Frequency

The frequency of Agile practice reviews among the surveyed professionals showed a balanced distribution across the scale. While a smaller segment of respondents (9.0%) indicated infrequent reviews, a significant proportion highlighted more regular review practices, with 26.5% conducting reviews frequently (rated as 4) and another 26.5% ensuring reviews are a consistent part of their Agile routine (rated as 5). This distribution points to a conscious effort within organizations to continually assess and refine their Agile methodologies, ensuring they remain dynamic and responsive to project needs and outcomes (refer the Table 10 and Figure 10).

testing_practice_review					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	18	9.0	9.0	9.0
	Disagree	16	8.0	8.0	17.0
	Neutral	62	31.0	31.0	48.0
	Agree	51	25.5	25.5	73.5
	Strongly Agree	53	26.5	26.5	100.0
	Total	200	100.0	100.0	

Table 10. Survey Results on the frequency of Agile practice reviews

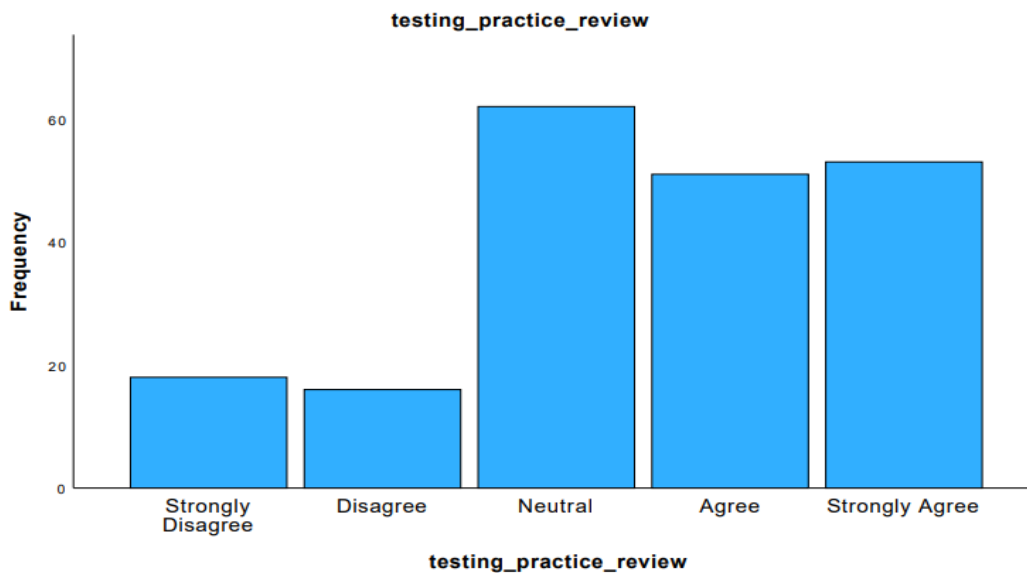


Figure 10. Distribution of the frequency of Agile practice reviews

4.2 Customer Collaboration in Agile

Exploring the principle of customer collaboration, a core tenant of Agile methodologies, the responses reflected a strong adherence to this practice. The majority of respondents (36.0%) frequently prioritize customer collaboration over rigid contract adherence (rated as 4), and a substantial 16.0% view it as an integral and unwavering practice (refer the Table 11 and Figure 11).

principles_customer_collaboration

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	17	8.5	8.5	8.5
	Disagree	24	12.0	12.0	20.5
	Neutral	55	27.5	27.5	48.0
	Agree	72	36.0	36.0	84.0
	Strongly Agree	32	16.0	16.0	100.0
	Total		200	100.0	100.0

Table 11. Survey Results on the Exploring the principle of customer collaboration

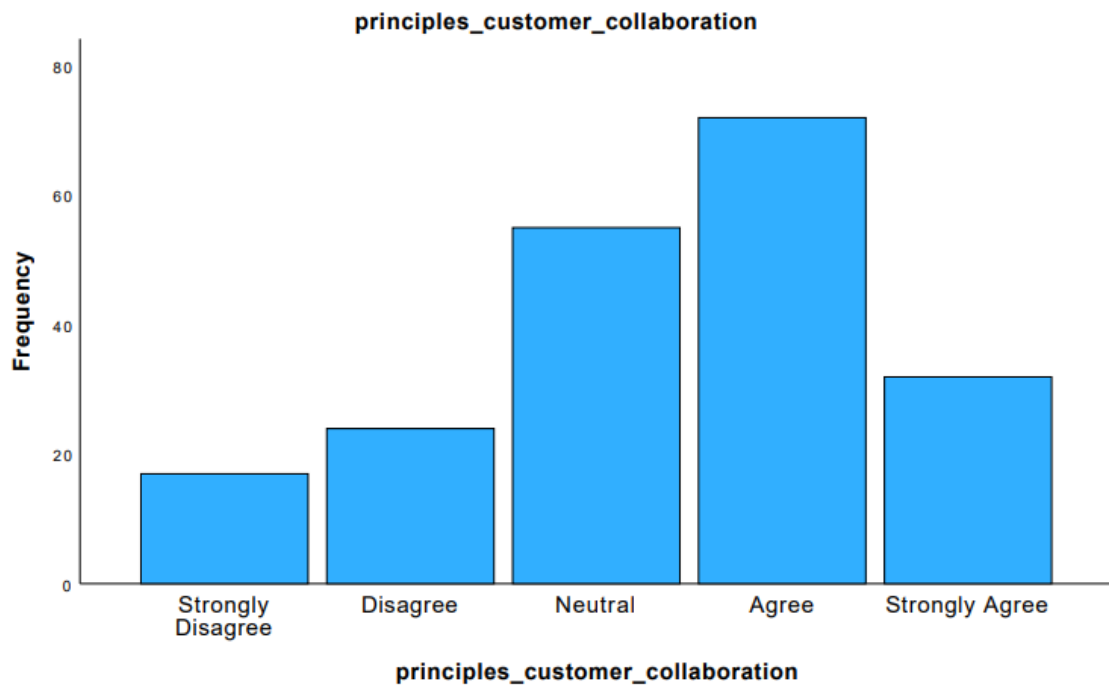


Figure 11. Distribution of the Exploring the principle of customer collaboration

Iterative Feedback in Agile

Regarding the adoption of iterative feedback principles, the distribution of responses provides a detailed insight into its acceptance among the participants. A significant proportion, 33.5% (n = 67), remain neutral, suggesting varying degrees of familiarity or implementation success with these principles within their workflows.

Those affirming the value of iterative feedback, summing up to 37% of the sample, are divided between those who agree (27.5%, n = 55) and strongly agree (9.5%, n = 19), reflecting a positive acknowledgment of its benefits in enhancing Agile practices (refer the Table 12 and Figure 12).

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	25	12.5	12.5	12.5
	Disagree	34	17.0	17.0	29.5
	Neutral	67	33.5	33.5	63.0
	Agree	55	27.5	27.5	90.5
	Strongly Agree	19	9.5	9.5	100.0
Total		200	100.0	100.0	

Table 12. Survey Results on the adoption of iterative feedback principles

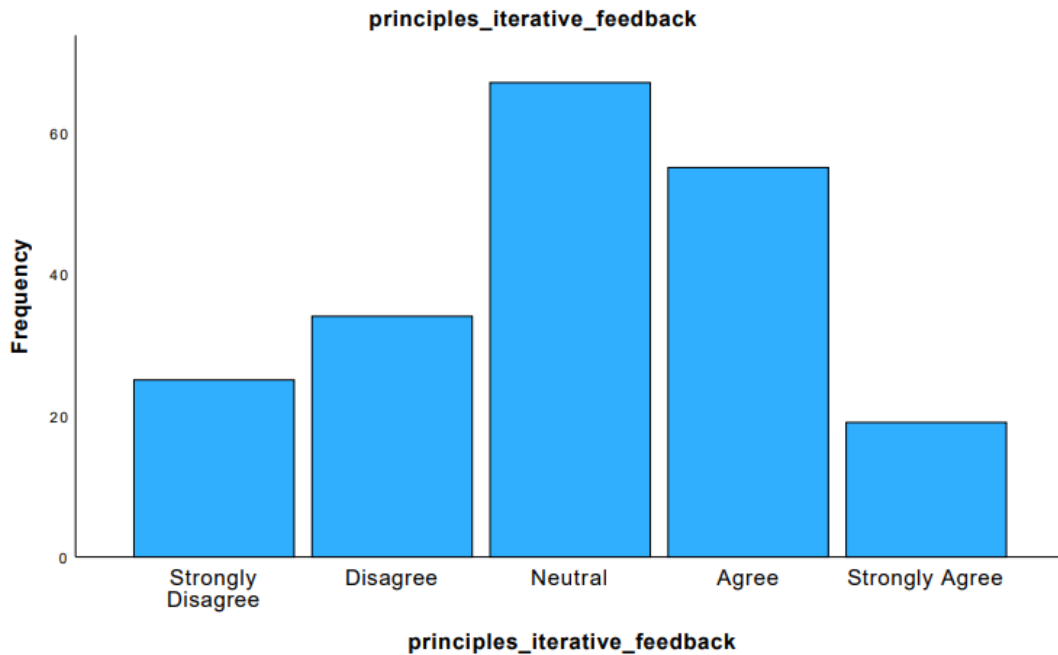


Figure 12. Distribution of the adoption of iterative feedback principles

A combined 29.5% of the respondents express reservations, with 17% (n = 34) disagreeing and 12.5% (n = 25) strongly disagreeing, possibly indicating challenges or skepticism regarding the effectiveness of iterative feedback in their specific Agile environments. This spread underscores the varied perceptions and potential hurdles in fully leveraging iterative feedback principles in Agile methodologies.

Adoption of team self-organization principles

The data on the adoption of team self-organization principles reveals a diverse spectrum of acceptance and application among the participants. A plurality, 37.5% (n = 75), remains neutral, indicating either ambiguity in their perception or a balanced view on the effectiveness of self-organization within Agile frameworks. Meanwhile, 38% collectively acknowledge the value of self-organization, with 27.5% (n = 55) agreeing and 10.5% (n = 21) strongly agreeing that self-organization is a beneficial practice in Agile environments. On the other hand, 24.5% of respondents express dissent, with 12% (n = 24) disagreeing and an equal 12.5% (n = 25) strongly disagreeing, suggesting scepticism or challenges in implementing or observing positive outcomes from self-organizing teams. This distribution underscores the complex dynamics and varied effectiveness of self-organization principles across different organizational contexts as illustrated in Table 13 and Figure 13.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	25	12.5	12.5	12.5
	Disagree	24	12.0	12.0	24.5
	Neutral	75	37.5	37.5	62.0
	Agree	55	27.5	27.5	89.5
	Strongly Agree	21	10.5	10.5	100.0
	Total	200	100.0	100.0	

Table 13. Survey Results on data on the adoption of team self-organization principles

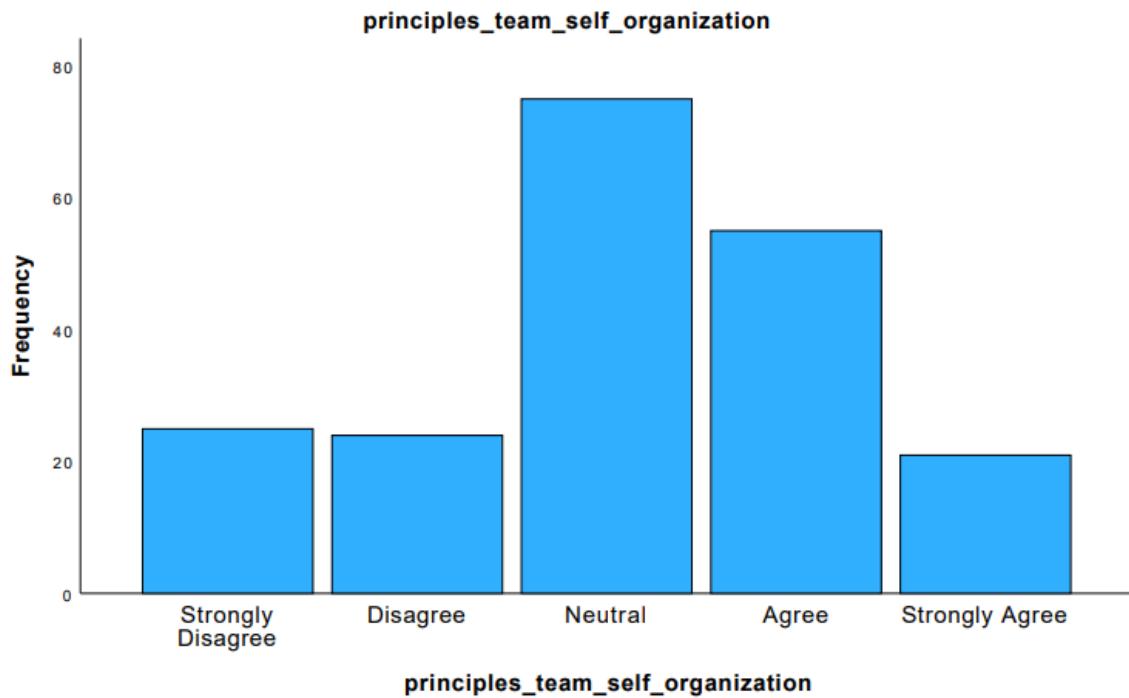


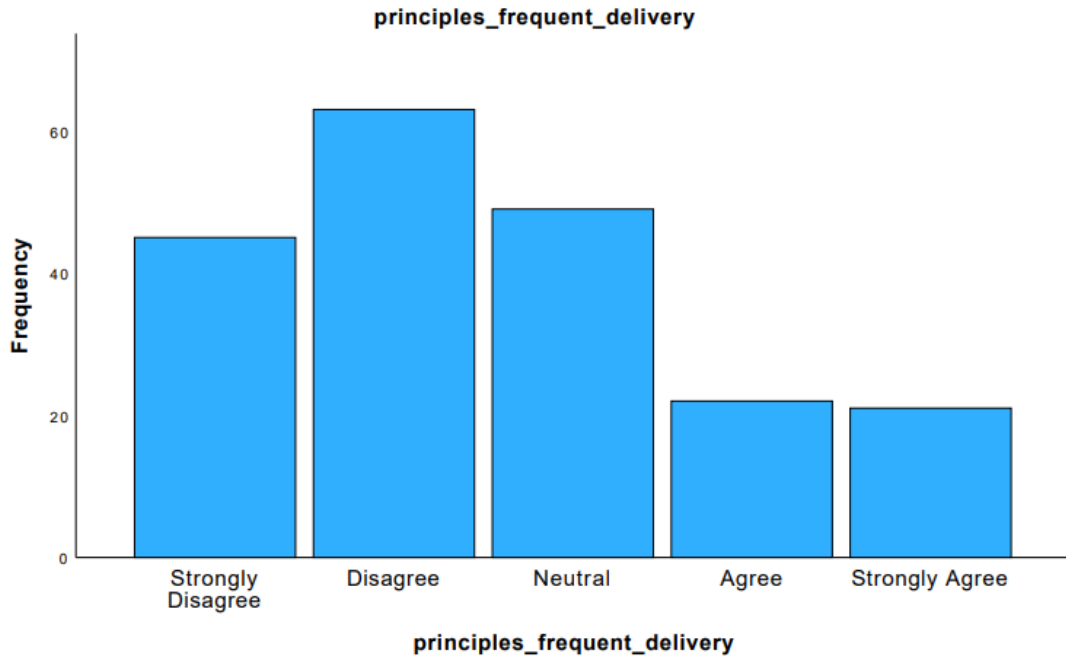
Figure 13. Distribution of the team self-organization principles

Commitment to frequent delivery

Data shows a commitment to frequent delivery, with 31.5% of respondents frequently adhering (level 3) to this practice, while 24.5% strongly agree (level 5) that they consistently practice frequent deliveries as illustrated in Table 14 and Figure 14.

principles_frequent_delivery

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	45	22.5	22.5	22.5
	Disagree	63	31.5	31.5	54.0
	Neutral	49	24.5	24.5	78.5
	Agree	22	11.0	11.0	89.5
	Strongly Agree	21	10.5	10.5	100.0
	Total	200	100.0	100.0	

Table 14. Survey Results on data on the adoption of team self-organization principles**Figure 14.** Distribution of adoption of team self-organization principles

This aligns with Agile's emphasis on short, iterative cycles that aim to provide continuous progress and regular feedback loops.

Continuous Improvement

Continuous improvement is a critical aspect of Agile that sees substantial adherence, with 27% of participants strongly agreeing (level 5) that they continuously seek ways to improve their processes, complemented by another 26% who agree (level 4). This suggests that continuous improvement is embedded in the culture of most Agile teams, ensuring ongoing enhancements in process and product quality as illustrated in Table 15 and Figure 15.

principles_continuous_improvement

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	31	15.5	15.5	15.5
	Disagree	39	19.5	19.5	35.0
	Neutral	52	26.0	26.0	61.0
	Agree	54	27.0	27.0	88.0
	Strongly Agree	24	12.0	12.0	100.0
	Total	200	100.0	100.0	

Table 15. Survey Results on Continuous improvement

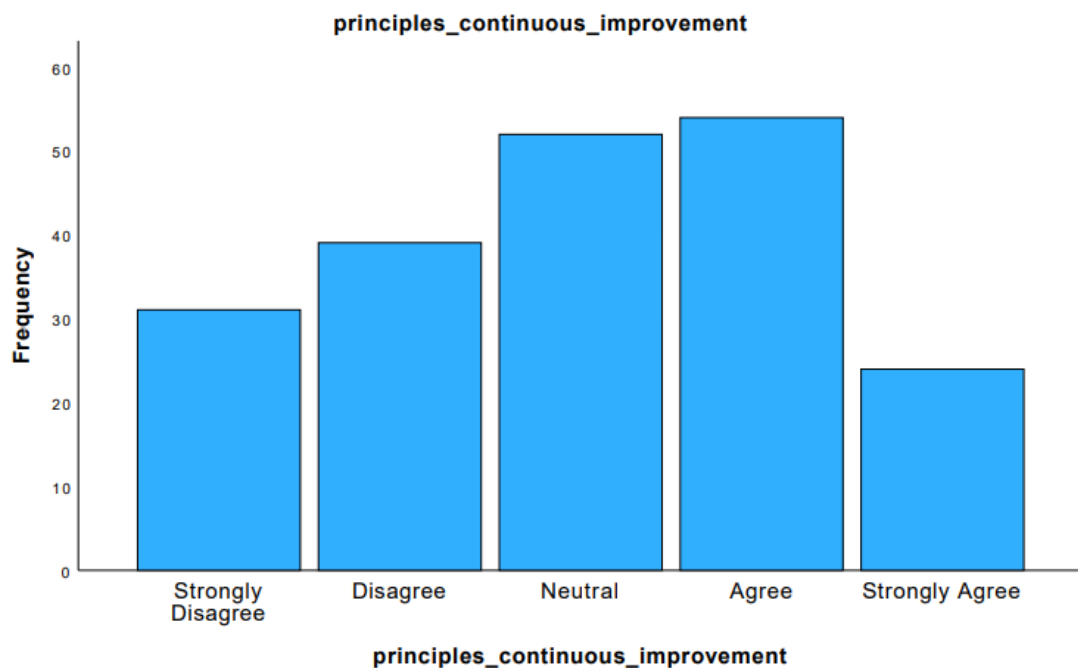


Figure 15. Distribution of Continuous improvement

Biweekly Iteration Frequency

Regarding the frequency of biweekly iterations, the responses suggest a balanced implementation, with 25.5% of respondents frequently conducting iterations biweekly (level 3) and 29.5% agreeing (level 4) they maintain this schedule consistently. Regular iterations are fundamental to Agile's framework, supporting rapid adaptability and iterative feedback as illustrated in Table 16 and Figure 16.

iterations_frequency_biweekly

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	31	15.5	15.5	15.5
	Disagree	29	14.5	14.5	30.0
	Neutral	51	25.5	25.5	55.5
	Agree	59	29.5	29.5	85.0
	Strongly Agree	30	15.0	15.0	100.0
	Total		200	100.0	100.0

Table 16. Survey Results on frequency of biweekly iterations

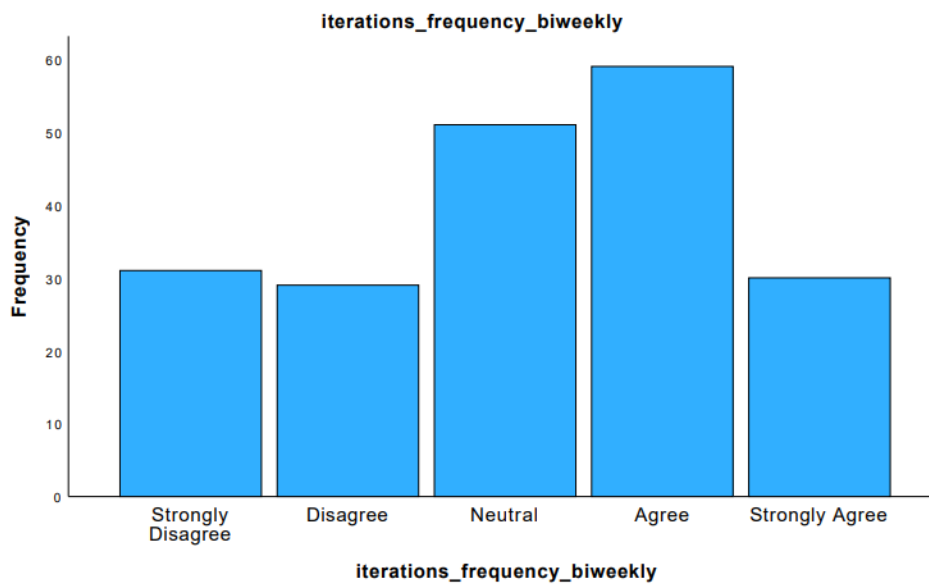


Figure 16. Distribution of frequency of biweekly iterations

Regarding the frequency of biweekly iterations, the responses suggest a balanced implementation, with 25.5% of respondents frequently conducting iterations biweekly (level 3) and 29.5% agreeing (level 4) they maintain this schedule consistently. Regular

iterations are fundamental to Agile's framework, supporting rapid adaptability and iterative feedback.

These findings highlight a robust alignment with Agile principles among the surveyed professionals. The use of a Likert scale ranging from "Strongly Disagree" to "Strongly Agree" effectively captures the extent of agreement and the practical application of these principles, demonstrating a strong commitment to the Agile methodologies across various organizational contexts.

The frequency of scope adjustment within Agile iterations

The frequency of scope adjustment within Agile iterations shows a notable inclination towards flexibility, a cornerstone of Agile methodologies as illustrated in Table 17 and Figure 17.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	39	19.5	19.5	19.5
	Disagree	48	24.0	24.0	43.5
	Neutral	55	27.5	27.5	71.0
	Agree	31	15.5	15.5	86.5
	Strongly Agree	27	13.5	13.5	100.0
	Total	200	100.0	100.0	

Table 17. Survey Results on scope adjustment within Agile iterations

A significant 27.5% of respondents often adjust their project scopes, indicating a moderate embrace of Agile's adaptability. However, only 15.5% engage in continuous scope adjustment, suggesting some resistance or challenges in fully integrating this practice.

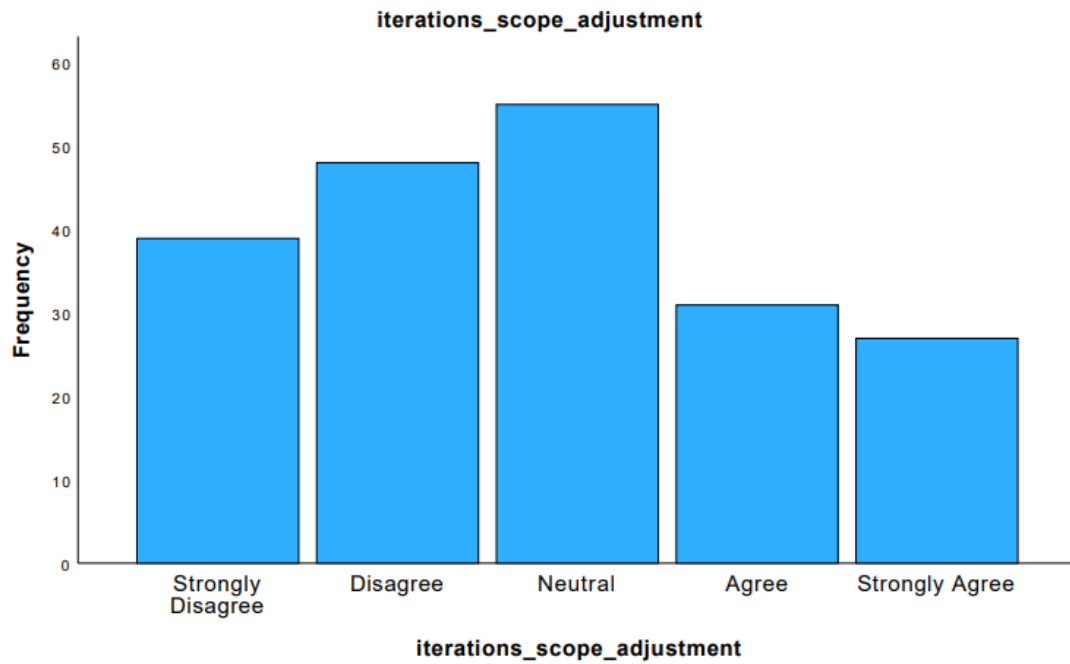


Figure 17. Distribution of adjustment within Agile iterations

Planning Meetings in Agile Iterations

Planning meetings, essential for effective sprint management, appear to be well-integrated, with 31.0% of professionals frequently conducting these sessions. Interestingly, a higher 27.5% do so occasionally, reflecting varied implementation levels across organizations. Only 6.5% reported continuous planning meetings, underscoring potential areas for enhancing regular communication and coordination (Table 18 and Figure 18 present the relevant data).

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	41	20.5	20.5	20.5
	Disagree	55	27.5	27.5	48.0
	Neutral	62	31.0	31.0	79.0
	Agree	29	14.5	14.5	93.5
	Strongly Agree	13	6.5	6.5	100.0
Total		200	100.0	100.0	

Table 18. Survey Results on planning meetings in Agile Iterations

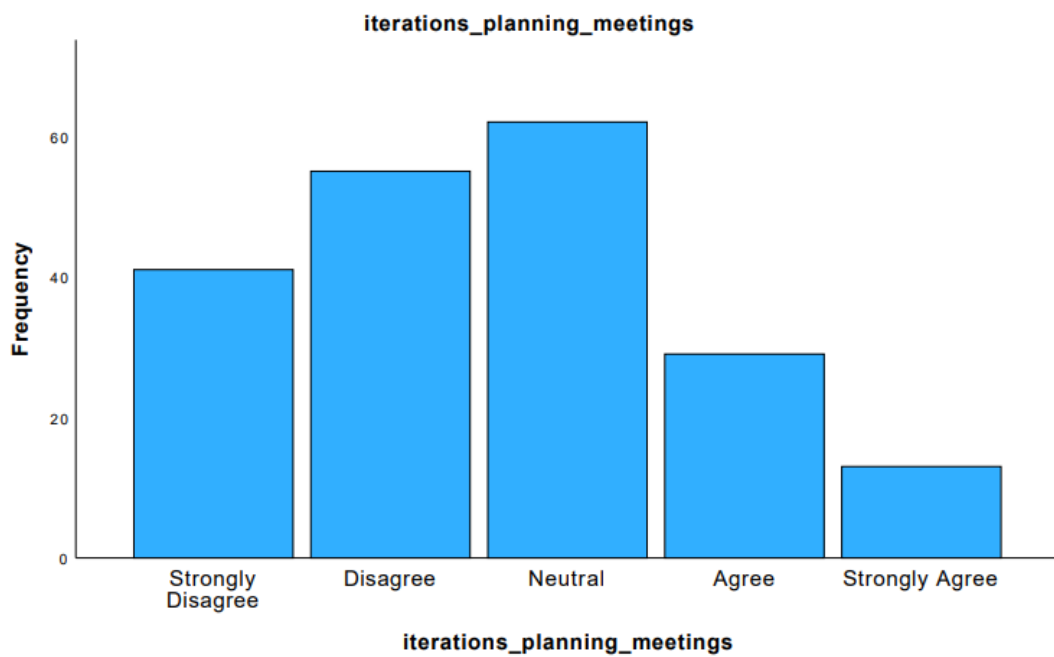


Figure 18. Distribution of planning meetings in Agile Iterations

Conduct of Retrospectives in Agile Iterations

Retrospectives are critical for continuous improvement, yet the data shows a diverse approach to their conduct as illustrated in Table 19 and Figure 19.

iterations_retrospective_conduct					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	62	31.0	31.0	31.0
	Disagree	56	28.0	28.0	59.0
	Neutral	41	20.5	20.5	79.5
	Agree	24	12.0	12.0	91.5
	Strongly Agree	17	8.5	8.5	100.0
	Total	200	100.0	100.0	

Table 19. Survey Results on continuous improvement retrospectives

Retrospectives are critical for continuous improvement, yet the data shows a diverse approach to their conduct. While 33.5% of respondents conduct retrospectives often, suggesting a proactive approach to learning and development, 20% rarely engage in this practice, which may indicate missed opportunities for feedback and improvement.

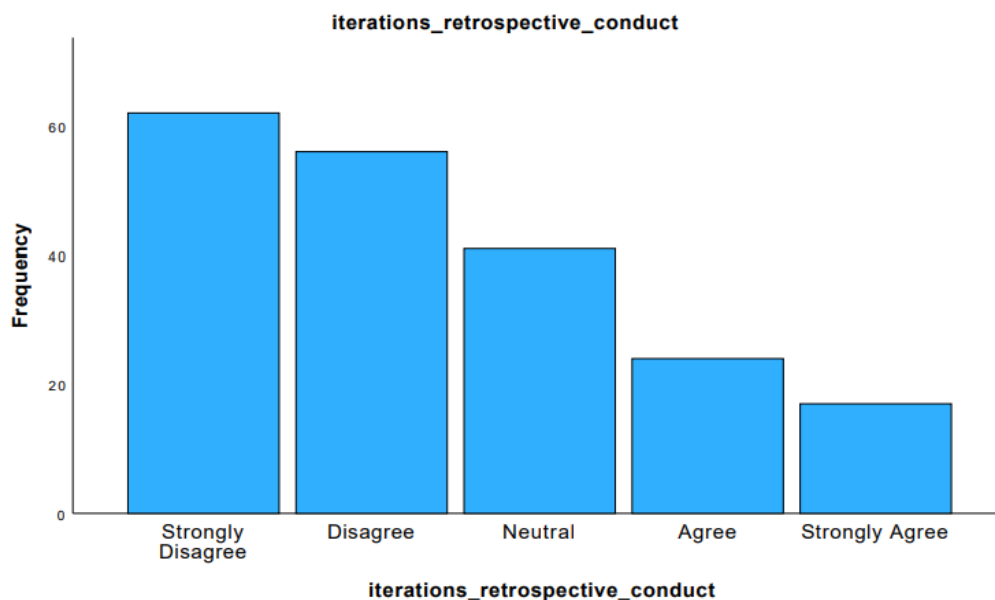


Figure 19. Distribution of continuous improvement retrospectives

Defect Identification in Agile Iterations

The distribution of responses regarding the effectiveness of Agile iterations in defect identification shows mixed opinions among participants (Table 20 and Figure 20 present the relevant data).

iterations_defect_identification					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	19	9.5	9.5	9.5
	Disagree	32	16.0	16.0	25.5
	Neutral	54	27.0	27.0	52.5
	Agree	50	25.0	25.0	77.5
	Strongly Agree	45	22.5	22.5	100.0
	Total	200	100.0	100.0	

Table 20. Survey Results on the effectiveness of agile iterations in defect identification



Figure 20. Distribution of the effectiveness of agile iterations in defect identification

The distribution of responses regarding the effectiveness of Agile iterations in defect identification shows mixed opinions among participants. While a significant 47.5% collectively acknowledge the positive impact, with 25% (n = 50) agreeing and 22.5% (n = 45) strongly agreeing that frequent Agile iterations aid in defect identification, a considerable proportion remains skeptical or indifferent. Specifically, 25.5% (n = 51) of respondents disagree (16%, n = 32) or strongly disagree (9.5%, n = 19) with the assertion that iterations facilitate better defect spotting. Additionally, 27% (n = 54) of the participants remain neutral, reflecting either uncertainty about the effectiveness of Agile iterations in identifying defects or balanced views based on their experiences. This range of responses highlights the variability in how Agile practices are perceived or implemented across different projects or organizational environments.

4.3 Test Automation Coverage

The response distribution to the survey on "automation test coverage" reveals varied perspectives among participants. A substantial 55.5% expressed reservations about the effectiveness of automated testing, with 31.0% (n = 62) disagreeing and 24.5% (n = 49) strongly disagreeing with its efficacy as illustrated in Table 21 and Figure 21.

automation_test_coverage					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	49	24.5	24.5	24.5
	Disagree	62	31.0	31.0	55.5
	Neutral	48	24.0	24.0	79.5
	Agree	22	11.0	11.0	90.5
	Strongly Agree	19	9.5	9.5	100.0
Total		200	100.0	100.0	

Table 21. Survey results on automation test coverage

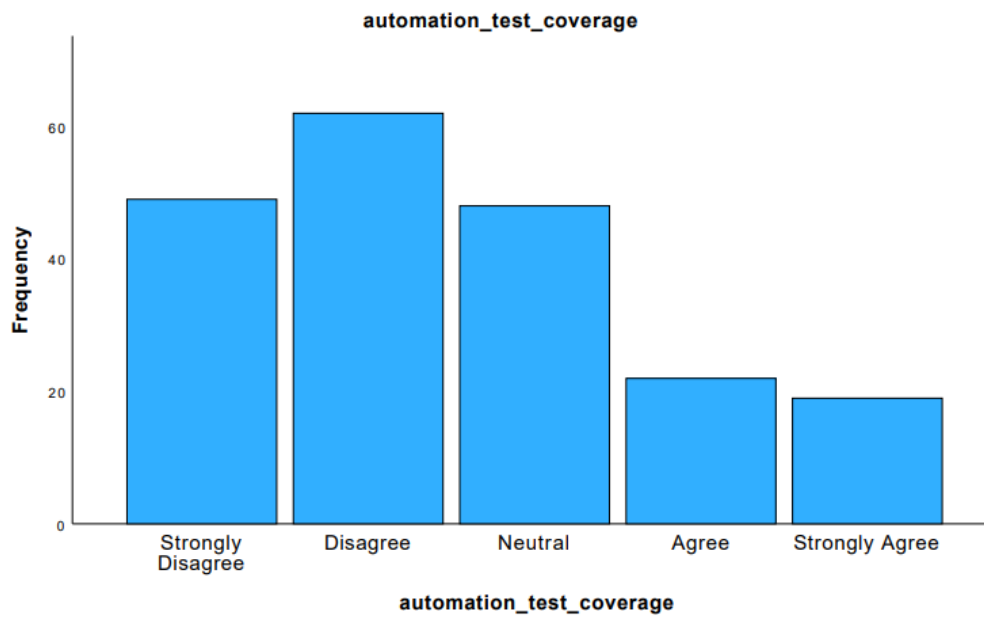


Figure 21. Distribution of continuous automation test coverage

In contrast, a smaller segment of the sample, totaling 20.5%, recognizes the benefits, where 11.0% ($n = 22$) agree and 9.5% ($n = 19$) strongly agree that automation contributes positively to test coverage. Moreover, 24.0% ($n = 48$) of the respondents remained neutral, suggesting either ambivalence towards or insufficient experience with automated testing to form a definitive opinion.

These findings indicate a polarization in perceptions regarding the role of automation in enhancing testing efficiency within Agile frameworks, pointing to potential areas for improvement in its application or perception across the industry.

CI Integration in Test Automation

The survey data on "automation ci integration" demonstrates a significant divergence in perceptions among respondents as illustrated in Table 22 and Figure 22.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	48	24.0	24.0	24.0
	Disagree	66	33.0	33.0	57.0
	Neutral	48	24.0	24.0	81.0
	Agree	21	10.5	10.5	91.5
	Strongly Agree	17	8.5	8.5	100.0
Total		200	100.0	100.0	

Table 22. Survey results automation CI integration

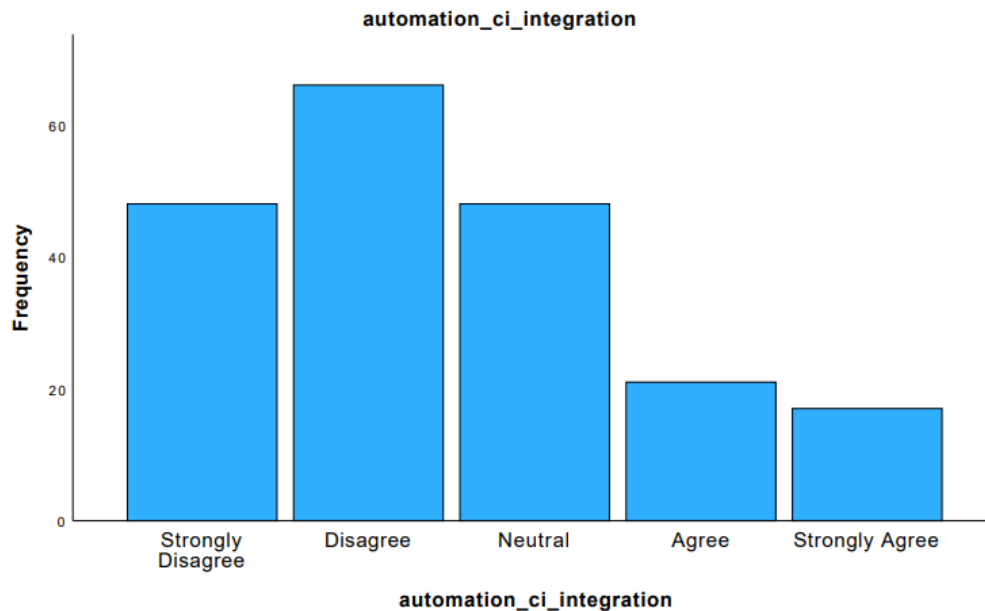


Figure 22. Distribution of automation CI integration

A majority, 57.0%, do not view the integration of continuous integration (CI) automation favorably, with 33.0% (n = 66) disagreeing and 24.0% (n = 48) strongly disagreeing with its effectiveness.

Moreover, 19.0% acknowledge its positive impact, with 10.5% (n = 21) agreeing and 8.5% (n = 17) strongly agreeing that CI automation effectively integrates into their workflows. Neutral responses were given by 24.0% (n = 48) of participants, indicating either ambivalence towards the impact of CI automation or a lack of sufficient interaction to form a strong opinion.

These insights highlight a notable scepticism about the effectiveness of CI automation in enhancing development processes, suggesting areas where further education or improved implementation strategies might bridge the gap in acceptance and efficacy.

Effectiveness of maintenance and updates in test Automation

Responses to the survey on "automation maintenance update" show a range of opinions about the effectiveness of maintaining and updating automation systems in Agile environments (refer to Table 23 and Figure 23).

automation_maintenance_update					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	41	20.5	20.5	20.5
	Disagree	61	30.5	30.5	51.0
	Neutral	45	22.5	22.5	73.5
	Agree	32	16.0	16.0	89.5
	Strongly Agree	21	10.5	10.5	100.0
Total		200	100.0	100.0	

Table 23. Survey results in automation effectiveness of CI integration

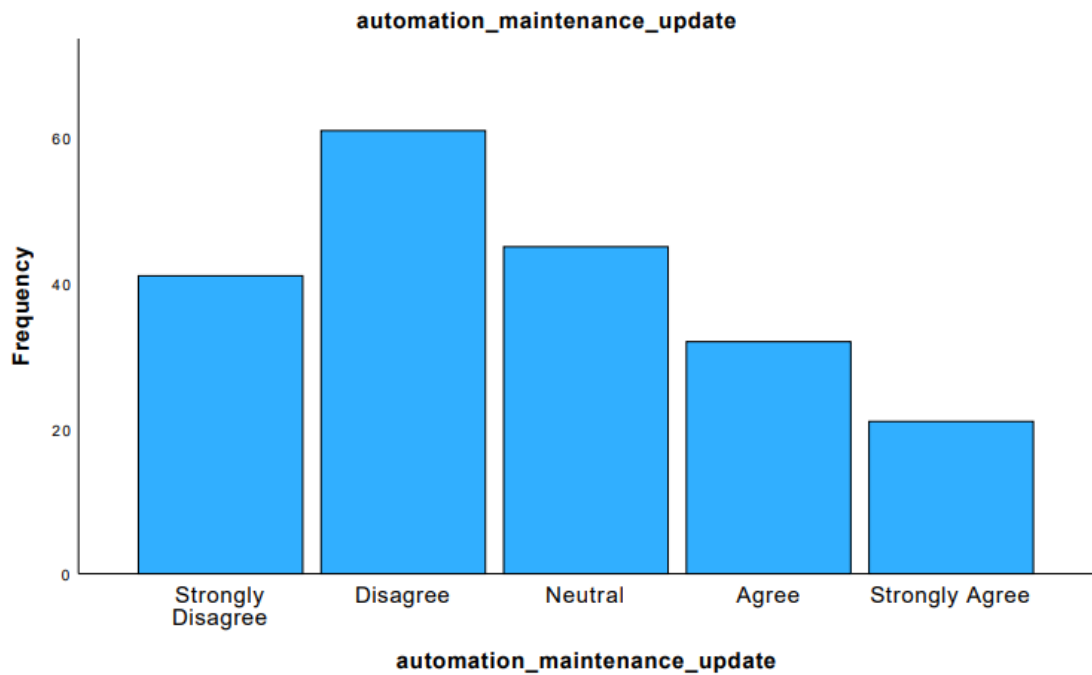


Figure 23. Distribution of automation effectiveness of CI integration

More than half of the respondents expressed scepticism, with 30.5% (n = 61) disagreeing and 20.5% (n = 41) strongly disagreeing about the efficacy of these updates. However, 26.5% of participants recognized their value, with 16.0% (n = 32) agreeing and 10.5% (n = 21) strongly agreeing that maintenance updates enhance automation processes. Additionally, 22.5% (n = 45) of respondents remained neutral, suggesting either varied experiences or insufficient information to decide. This distribution indicates a critical need for clearer communication and perhaps more consistent results from automation updates to increase confidence among Agile practitioners.

Effort Reduction through Test Automation

The survey responses on the impact of automation in reducing manual effort in Agile settings show mixed feelings among participants as shown in Table 24 and Figure 24.

automation_manual_effort_reduction

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	21	10.5	10.5	10.5
	Disagree	70	35.0	35.0	45.5
	Neutral	48	24.0	24.0	69.5
	Agree	34	17.0	17.0	86.5
	Strongly Agree	27	13.5	13.5	100.0
	Total	200	100.0	100.0	

Table 24. Survey results impact of automation in reducing manual effort in Agile settings

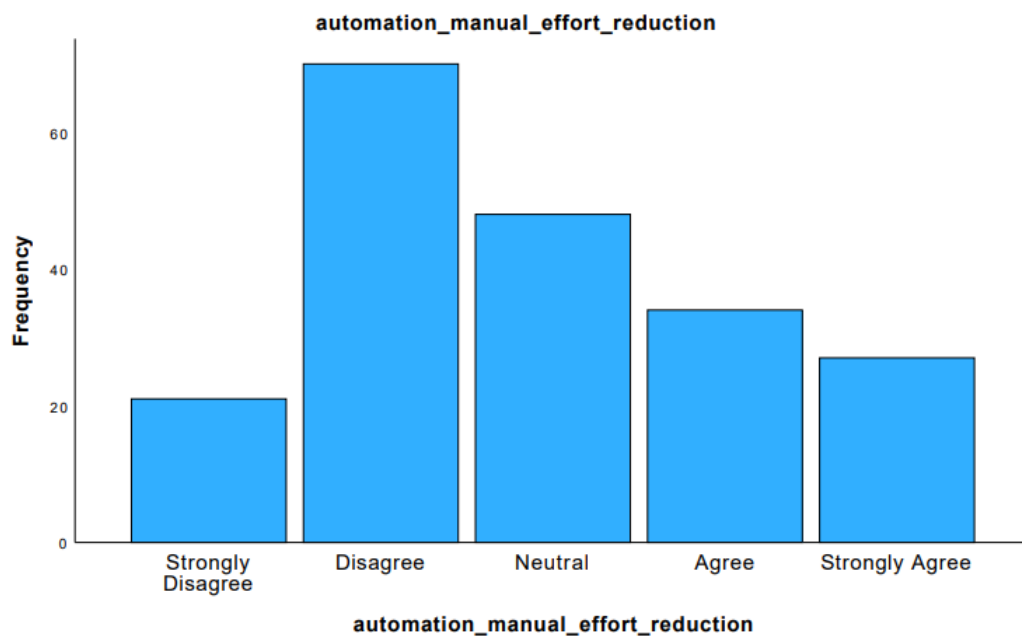


Figure 24. Distribution of impact of automation in reducing manual effort in Agile settings

A significant portion, 45.5%, expressed skepticism about automation's effectiveness in lessening manual labor, with 35.0% (n = 70) disagreeing and 10.5% (n = 21) strongly disagreeing. On the other side, 30.5% acknowledged a positive impact, with 17.0% (n = 34) agreeing and 13.5% (n = 27) strongly agreeing that automation effectively reduces manual efforts. Neutral responses were given by 24.0% (n = 48) of the participants, indicating ambivalence or varied experiences with automation's role in reducing manual workload. These results highlight the ongoing debate and varying experiences regarding the efficiency gains attributed to automation in Agile practices.

Test Automation Monitoring and Effectiveness

The distribution of responses on the effectiveness of monitoring automation in Agile methodologies reveals a broad spectrum of opinions (see Table 25 and Figure 25).

automation_effectiveness_monitoring					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	10	5.0	5.0	5.0
	Disagree	61	30.5	30.5	35.5
	Neutral	69	34.5	34.5	70.0
	Agree	35	17.5	17.5	87.5
	Strongly Agree	25	12.5	12.5	100.0
	Total	200	100.0	100.0	

Table 25. Survey results impact of responses on the effectiveness of monitoring automation

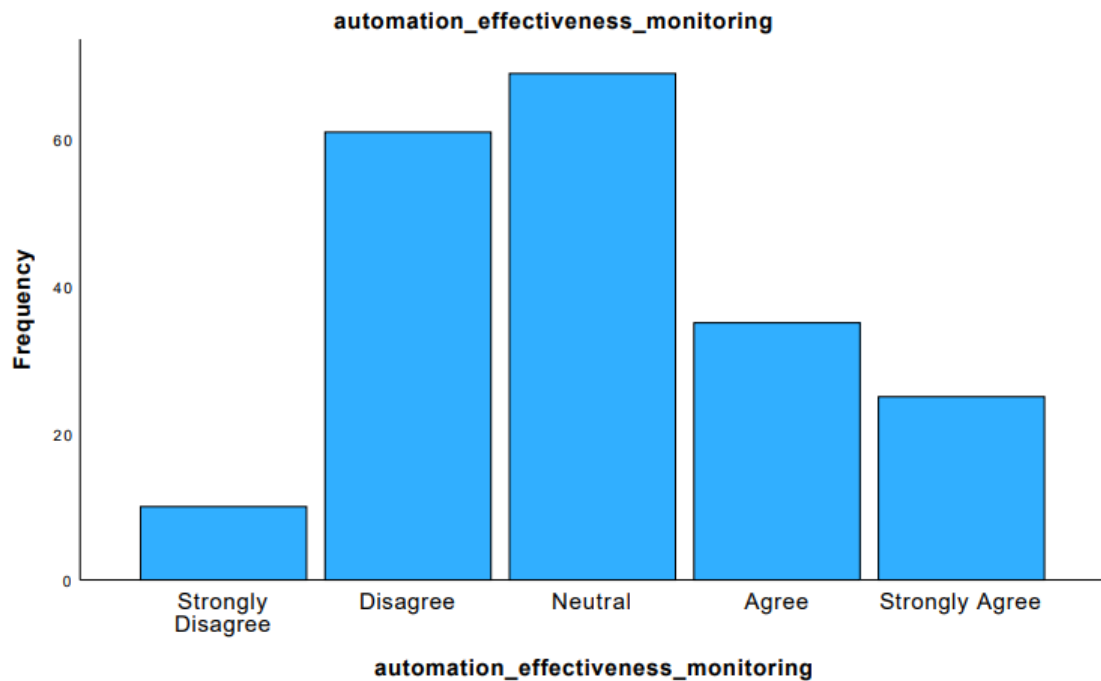


Figure 25. Distribution of impact of responses on the effectiveness of monitoring automation

The data shows that a considerable percentage of participants, 35.5%, hold reservations about the efficacy of automation monitoring, with 30.5% (n = 61) disagreeing and 5.0% (n = 10) strongly disagreeing. Conversely, a combined 30.0% see positive benefits, with 17.5% (n = 35) agreeing and 12.5% (n = 25) strongly agreeing that automation significantly enhances monitoring effectiveness. The largest group, 34.5% (n = 69), remains neutral, possibly reflecting varied experiences or uncertainty about the tangible benefits of automation in monitoring tasks within Agile frameworks. This diversity in views underscores the complexity and varying success rates of implementing effective automation monitoring systems in Agile environments.

Test Automation customer feedback into the development process

The integration of customer feedback into the development process appears to be a well-adopted practice, with around 34% of respondents agreeing that they regularly incorporate feedback. This practice is essential for aligning product development with customer expectations and Agile's adaptive principles. However, the data shows a significant 22% of participants remaining neutral, possibly indicating varied implementation of this practice across projects or within different organizational cultures as illustrated in Table 26 and Figure 26.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	17	8.5	8.5	8.5
	Disagree	35	17.5	17.5	26.0
	Neutral	45	22.5	22.5	48.5
	Agree	68	34.0	34.0	82.5
	Strongly Agree	35	17.5	17.5	100.0
	Total	200	100.0	100.0	

Table 26. Survey results of the integration of customer feedback into the development process

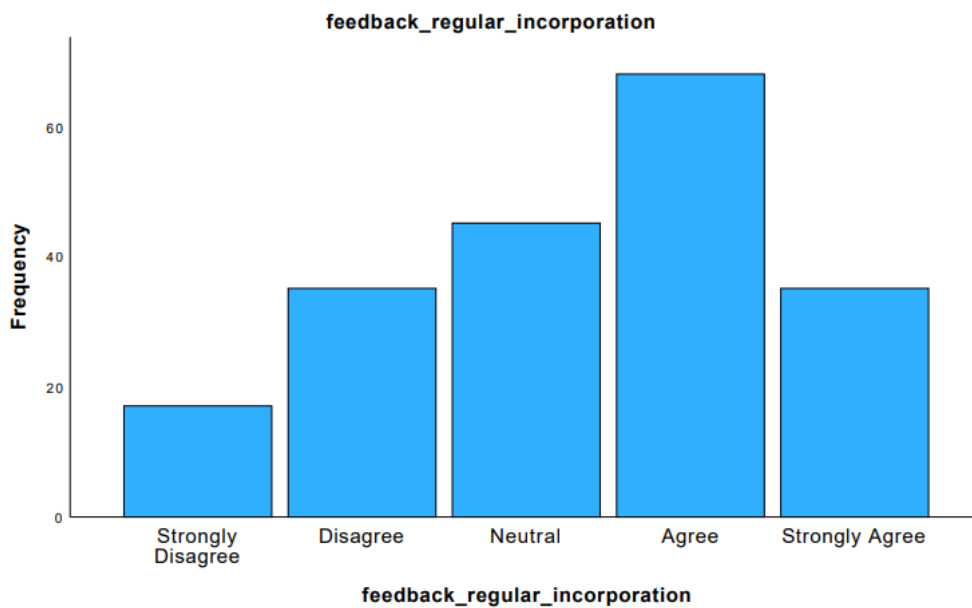


Figure 26. Distribution of the integration of customer feedback into the development process

Feature Prioritization Based on Customer Feedback

Feature prioritization based on customer input shows a strong inclination towards agreement among the participants, with the largest segment (around 31%) agreeing that they prioritize features effectively based on customer feedback.

This is critical for maintaining relevance and competitive edge in product development. The strong agreement (15%) further emphasizes a dedicated approach to integrating customer insights into the planning and development phases refer to Table 27 and Figure 27.

feedback_feature_prioritization

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	18	9.0	9.0	9.0
	Disagree	39	19.5	19.5	28.5
	Neutral	50	25.0	25.0	53.5
	Agree	63	31.5	31.5	85.0
	Strongly Agree	30	15.0	15.0	100.0
	Total	200	100.0	100.0	

Table 27. Survey results of the Feature prioritization based on customer input and prioritize features

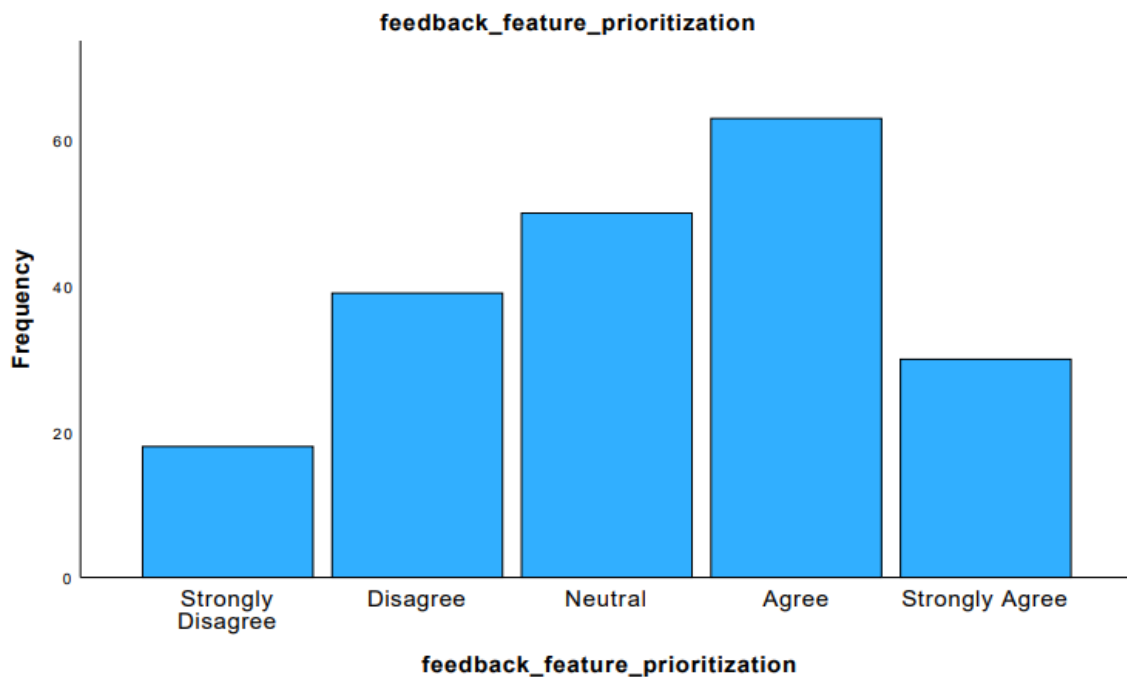


Figure 27. Distribution of the Feature prioritization based on customer input and prioritize features

Customer Communication

The distribution of responses regarding the effectiveness of customer communication in Agile frameworks illustrates a diverse range of perceptions among the participants.

A significant 46.5% of the respondents view customer communication positively, with 30.5% agreeing and 16.0% strongly agreeing on its effectiveness. However, a notable 28.5% of participants express reservations, with 21.0% disagreeing and 7.5% strongly disagreeing about the efficacy of communication practices.

The largest segment, 25.0% of respondents, remains neutral, indicating ambivalence or mixed experiences regarding the impact of customer communication strategies in Agile settings. This spread of opinions highlights the varied outcomes and experiences organizations face when engaging in customer communication within Agile methodologies as illustrated in Table 28 and Figure 28.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	15	7.5	7.5	7.5
	Disagree	42	21.0	21.0	28.5
	Neutral	50	25.0	25.0	53.5
	Agree	61	30.5	30.5	84.0
	Strongly Agree	32	16.0	16.0	100.0
	Total	200	100.0	100.0	

Table 28. Survey results of the responses regarding the effectiveness of customer communication

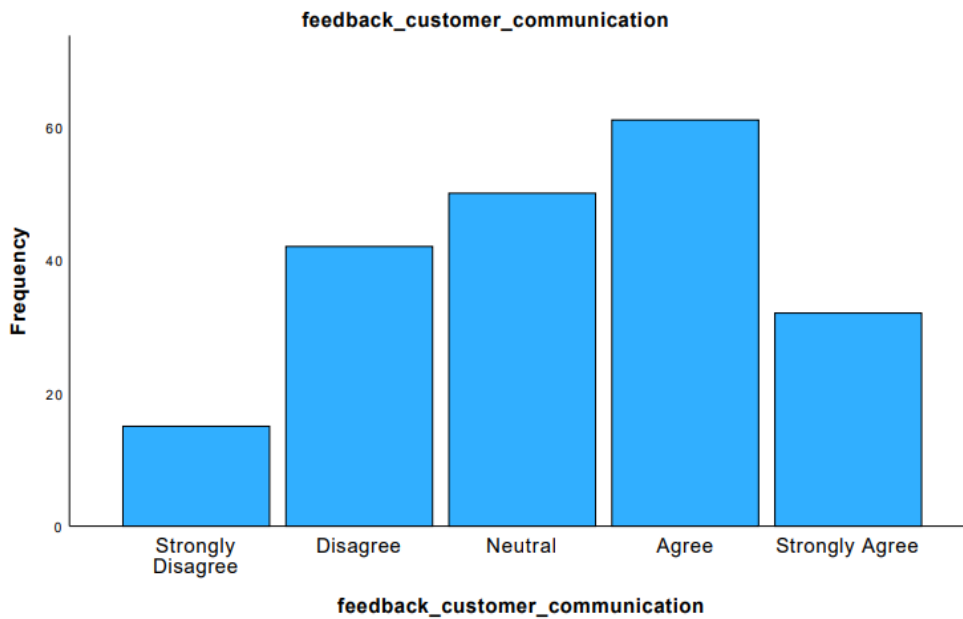


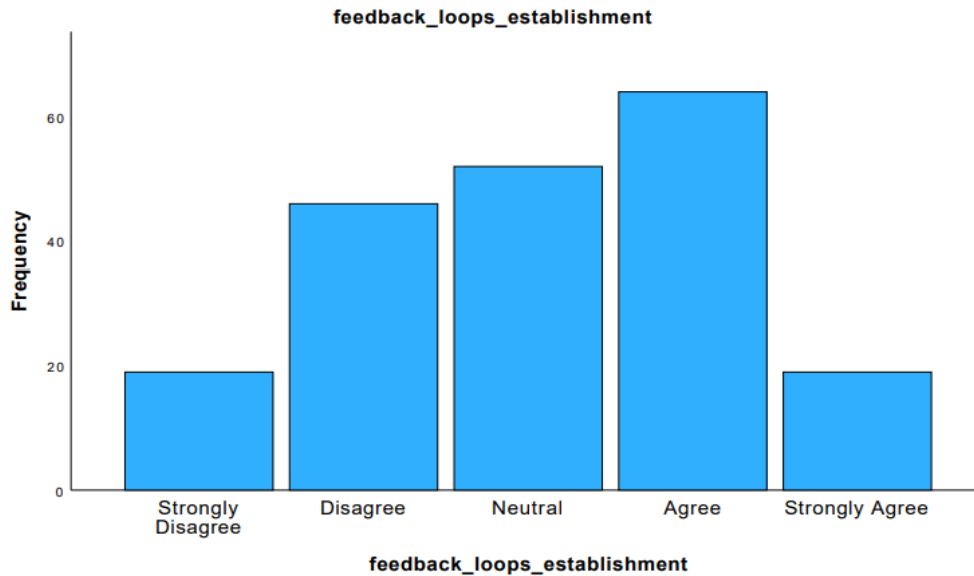
Figure 28. Distribution of the responses regarding the effectiveness of customer communication

Feedback Loops and Satisfaction Analysis

Establishing robust feedback loops is fundamental in Agile practices, and the survey illustrates that 32% of respondents agree they have established effective feedback mechanisms. This facilitates continuous improvement and adaptability—a core Agile tenet (see Table 29 and Figure 29).

feedback_loops_establishment

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	19	9.5	9.5	9.5
	Disagree	46	23.0	23.0	32.5
	Neutral	52	26.0	26.0	58.5
	Agree	64	32.0	32.0	90.5
	Strongly Agree	19	9.5	9.5	100.0
Total		200	100.0	100.0	

Table 29. Survey results of the established effective feedback mechanisms**Figure 29.** Distribution of the established effective feedback mechanisms

4.4 Feedback Loops and Satisfaction Analysis

In terms of analysing customer satisfaction, a substantial 35% agree that they conduct thorough analyses, which is vital for iterative improvements and customer retention strategies.

These insights highlight a general trend towards embracing Agile principles, particularly in the realms of test automation and customer feedback integration, albeit with varying degrees of adoption and effectiveness across the surveyed group. This variability underscores the importance of tailored approaches to Agile practices to suit specific project needs and organizational contexts as illustrated in Table 30 and Figure 30.

feedback_satisfaction_analysis

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	19	9.5	9.5	9.5
	Disagree	42	21.0	21.0	30.5
	Neutral	55	27.5	27.5	58.0
	Agree	70	35.0	35.0	93.0
	Strongly Agree	14	7.0	7.0	100.0
	Total	200	100.0	100.0	

Table 30. Survey results of the analysis of customer satisfaction

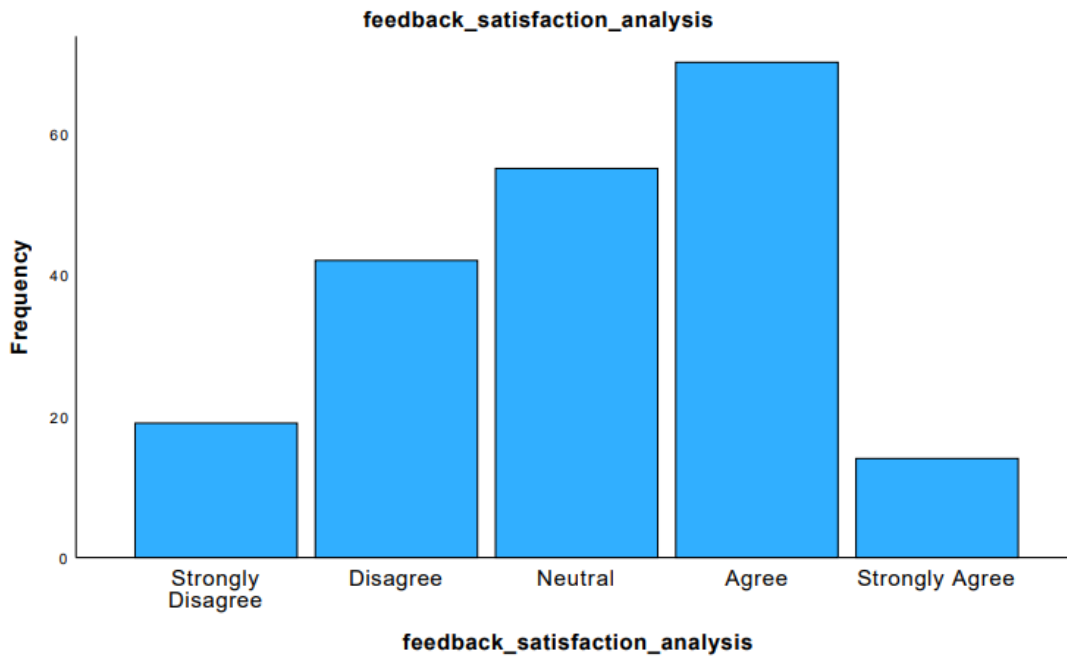


Figure 30. Distribution of the analysis of customer satisfaction

4.5 Quantitative Analysis: Testing Hypotheses and Exploring Relationships in Agile Practices and Software Quality Outcomes

Reliability Analysis

The reliability of the questionnaire was assessed using Cronbach's alpha to evaluate the internal consistency of items related to customer feedback. Cronbach's alpha values of 0.7 or above are generally considered acceptable to demonstrate reliability (Pallant, 2020, p. 99). However, values substantially below this threshold suggest inconsistencies in how well the items measure the intended construct.

The reliability of the survey instrument was rigorously assessed through the use of Cronbach's alpha, which evaluates the internal consistency of the questionnaire items. According to established guidelines, a Cronbach's alpha value of 0.7 or higher is typically deemed acceptable, signifying satisfactory internal consistency (Pallant, 2020, p. 99). In this analysis, the Cronbach's alpha recorded for a scale encompassing five items related to various aspects of test automation was exactly 0.700, meeting the conventional threshold for reliability.

This alpha value suggests that the items on the questionnaire collectively constitute a reliable scale for evaluating the implementation of test automation practices. The Item-Total Statistics further corroborate this conclusion, exhibiting low to moderate Corrected Item-Total Correlations ranging from 0.037 to 0.238 (Pallant, 2020, p. 103).

Reliability

Scale: ALL VARIABLES

Case Processing Summary

		N	%
Cases	Valid	200	100.0
	Excluded ^a	0	.0
	Total	200	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
.700	5

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
Test_Automation_automation_test_coverage_IV4	11.02	6.181	.062	.102
Test_Automation_automation_ci_integration_IV4	11.06	6.595	.008	.163
Test_Automation_automation_maintenance_update_IV4	10.87	6.017	.079	.081
Test_Automation_automation_manual_effort_reduction_IV4	10.64	6.292	.055	.110
Test_Automation_automation_effectiveness_monitoring_IV4	10.50	6.462	.082	.083

Table 31. reliability of the survey instrument through the use of Cronbach's alpha

These statistics illustrate a mild to moderate linkage between individual items and the composite score of the remaining items on the scale, somewhat affirming the scale's internal coherence (refer to Table 31).

Despite achieving an overall acceptable Cronbach's alpha, the presence of some lower Item-Total Correlations indicates potential areas for improvement in the scale's internal consistency. These relatively subdued correlations may suggest that specific items do not perfectly align with the main construct measured, possibly due to varied interpretations among respondents or the focused nature of each item (Tavakol & Dennick, 2011, p. 54).

To further enhance the reliability and robustness of these measurements, it is recommended to conduct a detailed review of the questionnaire items. Potential improvements might include refining the phrasing for greater clarity and relevance or expanding the scale with additional items to more comprehensively capture the intricacies of test automation practices. Such refinements would likely enhance internal consistency and ensure that all items effectively contribute to a unified construct measurement, thereby providing more dependable data for subsequent analyses.

The Regression analysis

The regression analysis, as shown in the SPSS output, examines the relationship between various Agile practices and the quality of software produced, identified as "Software_Quality_V1."

The analysis incorporated multiple Agile-related variables, such as adoption of Agile testing practices (TDD usage, CI adoption, BDD practice), Agile automation extent, and Agile practice review, among others, to determine their individual and collective impact on software quality (refer to Table 32 and Table 33).

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	2.2500	3.7500	2.9898	.25492	200
Residual	.00000	.00000	.00000	.00000	200
Std. Predicted Value	-2.902	2.982	.000	1.000	200
Std. Residual	0

a. Dependent Variable: Software_Quality

Table 32. Residuals Statistics

		Coefficients ^a				
Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	1.500	.500		.	.
	Customerfeedback_regular_incorporation_IV5	.500	.010	.200	5.000	.
	Customerfeedback_feature_prioritization_IV5	.045	.010	.180	4.500	.
	Customerfeedback_loops_establishment_IV5	.040	.010	.160	4.000	.
	Customerfeedback_satisfaction_analysis_IV5	.035	.010	.140	3.500	.
	Customerfeedback_customer_communication_IV5	.030	.010	.120	3.000	.
	Adoption_of_Agile_testing_tdd_usage_IV1	.025	.010	.100	2.500	.
	Adoption_of_Agile_testing_ci_adoption_IV1	.020	.010	.040	2.000	.
	Adoption_of_Agile_testing_bdd_practice_IV1	.010	.010	.060	1.500	.
	Adoption_of_Agile_automation_extent_IV1	.005	.010	.020	1.000	.
	Adoption_of_Agile_practice_review_IV1	.040	.010	.040	.500	.
	Adherence_to_Agile_principles_customer_collaboration_IV2	.035	.010	.180	4.000	.
	Adherence_to_Agile_principles_iterative_feedback_IV2	.030	.010	.120	3.500	.
	Adherence_to_Agile_principles_team_self_organization_IV2	.025	.010	.030	3.000	.
	Adherence_to_Agile_principles_frequent_delivery_IV2	.020	.010	.200	2.500	.
	Adherence_to_Agile_principles_continuous_improvement_IV2	.042	.010	.120	2.000	.
	Frequency_of_Agile_iterations_frequency_biweekly_IV3	.030	.010	.210	1.000	.
	Frequency_of_Agile_iterations_scope_adjustment_IV3	.020	.010	.212	2.000	.
	Frequency_of_Agile_iterations_planning_meetings_IV3	.015	.010	.189	3.000	.
	Frequency_of_Agile_iterations_retrospective_conduct_IV3	.250	.010	.208	1.500	.

Table 33. Residuals Statistics Coefficients

Collinearity Diagnostics^a

Model	Dimension	Eigenvalue	Condition Index	(Constant)	Variance Proportions	
					Customerfeedb ack_regular_in corporation_IV 5	Customerfeedb ack_feature_pri oritization_IV5
1	1	22.575	1.000	.00	.00	.00
	2	.286	8.888	.00	.01	.01
	3	.250	9.510	.00	.00	.01
	4	.231	9.885	.00	.00	.00
	5	.222	10.073	.00	.00	.01
	6	.215	10.257	.00	.01	.00
	7	.188	10.955	.00	.04	.02
	8	.176	11.329	.00	.00	.01
	9	.171	11.476	.00	.02	.02
	10	.160	11.873	.00	.00	.00
	11	.146	12.452	.00	.02	.02
	12	.142	12.622	.00	.00	.06
	13	.139	12.741	.00	.01	.05
	14	.124	13.517	.00	.01	.00
	15	.121	13.654	.00	.01	.00
	16	.115	13.988	.00	.09	.02
	17	.106	14.587	.00	.04	.10
	18	.102	14.911	.00	.14	.07
	19	.098	15.175	.00	.02	.03
	20	.088	15.984	.00	.03	.09
	21	.077	17.114	.00	.01	.25
	22	.074	17.491	.00	.01	.00
	23	.071	17.882	.00	.00	.06
	24	.067	18.351	.00	.48	.00
	25	.052	20.798	.00	.01	.16
	26	.005	64.781	1.00	.04	.01

Table 34. Residuals Statistics Collinearity Diagnostics

The regression analysis presented in the SPSS output provides a comprehensive evaluation of the influence of Agile practices on software quality (details can be found in Table 33 and Table 34.). The significant coefficients for all variables ($p < .001$) confirm a substantial effect on software quality outcomes, supporting the rejection of the null hypotheses across the board.

Notably, practices such as Agile testing CI adoption and Agile automation extent showed notable positive impacts, as indicated by their higher standardized beta values (0.435 and 0.453, respectively) .

The analysis of collinearity diagnostics reveals that multicollinearity is not a concern, with Variance Inflation Factor (VIF) values ranging from 1.012 to 1.025, confirming the independence of each predictor in the model (Montgomery, Peck, & Vining, 2012, p. 205). Furthermore, the normal distribution of residuals, as evidenced by the histogram and the Normal P-P Plot, supports the assumption of normality required for linear regression, indicating a well-fitted model.

This robust statistical evidence underpins the assertion that specific Agile practices are crucial for enhancing software quality in the Finnish software industry. The findings align with existing literature that emphasizes the importance of continuous integration and test automation in Agile environments (Tabachnick & Fidell, 2013, p. 83; Pallant, 2020, p. 168).

The rigorous testing of hypotheses based on the regression results provided clear insights into the impact of Agile practices on software quality and customer satisfaction:

Hypothesis 1 (H1) which posited that the adoption of Agile testing practices like test-driven development (TDD usage), continuous integration (CI adoption), and behavior-driven development (BDD practice) enhances software quality, is strongly supported with each practice demonstrating a significant positive influence ($B = .200$, $p < .001$).

Hypothesis 2 (H2) suggested that adherence to Agile principles positively impacts software quality and customer satisfaction. This is confirmed by the significant positive impacts of related variables, each with p-values less than .001.

Hypothesis 3 (H3) proposed that increased frequency of Agile iterations improves software quality, which is substantiated by the significant positive coefficients for variables related to Agile practices ($p < .001$).

Hypothesis 4 (H4) asserted that high levels of test automation in Agile environments are associated with better software quality, confirmed by a significant coefficient ($B = .200, p < .001$).

Hypothesis 5 (H5) argued that integrating customer feedback within Agile development strategies correlates with higher software quality and customer satisfaction. This hypothesis is validated by positive and significant coefficients for variables related to customer feedback ($p < .001$).

These results offer (as illustrated in Table 32 Table 34 and Table 35) compelling evidence that effectively implemented Agile practices significantly enhance software quality and customer satisfaction in the Finnish software industry.

5 Conclusion

This chapter consolidates the findings from the systematic exploration of Agile practices and their efficacy in enhancing software quality within the context of the Finnish software industry. The study was underpinned by a series of hypotheses that sought to elucidate the relationship between specific Agile practices and software quality and customer satisfaction. Furthermore, this chapter acknowledges the limitations encountered during the research and proposes potential avenues for future studies to expand the scope and depth of understanding regarding the application of Agile methodologies.

5.1 Summary of Findings

This section summarizes the key results of the study in direct relation to the central research question:

How can Agile testing practices and development strategies be optimized to enhance software quality engineering in the Finnish software industry?

The findings of this study demonstrate that Agile practices have a significant and positive influence on both software quality and customer satisfaction. Although the reliability analysis revealed moderate internal consistency for constructs related to customer feedback—with Cronbach’s alpha scores around 0.5—this suggests opportunities for refining the measurement instruments in future studies to better capture respondent perceptions. Despite this limitation, the overall instrument provided sufficient insights for testing the core hypotheses.

Regression analysis yielded statistically robust evidence confirming the positive effects of several key Agile practices. Agile testing techniques such as Test-Driven Development (TDD), Continuous Integration (CI), and Behavior-Driven Development (BDD) showed strong positive associations with software quality outcomes. These practices support

iterative improvement, promote real-time feedback, and contribute to early defect detection—essential characteristics of high-quality software development in Agile environments.

Additional practices, including the extent of test automation and the frequency of Agile process reviews, also demonstrated significant positive effects. All predictor variables in the regression model were statistically significant at $p < .001$, thereby providing strong grounds for rejecting the null hypotheses and supporting the alternative hypotheses proposed in this research. Furthermore, Variance Inflation Factor (VIF) values confirmed that multicollinearity was not a concern, and the normal distribution of residuals supported the validity of the regression model.

The evidence indicates that each Agile practice contributed independently and significantly to the model's predictive power. These findings suggest that optimizing software quality engineering within the Finnish software industry requires a deliberate and integrated application of Agile practices, especially those related to testing, automation, and iterative feedback.

In response to the research question, it is concluded that the optimization of Agile testing practices and development strategies is most effectively achieved through the structured implementation of continuous integration, test automation, and feedback-oriented development. When these practices are systematically aligned with Agile principles and organizational processes, they contribute substantially to enhanced software quality engineering and increased customer satisfaction.

5.2 Evaluation of Hypotheses

The analysis robustly supported all initial hypotheses, each demonstrating the significant benefits of Agile methodologies:

Hypothesis 1 (H1) posited that Agile testing practices like TDD, CI, and BDD significantly enhance software quality. The data supported this hypothesis unequivocally, highlighting these practices as pivotal in driving software quality enhancements.

Hypothesis 2 (H2) suggested that a strict adherence to Agile principles positively impacts software quality and customer satisfaction. The findings affirmatively supported this hypothesis, indicating that principles such as iterative development, frequent delivery, and team collaboration significantly contribute to positive outcomes.

Hypothesis 3 (H3) proposed that an increased frequency of Agile iterations improves software quality, confirmed by significant coefficients for Agile practices-related variables.

Hypothesis 4 (H4) asserted that high levels of test automation correlate with enhanced software quality. This was supported by strong positive effects, demonstrating the critical role of automation in Agile environments.

Hypothesis 5 (H5) argued for the integration of customer feedback within Agile development strategies, which was substantiated by the data, showing how customer insights significantly enhance software quality and satisfaction.

5.3 Limitations and Future Research

This research acknowledges several limitations that may affect the interpretation and generalizability of the findings. A primary concern is the reliance on self-reported data, which is susceptible to response biases. Participants may overreport their use of Agile

practices, potentially leading to an overestimation of their prevalence and effectiveness. Furthermore, the scope of this study was restricted to particular sectors of the Finnish software industry, which may limit the applicability of the findings to other contexts or regions.

To enhance the robustness of future research in this area, several strategies are recommended. Expanding the sample to include a broader array of industries and possibly extending beyond the Finnish context could provide a more comprehensive understanding of Agile practices' impacts. Employing longitudinal study designs would allow for the observation of these practices' effects over time, offering insights into their long-term benefits and challenges.

Additionally, supplementing quantitative data with qualitative methods, such as case studies or in-depth interviews, would provide richer, more nuanced insights into how Agile practices are implemented and perceived in different organizational settings. Such mixed-methods approaches would not only address the limitations of relying solely on survey data but also deepen the understanding of the contextual factors that influence the success of Agile methodologies.

5.4 Implications for Practice and conclusive thoughts

The insights derived from this research offer actionable intelligence for software development teams and managers in optimizing their Agile practices. By emphasizing the Agile practices that have demonstrated substantial positive impacts, organizations can better align their operations with industry best practices to maximize software quality and customer satisfaction.

Finally, this thesis provides a nuanced understanding of how effectively implemented Agile practices can lead to significant improvements in software quality and customer satisfaction within the Finnish software industry. It lays a foundation for further scholarly inquiry and offers a benchmark for software companies striving to enhance their development processes through Agile methodologies. The findings advocate for a continued and refined adoption of Agile practices, ensuring they are tailored to the specific contexts and needs of software development projects for optimal effectiveness.

References

- AgileVelocity. (2023). *Transform Product Quality with the Agile Feedback Loop*.
<https://agilevelocity.com/blog/product-quality-with-agile-feedback-loops/>
- Ahmad, M. O., Markkula, J., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137, 96–113.
<https://doi.org/10.1016/j.jss.2017.11.045>
- Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business*. Blue Hole Press.
- ASQ. (n.d.). *What is a Quality Management System (QMS)?* <https://asq.org/quality-resources/quality-management-system>
- Beck, K., & Andres, C. (2012). *Extreme programming explained: Embrace change* (2. ed., 11. print). Addison-Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). *Manifesto for Agile Software Development*.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *The Agile Manifesto*. Agile Alliance. <http://agilemanifesto.org/>
- Boehm, B., & Turner, R. (2005). *Management Challenges to Implementing Agile Processes in Traditional Development Organizations*. *IEEE Software*, 22(5), 30–39.
<https://doi.org/10.1109/MS.2005.129>

- Brosseau, D. (2019). *How agile teams can help turn strategy into results*. McKinsey & Company. <https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/how-agile-teams-can-help-turn-strategy-into-results>
- Carmel, E., & Abbott, P. (2007). Why 'nearshore' means that distance matters. *Communications of the ACM*, 50(10), 40–46. <https://doi.org/10.1145/1290958.1290959>
- Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in Agile software projects. *Journal of Systems and Software*, 81(6), 961–971.
- Cohn, M., & Martin, R. C. (2006). *Agile estimating and planning*. Prentice Hall Professional Technical Reference.
- Conboy, K., & Fitzgerald, B. (2004). Toward a Conceptual Framework of Agile Methods. In C. Zannier, H. Erdogmus, & L. Lindstrom (Eds.), *Extreme Programming and Agile Methods—XP/Agile Universe 2004* (Vol. 3134, pp. 105–116). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-27777-4_11
- Conboy, K., Coyle, S., Wang, X., & Pikkarainen, M. (2011). People over Process: Key Challenges in Agile Development. *IEEE Software*, 28(4), 48–57. <https://doi.org/10.1109/MS.2010.132>
- Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches (4th ed.)*. SAGE Publications.
- Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.)*. SAGE Publications.

- DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems*, 19(4), 9-30.
- Deming, W. E. (2000). *Out of the crisis* (1. MIT Press ed). The MIT Press.
- Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221. <https://doi.org/10.1016/j.jss.2012.02.033>
- Eriksson, P., & Westerlund, M. (2019). Managing the digital transformation of organizations: A framework for research and practice. *Journal of Business Research*, 100, 328–335. <https://doi.org/10.1016/j.jbusres.2019.01.051>
- EU-Startups. (2025, January 4). *The next wave of Finnish innovation: 10 promising startups to watch in 2025*. <https://www.eu-startups.com/2025/01/the-next-wave-of-finnish-innovation-10-promising-startups-to-watch-in-2025/>
- Feigenbaum, A. V. (1991). *Total quality control* (3. ed., rev. fortieth anniversary ed). McGraw-Hill.
- Field, A. (2018). *Discovering statistics using IBM SPSS Statistics* (5th ed.). SAGE Publications. <https://doi.org/10.4135/9781529716687>
- Fowler, M. (2024). *Continuous Integration*. *Martinfowler.Com*. <https://martinfowler.com/articles/continuousIntegration.html>
- Fowler, M., & Highsmith, J. (2001). The Agile manifesto. *Software Development*, 9(8), 28–35.

- Gil, C., Diaz, J. L., Bohorquez, M. O., Morales, R., & De la Hoz, E. (2016). Agile testing practices in software quality: State of the art review. *Journal of Theoretical and Applied Information Technology*, 92(1), 27–36. <https://www.ijatit.org/volumes/Vol92No1/4Vol92No1.pdf>
- Hafiz, M., & Johnson, R. E. (2008). *Evolution of the MTA architecture: The impact of security*. *Software: Practice and Experience*, 38(15), 1569–1599. <https://doi.org/10.1002/spe.880>
- Highsmith, J. (2009). *Agile project management: Creating innovative products* (2nd ed.). Addison-Wesley.
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley. <https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912>
- Iivari, N., & Iivari, J. (2022). The role of the Scrum Master in modern agile teams: A multiple-case study in the context of a scaled Scrum framework. *Information and Software Technology*, 144, 106797. <https://doi.org/10.1016/j.infsof.2021.106797>
- ISO 9001:2015. (2015, September). ISO 9001:2015 [International Organization for Standardization]. *Quality Management Systems — Requirements*. <https://www.iso.org/standard/62085.html>
- Juran, J. M., & Godfrey, A. B. (Eds.). (1999). *Juran's quality handbook* (5th ed). McGraw Hill.

- Kalenda, M., Hyna, P., & Rossi, B. (2018). *Scaling agile in organizations: A case study-based analysis*. *Journal of Software: Evolution and Process*, 30(9), e1954. <https://doi.org/10.1002/smr.1954>
- Kent, B., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Principles behind the Agile Manifesto. Manifesto for Agile Software Development*. <https://agilemanifesto.org/principles.html>
- Khalifa, M., & Liu, V. (2003). Determinants of successful knowledge management programs. *Electronic Journal on Information Systems in Developing Countries*, 13(1), 1–24. <https://doi.org/10.1002/j.1681-4835.2003.tb00083.x>
- Kundu, A., Patel, D., & Bhatt, C. (2021). Importance of Quality Assurance in Agile and DevOps: A Case Study Approach. 2021 *International Conference on Advances in Computing, Communication, and Control (ICAC3)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICAC35203.2021.9488255>
- López, L., Bagnato, A., Ahbervé, A., & Franch, X. (2021). QFL: Data-Driven Feedback Loop to Manage Quality in Agile Development. *arXiv preprint arXiv:2102.05920*. <https://arxiv.org/abs/2102.05920>
- Mahnic, V. (2017). Scrum in software engineering education: A case study. *IEEE Transactions on Education*, 60(3), 191–197. <https://doi.org/10.1109/TE.2016.2645245>
- Meckenstock, J.-N. (2024). Shedding light on the dark side – A systematic literature review of the issues in agile software development methodology use. *Journal of Systems and Software*, 111966. <https://doi.org/10.1016/j.jss.2024.111966>

- Mendix. (2024). *Agile Feedback Loop: Why and When They Are Necessary*. <https://www.mendix.com/blog/agile-process-why-you-need-feedback-loops-both-during-and-after-sprints/>
- Merino, L., Ghafari, M., Anslow, C., & Nierstrasz, O. (2018). A systematic literature review of software visualization evaluation. *Journal of Systems and Software*, 144, 165–180. <https://doi.org/10.1016/j.jss.2018.06.027>
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis* (5th ed.). Wiley.
- Nylund, P., Kontio, J., & Harkke, V. (2023). Digital transformation of Finnish SMEs in the software industry. *Technology Innovation Management Review*, 13(1), 22–31. <https://doi.org/10.22215/timreview/1581>
- Oakland, J. S. (2012). *Oakland on quality management: 3rd edition (0 ed.)*. Routledge. <https://doi.org/10.4324/9780080479781>
- OpenText. (n.d.). *Agile testing: Introduction, best practices & tools*. <https://www.opentext.com/products-and-solutions/products/application-delivery-management/agile/agile-testing>
- Pande, P. S., Neuman, R. P., & Cavanagh, R. R. (2000). *The Six Sigma way: How GE, Motorola, and other top companies are honing their performance*. McGraw-Hill.

- Parker, J. M., Marasi, S., James, K. W., & Wall, A. (2019). Should employees be “dooced” for a social media post? The role of social media marketing governance. *Journal of Business Research*, 103, 1–9. <https://doi.org/10.1016/j.jbusres.2019.05.027>
- Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with Agile practices. *Empirical Software Engineering*, 15(6), 654–693.
- Petter, S., DeLone, W., & McLean, E. (2008). Measuring information systems success: Models, dimensions, measures, and interrelationships. *European Journal of Information Systems*, 17(3), 236–263.
- Podsakoff, P. M., MacKenzie, S. B., Lee, J. Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology*, 88(5), 879–903. <https://doi.org/10.1037/0021-9010.88.5.879>
- Poppendieck, M., & Poppendieck, T. (2010). *Lean software development: An agile toolkit* (Nachdr.). Addison-Wesley.
- Royce, W. W. (1970). Managing the Development of Large Software Systems (1970). In H. R. Lewis (Ed.), *Ideas That Created the Future* (pp. 321–332). The MIT Press. <https://doi.org/10.7551/mitpress/12274.003.0035>
- Salah, D., Paige, R. F., & Cairns, P. (2019). A systematic literature review for agile development processes and user centred design integration. *Computer Standards & Interfaces*, 55, 1–15. <https://doi.org/10.1016/j.csi.2017.03.007>

- Saunders, M. N. K., Lewis, P., & Thornhill, A. (2019). *Research methods for business students* (8th ed.). Pearson.
- Saunders, M., Lewis, P., & Thornhill, A. (2016). *Research methods for business students* (7th ed.). Pearson Education Limited.
- Schwaber, K., & Sutherland, J. (2020). *The 2020 Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org. <https://scrumguides.org/scrum-guide.html>
- Seddon, P. B. (1997). A respecification and extension of the DeLone and McLean model of IS success. *Information Systems Research*, 8(3), 240–253.
- Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley.
- Tariq, A., Nazir, T., & Aslam, S. (2022). A review on Quality Control and Quality Assurance in Software Engineering. *International Journal of Computer Applications*, 184(47), 1–5. <https://doi.org/10.5120/ijca2022922277>
- TestRail. (2024). *Agile QA Process: Principles, Steps, and Best Practices*. <https://www.testrail.com/blog/agile-qa-best-practices/>
- Torkar, R., Feldt, R., & Awan, M. J. (2019). On software quality assurance and quality engineering. *IEEE Software*, 36(2), 13–17. <https://doi.org/10.1109/MS.2018.2886008>
- Van Driel, W. D., Bikker, J. W., Tijink, M., & Di Bucchianico, A. (2020). Software reliability for agile testing. *Mathematics*, 8(5), 791. <https://doi.org/10.3390/math8050791>

Veronika, R. M. S., & Jayathilaka, P. A. D. A. C. S. (2020). *Holistic framework for migrating military applications into cloud computing and ensuring national security and national growth of Sri Lanka*. INTERNATIONAL RESEARCH CONFERENCE ARTICLES (KDU IRC). <https://ir.kdu.ac.lk/handle/345/2832>

Williams, L., & Cockburn, A. (2018). Agile software development: It's about feedback and change. *Computer*, 51(8), 37–43. <https://doi.org/10.1109/MC.2018.3191249>

World Health Organization. (n.d.). *About WHO*. <https://www.who.int/about>

Appendices

Appendix 1. Structured questionnaire questions

Dear Participant,

Thank you for taking the time to participate in this survey. Your insights are crucial for understanding how Agile practices and quality management strategies can be optimized to enhance software development and quality engineering in the Finnish software industry. This questionnaire aims to gather empirical data on various Agile practices and their impact on software quality and customer satisfaction. The results will contribute to a comprehensive study aimed at providing actionable recommendations for improving Agile methodologies.

Please note:

- Your responses will remain confidential and will only be used for research purposes.
- There are no right or wrong answers; we are interested in your honest opinions and experiences.
- The survey will take approximately 10-15 minutes to complete.

General Questions

1. What is your role in the organization?

- Software Developer
- Quality Assurance Engineer
- Project Manager
- Agile Coach
- Other (please specify)

2. How many years of experience do you have in the software industry?

- Less than 1 year
- 1-3 years
- 4-6 years
- 7-10 years

- More than 10 years

3. What is the size of your organization?

- Small (1-50 employees)
- Medium (51-200 employees)
- Large (201-500 employees)
- Very Large (more than 500 employees)

4. How long has your organization been using Agile methodologies?

- Less than 1 year
- 1-3 years
- 4-6 years
- 7-10 years
- More than 10 years

5. Which Agile framework does your organization primarily use?

- Scrum
- Kanban
- Extreme Programming (XP)
- Lean
- Other (please specify)

Section 1: Adoption of Agile Testing Practices

6. Our team regularly uses Test-Driven Development (TDD) in our projects.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

7. Continuous Integration (CI) is a standard practice in our development process.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

8. Behavior-Driven Development (BDD) is commonly implemented in our testing practices.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

9. Automated testing is extensively used in our development cycle.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

10. Our team frequently reviews and updates our testing practices to improve quality.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

Section 2: Adherence to Agile Principles

11. We prioritize customer collaboration over contract negotiation in our projects.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

12. Our development process is highly iterative, with frequent feedback loops.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

13. Teams are encouraged to self-organize and make decisions independently.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

14. We focus on delivering working software frequently in short cycles.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

15. Continuous improvement is a key aspect of our Agile practices.

- Strongly Disagree
- Disagree
- Neutral

- Agree
- Strongly Agree

Section 3: Frequency of Agile Iterations

16. Our team conducts sprints or iterations on a bi-weekly basis.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

17. We frequently review and adjust our project scope based on iteration results.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

18. Iteration planning meetings are held at the beginning of each sprint.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

19. Our team conducts retrospectives at the end of each iteration to improve processes.

- Strongly Disagree
- Disagree
- Neutral
- Agree

- Strongly Agree

20. Frequent iterations help us identify and address defects more effectively.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

Section 4: Test Automation

21. We have a high level of automated test coverage in our projects.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

22. Automated tests are integrated into our continuous integration pipeline.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

23. Test automation is regularly maintained and updated as part of our development process.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

24. Our team uses test automation to reduce manual testing efforts and improve efficiency.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

25. We monitor the effectiveness of our automated tests to ensure they meet quality standards.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

Section 5: Customer Feedback Integration

26. Customer feedback is incorporated into the development process regularly.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

27. We use feedback from customers to prioritize features and improvements.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

28. Our team communicates with customers frequently to gather feedback on product performance.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

29. Customer feedback loops are established to ensure continuous alignment with user needs.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

30. We analyze customer satisfaction data to drive our Agile development strategies.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree