NOVA IMS Information Management School

MGI

Master Degree Program in

Information Management

Guidelines for the Use of Emerging Technologies in Agile Software Development Project

Victoria Uju Nwakuwa

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

GUIDELINES FOR THE USE OF EMERGING TECHNOLOGIES IN AGILE SOFTWARE DEVELOPMENT PROJECT

By

VICTORIA UJU NWAKUWA

Master Thesis

Master Thesis presented as partial requirement for obtaining the master's degree in Information Management, with a specialization in Information Systems and Technologies Management

Supervised by

Vitor Manuel Pereira Duarte dos Santos, PhD, NOVA Information Management School

July 2025

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

11/07/2025

VICTORIA UJU NWAKUWA

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to God Almighty for granting me the strength, wisdom, and perseverance to successfully complete this project.

I am deeply thankful to my supervisor, [Professor Vitor Santos], for their invaluable guidance, continuous support, and constructive feedback throughout the course of this research. Their insight and encouragement played a crucial role in shaping this work.

My appreciation also goes to the faculty and staff of [Information system Management] at [NOVAIMS] for creating an academic environment that nurtures research and critical thinking.

"Thank you to all the participants and respondents who took the time to share their thoughts. This research wouldn't have been possible without your help." To my family and friends, thank you for your unwavering love, encouragement, and patience during the times I needed it most.

Finally, I appreciate everyone mentioned or unmentioned that make out time to answer my survey in one way or another for the successful of this project.

Thank you all.

Dedication

This thesis work is dedicated to God Almighty, whose grace, wisdom, and guidance have sustained me throughout this journey.

To my beloved parents, whose love, prayers, and sacrifices have been my foundation thank you for your constant encouragement and belief in my abilities.

To my family and friends, who stood by me with patience, support, and understanding I am deeply grateful.

And to everyone who inspires me to grow and never give up this work is for you.

ABSTRACT

This study focuses on how emerging technologies can be effectively integrated into Agile software development projects to create a guideline or framework that helps Agile teams adopt and manage new technologies. Emerging technology such as Blockchain, Internet of Things and Artificial Intelligence is reshaping Agile software development, offering new opportunities and challenges. This study explores how these technologies influence Agile projects and identifies key factors for their successful adoption. Therefore, the research adopts a design science research methodology to explore how organizations can effectively implement and manage emerging technologies within Agile software development projects. Using a mixed-method approach, including surveys and, the research provides insights into optimizing Agile frameworks for technological advancements. Grounded in Technological Determinism theory, the findings offer practical guidelines for industry professionals and researchers. Ultimately, this study aims to enhance project performance, improve decision making, and support future innovations of the project. To examine the literature review and systematic interviews analysis are conducted to establish the foundation for a comprehensive implementation framework. This includes key components such as readiness assessment, strategic alignment, performance evaluation, and continuous improvement. The framework is developed to guide organizations through the integration process while aligning with Agile principles. Validation is achieved through expert feedback and use case applications, confirming the framework is both relevant and practical for real-world use. As a result, this research contributes a structured, evidence-based approach to bridging the gap between technological advancement and Agile practice. Furthermore, the study provides important implications for both practitioners and future academic inquiry into the intersection of Agile methodologies and emerging technologies. The study resulted in a structured implementation framework that includes readiness assessment tools, performance evaluation metrics, and iterative improvement mechanisms tailored to emerging technologies within Agile settings. Key outcomes revealed that organizations with clear integration strategies and continuous learning cultures were more successful in leveraging new technologies to enhance flexibility, speed, and team collaboration. Additionally, empirical evaluation with industry professionals confirmed the practicality and relevance of the proposed guidelines. These outcomes reinforce the importance of aligning Agile practices with technological evolution to sustain competitive advantage in fast-paced development environments.

KEYWORDS

Emerging Technology; Automation Tools; Agile Methodologies; Software Development Project

INDEX

Contents

1. Introduction....1
1.1 Context and Problem Identification....1
1.2 Research Gap and Objectives....2
1.3 Study Relevance and Contributions....3
1.4 Expected Results....3
2.1 Agile Methodology....5
2.1.1 Twelve Principles of Agile Software Development....7
2.1.2 Overview in Agile Methodology....7
2.2 Popular Agile Methodologies and Frameworks....10
2.2.1 Scrum....10
2.2.2 Extreme Programming (XP)Methodology....12
2.2.3 Adaptive Software Development (ASD)....14
2.2.4 Lean Software Development....15
2.3 The Role of Technology in Agile Project....16
2.4 Challenges and Opportunities....17
2.5 Systematic Review for Emerging Technology in Agile Software Development....18
2.5.1 Prisma Methodology....19
3. Methodology....34
3.1 Design Science Research....34
3.2 Design Science Research Process....36
3.3 Research Strategy....37
3.4 Research Implementation....39
4. Empirical Study Guidelines for Emerging Technologies in Agile Software....40
4.1 Assumptions....40
4.2.1 Emerging Technology in Agile software Development Strategy....44
4.2.2 Readiness Assessment for Agile Implementation....46

4.2.3 Agile Implementation 50
4.2.4 Performance Assessment for Emerging Technology in Agile Software 51
4.2.5 Continuous Improvement in Agile Implementation 52
4.3 Use Case 53
5. Evaluation and Discussion 57
5.1 Revised Model 60
5.2 Result of the Survey Respondent 63
5.3 Correlational Analysis Overview 73
6. Conclusions 78
6.1 Synthesis of the Proposed Framework 78
6.2 Study Limitations and Suggestions for Future Research 78
Bibliographical 80
APPENDIX A 88
APPENDIX B 89
APPENDIX C 91

LIST OF FIGURES

Figure 1: Agile Project Management Methodology- Source: ChatGPT (OpenAI), 2025.....5
Figure 2: Four (4) Values of Agile Manifesto Source - ChatGPT (OpenAI), 2025.....6
Figure 3: 12 Principles of Agile Methodology - Source: Beck, K., et al. (2001).....7
Figure 4: 12 Principles of Agile Practices- Source, OpenAI, (2025).....9
Figure 5 : Popular Agile Methodologies and Model - Source: OpenAI, (2025).....10
Figure 6: Scrum Process- Source: OpenAI, (2025).....12
Figure 7 : Extreme Programming (XP) Methodology- Source: OpenAI, (2025).....14
Figure 8: Adaptive Software Development -Source: OpenAI, 2025.....15
Figure 9: Lean Software Development Guide - Source: OpenAI, (2025).....16
Figure 10: Prisma Execution.....22
Figure 11: Design Science Research Framework Adapted - Source: Hevner et al. (2004).....35
Figure 12: DSR Methodology Process Model Adapted - Source: Peffers et al. (2008).....36
Figure 13: DSRM Process Model Adaptation - Source: Peffers et al., (2007).....38
Figure 14: Systematic for emerging technology in agile software implementation framework (own illustration) ..41
Figure 15: Agile software development Strategy workflow (own illustration).....44
Figure 16 : Readiness Assessment workflow for the Adoption of Emerging Technologies in Agile own illustration)48
Figure 17: Performance Assessment for Emerging Technology in Agile Software own illustration).....52
Figure 18: Use case step one 53
Figure 19: Use case step two 53
Figure 20: Use case step Three 54
Figure 21: Use case step four.....54
Figure 22: Use case step five.....54
Figure 23: Use case step six.....55
Figure 24: Use case step seven 55
Figure 25: Use case step eight.....55
Figure 26: Completed emerging technology in agile software implementation process own illustration).....56
Figure 27: Q1 Result of the survey analyzed using Qualtrics.....63
Figure 28: Q2 Result of the survey analyzed using Qualtrics.....64
Figure 29: Q3 Result of the survey analyzed using Qualtrics.....64
Figure 30: Q4 Result of the survey analyzed using Qualtrics.....65
Figure 31: Q5 Result of the survey analyzed using Qualtrics.....65
Figure 32: Q6 Result of the survey analyzed using Qualtrics.....66
Figure 33 :Q7 Result of the survey analyzed using Qualtrics.....66
Figure 34: Q8 Result of the survey analyzed using Qualtrics.....67
Figure 35: Q9 Result of the survey analyzed using Qualtrics.....67
Figure 36: Q10 Result of the survey analyzed using Qualtrics.....68
Figure 37: Q11 Result of the survey analyzed using Qualtrics.....68
Figure 38: Q12 Result of the survey analyzed using Qualtrics.....69

Figure 39: Q13 Result of the survey analyzed using Qualtrics. 69
Figure 40: Q14 Result of the survey analyzed using Qualtrics. 69
Figure 41: Q15 Result of the survey analyzed using Qualtrics. 70
Figure 42: Q16 Result of the survey analyzed using Qualtrics. 70
Figure 43: Q17 Result of the survey analyzed using Qualtrics. 71
Figure 44: Q18 Result of the survey analyzed using Qualtrics. 71
Figure 45: Q19 Result of the survey analyzed using Qualtrics. 72
Figure 46: Q20 Result of the survey analyzed using Qualtrics. 72
Figure 47: Q21 Result of the survey analyzed using Qualtrics. 73
Figure 48: Q22 Result of the survey analyzed using Qualtrics. 73

LIST OF TABLES

Table 1: PRISMA Execution 20
Table *2*: Systematic Review's Keywords 20
Table 3:Systematic Review's Resource Data bases 21
Table 4: Systematic Review's inclusion and exclusion criteria 21
Table 5: Prisma Results Table 23
Table 6: List of selected experts, including their professional roles and areas of expertise 57
Table 7: Revised rating scale and readiness classification 60
Table 8**:** Revised readiness model 61
Table 9: Comprehensive Table: Thesis Objectives, Readiness Framework, Spearman Insights, and Scores 74

LIST OF ABBREVIATIONS AND ACRONYMS

AI Artificial Intelligence

ANFIS Adaptive Neuro-Fuzzy Inference System

APM Adaptive Project Management

ASD Adaptive Software Development

AWS Amazon Web Services

BU Business Unit

CI Continuous Integration

CIO Chief Information Officer

CVR Content Validity Ratio

DK Design Knowledge

DSR Design Science Research

DSRM Design Science Research Methodology

DevOps Development and Operation

ICT Information and Communication Technologies

IoT Internet of Things

IS Information Systems

ISO International Standardization for Organization

IT Information Technology

JAD Joint Application Design

JIRA (Proper name, often used as an acronym for project tracking tool)

LR Literature Review

ML Machine Learning

Nova Tech New Technology

OKR Objectives and Key Results

PRISMA Preferred Reporting Items for Systematic Reviews and Meta-Analysis RPA Robotic Process Automation

RQ Research Question

SAFe Scale Agile Framework

SDG Sustainable Development Goals

SLR Systematic Literature Review

SVN Subversion (Version Control System)

TDD Test-Driven Development

UN United Nations

XP Extreme Programming

1. Introduction

1.1 Context and Problem Identification

This study focuses on how emerging technologies can be effectively integrated into Agile software development projects, to create a guideline or framework that helps Agile teams adopt and manage new technologies. As the landscape of agile software development continues to change, emerging technologies have and will be shaping this environment with new opportunities but also challenges. Agile method emphasizes collaboration flexibility, continuous improvement, and fast delivery qualities that are especially valuable as new technologies like blockchain and IoT become more central to how we build modern applications. Rules for successfully adopting such practices in an agile environment are basic for the emerging of development triumph with accommodate as well technological progression. The entire build-or-buy open-source or third-party license landscape in commercial software shares no parallel with modern web development due to the rate of technological advancement. Cutting-edge innovations, like emerging technologies have significantly transformed the way to dismantle the former ways which were less efficient compared to these advanced technologies. Agile practices are inherently adaptive and responsive, making them particularly well-suited for integrating new technological advancements Schwaber & Sutherland, (2020). This flexibility is integral for entities looking to be modern monopolies - being more enduring and competitive in the rapidly changing digital world. The Agile Software Development methodologies such as Scrum, Kanban & Extreme Programming (XP) emphasis incremental development with a focus on feedback and cooperative team interaction. It is important to note that the incorporation of these frameworks with modern technologies can significantly improve their efficiency. This includes automation of repetitive tasks; predictive analytics offering insights into potential future events, and machine learning which adapts -hence learns- as it is fed more data providing an enhancement for decision making process. For example, Alenezi, M., & Akour, M. (2025) suggestive that AI-driven tools could be used to improve the agile development process by enabling the early detection of possible issues and even recommending changes in code or automating part of a testing procedure. These capabilities streamline workflows and enhance productivity, as well as the quality of software development projects. Agile projects are unique in that the nature of blockchain decentralization plays a direct advantage to this method as well its secure operation. It can be interesting to increase transparency, avoid data integrity and secure code among software developers. One approach to this problem is the ability of Blockchain, as, Kshetri, N., (2025) state "Blockchain's immutable ledger can be used to create a transparent record of all development activities that are stored on tamper-proof manner leading the whole team into accountability which twists trust among them". In environments where data integrity and security are critical (ex: financial services, healthcare applications) this can be of immense benefit. A new technology, Internet of Things (IoT), can also contribute to a more agile development. Internet of Things, or IoT devices generate vast amounts of data capable of quietly sitting in the background pulling user behavioural and system performance insights. This is useful for Agile teams to inform decisions as they design, iterate upon, and deliver more user-centric solutions. This was pointed out by Gubbi et al (2021) in a research effort to investigate the IoT as needs arise. Integration of IoT in Agile Development: Real-time monitoring and feedback through which teams can adapt quickly to the changing needs as well as environmental conditions. This kind of adaptability and flexibility in real-time is the lifeblood of agile. Moreover, cloud computing has also changed the game for agile teams that need to scale quickly because of their dynamic projects. Cloud platforms allow for collaborative continuous integration and deployment as it is a faster developed cycle process with reliable releases. According to Armbrust et al. & T., (2022), "Cloud computing supports agile practices by providing an elastic infrastructure that can rapidly expand or contract in order to meet project demands, thereby facilitating rapid development, testing and deployment". Ideally you want them

to be able to move quickly and create high quality software in short periods these are things driven by agile teams. While leveraging emerging technologies with agile projects does offer a plethora of advantages, the task is not without its own set of problems. This ranges from specialization to possible points of insecurity and issues regarding managing multiple different technologies within one company. As noted by Santosh B. Rane. (2022) as cited in, "Thus, for the successful utilization of emerging technologies within an agile project is not only dependent on the technical skillset possessed by those engaging with them but also requires a strategic attitude towards mitigating potential threats and issues". Training, clear guardrails and most importantly a culture of lifelong learning is something which organizations need to firmly believe in to use these technologies effectively for their Agile practices. In summary, the incorporation of new technologies in agile software development has great potential to improve efficiency, innovation, and quality. While organizations have benefits and challenges in using these technologies which they need to understand; hence, develop guidelines best practices as how they can effectively adapt them into their agile processes. Agility, with the potential of emerging technologies even in terms to embody entirely new levels and customer satisfaction levels that have never been seen before Schwaber & Sutherland, (2020). The combination of agile methodologies and cutting-edge technologies is going to redefine the future in software development. The gap of this research includes reviewing existing literature to identify areas that are not yet explored or fully understood. Existing studies often focus on isolated aspects of Information Systems without fully integrating the latest technological advancements. According to Ceravola, A., Stein, M., & Goerick**,** C. (2008). "The rapid evolution of technology often leaves gaps in research, particularly in how new tools and systems can be effectively integrated into existing infrastructures". This underscores the need for continuous review and updating of literature to keep pace with technological progress. Furthermore, this research will examine how emerging technologies integrate with traditional Information Systems and their impact on architectural choices and system performance. Emerging technologies, such as artificial intelligence, blockchain, and cloud computing, are transforming the landscape of Information Systems. As noted by Zhang and Lu (2021), "The integration of emerging technologies with traditional Information Systems presents both opportunities and challenges, particularly in terms of system architecture and performance optimization". This integration requires a thorough understanding of how these technologies can coexist and enhance overall system efficiency. In addition to the technological aspects, the research will also explore the strategic implications of these integrations. The architectural choices made during the integration of new technologies can significantly influence system performance and organizational agility. According to Escandon-Barbosa, D., & Salas-Paramo, J. (2025). "Strategic architectural decisions are crucial in leveraging the full potential of emerging technologies while maintaining system stability and performance". This highlights the importance of a well thought out approach to integrating new technologies into existing systems. Finally, the impact on system performance will be a key focus area. Performance metrics such as speed, reliability, and scalability are critical indicators of the success of integrating emerging technologies with traditional systems. Rai, A., Patnayakuni, R. (2016) emphasize, "Assessing the performance impact of integrating new technologies with traditional Information Systems is essential for ensuring that these integrations deliver the desired improvements in efficiency and effectiveness".

1.2 Research Gap and Objectives

This aspect of the research will provide valuable insights into how organizations can optimize their Information Systems to achieve better performance outcomes and try to answer the following research questions:

- RQ1 How do emerging technologies influence software development projects in agile environments?
- RQ2 What are the critical factors influencing the success of software development projects in agile environments?

Research Objectives

The following objective will guide this study: Build a comprehensive set of guidelines for the use of emerging technologies in agile software development projects. Here are the main steps we need to take to achieve this goal.

- ROBJ 1 To analyse the use of emerging technologies e.g (Blockchain, AI,) in agile software development.
- ROBJ 2 To explore the impact of agile methodologies on software development project outcomes.
- ROBJ 3 To build guidelines.
- ROBJ 4 To evaluate guidelines.

1.3 Study Relevance and Contributions

The study has the following significances. Firstly, it guarantees that implementation of these technologies is in accordance with the key values of the agile methodology, including flexibility, constant delivery, and interconnection. Since new technologies come with new challenges, guidelines ensure that these challenges are well handled without affecting the efficiency of development. Talukdar**,** M. R. I. (2020). opined that "Agile frameworks call for contingency plans that are flexible enough to support the evolution of the emergent tech and fundamental agile tenets". Secondly, there are clearly established procedures that are likely to prevent various risks that are likely to arise from the adoption of new technologies, for instance, security risks or compatibility. As it would be seen from the literature, Moran, A. (2014). that ''it is important to adopt efficient risk management programmes in agile environments because the stability of iterations demands that risks that may be created by new technologies are effectively managed.'' Here, risk management is critical towards ensuring that investment in technology yields secure and dependable economic systems to eliminate possibilities of incurring significant economic losses in case of system failure or breach. Moreover, these guidelines ensure that best practices are effectively documented and followed, reducing the potential for confusion or misunderstanding between team members and providing for consistent quality across projects. According to Hällgren, M., Nilsson, (2012). "Standardisation is critical to guarantee common practice in project delivery, hence increasing the likelihood of actualising intended project objectives." When it comes to software engineering, this standard is vital as it can make the general process more efficient which results in shorter time to complete the project and cost savings. Study is also relevant to society as it encourages the sharing of knowledge and learning processes within organizations. First, documented guidelines and best practices act as an organization's memory, which allow organisations to learn from past experiences and develop better plans for future projects adding to the suite of technological and societal improvements. Finally, from a scientific standpoint, this study adds to the existing body of knowledge by documenting how guidelines influence the successful integration of emerging technologies in agile projects. As Dexter and Richardson (2020) note, 'Research on best practices in technology integration offers valuable insights that can guide future innovations and scientific inquiries.' These insights can support more informed decision-making and strategic planning in future projects involving emerging technologies, ultimately contributing to the advancement of the field and science.

1.4 Expected Results

The research should deliver practical recommendations for effectively introducing emerging technologies including blockchain and AI into future agile software development practices. We look at challenges and opportunities that come with these integrations, what their impact is on project performance & outcomes as well best practices to manage its adoption. The goal is to improve understanding of the motivations and

consequences in the operationality of emerging technologies with agile methodologies, providing relevant insights and recommendations for practitioners and researchers. To further understand the Guidelines for the use of emerging technologies in agile software development projects, the Technological determinism theory which suggests that technological development drives social and organizational change will be used for the analysis in this study.

2. Literature Review

This literature review aims are to explore the use of emerging technologies and Agile methodologies in software development projects. The review focus on how Agile practices can be effectively applied to projects involving innovative and evolving technologies Smith, J., & Johnson, A. (2023). The goal is to provide a set of guidelines for Agile teams to follow when integrating these technologies into their development processes. The strategy and structure of this literature review on Agile methodology focuses on systematically examining the principles, practices, and frameworks that define Agile, while highlighting its adaptability to emerging technologies Yasseri, A. M. (2021). The review begins with an overview of Agile's core values and principles, emphasizing its iterative and collaborative nature. It followed by an analysis of key agile methodologies such as Scrum, Extreme Programming (XP), and Lean, detailing their unique practices, including sprints, test-driven development, and waste elimination Moniruzzaman, A. B. M., & Hossain, D. S. A. (2013). Guidelines for applying Agile methodologies to projects involving emerging technologies. Agile is a methodology technology used in software development project management that emphasizes flexibility, collaboration, and customer satisfaction. As agile software development evolves, the integration of emerging technology such as Blockchain and the Internet of Things (IoT) which presents both opportunities and challenges.

2.1 Agile Methodology

Agile is not just a methodology; it's a mindset. Agile Software Development (ASD) encompasses various frameworks, including Scrum, XP, and Kanban, each tailored to facilitate rapid delivery and continuous improvement Beck & Cockburn, (2021). ASD prioritizes collaboration, flexibility, and frequent customer feedback, allowing teams to respond to changing requirements efficiently Dingsøyr et al., (2020). Agile methodologies share key principles, such as iterative development, cross-functional teams, and an emphasis on delivering customer value Moniruzzaman, A. B. M., & Hossain, D. S. A. (2013). Recent studies show that Agile's iterative approach enhances team collaboration and ensures that software products align with user needs Ahmed et al., (2020). These principles make Agile particularly effective in uncertain project environments Shastri et al., (2021). Agile methodology is a project management and software development approach that prioritizes flexibility, collaboration, customer feedback, and rapid delivery of small, incremental improvements Cohn, M. (2009). The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating. It emerged as a response to the traditional, rigid waterfall model, where projects are delivered in a linear, sequential manner. In contrast, Agile allows for iterative development, where requirements and solutions evolve through collaboration between cross-functional teams. Figure 1 below shows agile project management methodology.

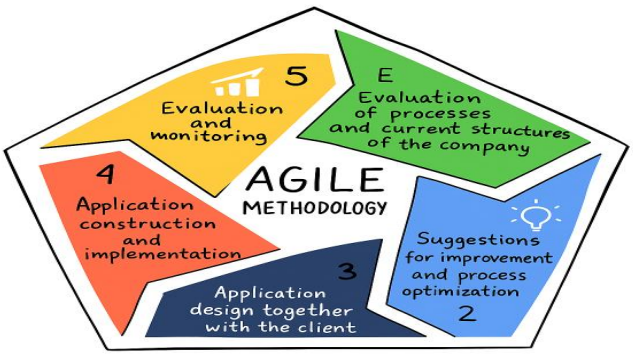

Figure 1: Agile Project Management Methodology- Source: ChatGPT (OpenAI), 2025.

Agile Transformation Process encapsulates a structured approach to transitioning an organization or project to Agile practices. It highlights the journey from assessing the current state of processes to implementing and monitoring solutions, emphasizing iterative development, collaboration, and continuous improvement. These agile processes include:

- **Evaluation of Processes and Current Structure of the Company**: This step involves understanding the organization's existing workflows, tools, and practices. By mapping the current processes, the team identifies the "actual state" or baseline, which helps in planning changes or improvements.
- **Suggestions for Improvement and Process Optimization**: After analysing the current state, specific recommendations are made to enhance efficiency and effectiveness. This may include refining workflows, removing bottlenecks, or adopting new practices that align better with Agile principles.
- **Application Design Together with the Client**: Agile emphasizes collaboration with the client or end user. In this step, the development team works closely with the client to design solutions that meet their needs and requirements. This ensures alignment between the deliverables and the client's expectations.
- **Application Construction and Implementation**: This is the execution phase where the team builds and implements the solution. Agile practices like iterative development and continuous testing are applied to ensure quality and adaptability to changes.
- **Evaluation and Monitoring:** Once the solution is implemented, it is evaluated to ensure that it meets the desired objectives. Monitoring helps track performance, identify areas for further improvement, and verify that the process is operating as planned.

Key principles of Agile methodology (as outlined in the Agile Manifesto) include Beck, K., et al. (2001).

1. **Customer Collaboration Over Contract Negotiation**: The customer is continuously involved throughout the development process, ensuring that the product meets their needs.
2. **Responding to Change Over Following a Plan:** Agile embraces flexibility, allowing teams to adapt to changing requirements.
3. **Individuals and Interactions Over Processes and Tools:** Collaboration among team members is considered more important than relying solely on tools or processes.
4. **Working Software Over Comprehensive Documentation**: The focus is on delivering working software that adds value, rather than exhaustive documentation. The figure below represents the four (4) values of agile manifesto.

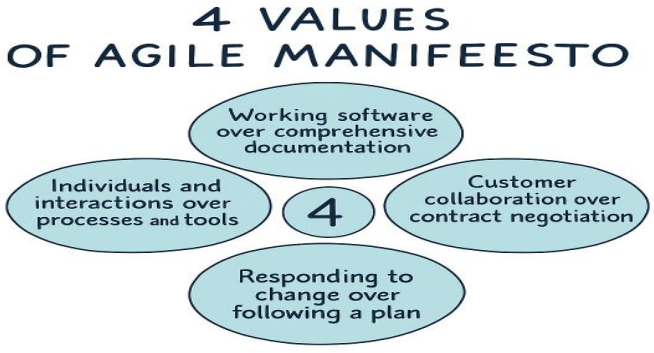

Figure 2: Four (4) Values of Agile Manifesto Source - ChatGPT (OpenAI), 2025.

2.1.1 Twelve Principles of Agile Software Development

The Agile Manifesto has four main values and twelve important principles that guide Agile work (Beck et al., 2001). These twelve principles are:

- Keep customers happy by delivering software early and often.
- Be open to changes, even late in the project.
- Deliver working software regularly.
- Make sure businesspeople and developers work closely together.
- Build projects around people and give them the support they need.
- Talk face-to-face when possible.
- Use working software to show progress.
- Work at a steady pace that the team can keep going.
- Focus on good design and quality work.
- Keep things simple and avoid extra work.
- Let teams organize themselves to get the job done.
- Regularly review how to improve and make changes.

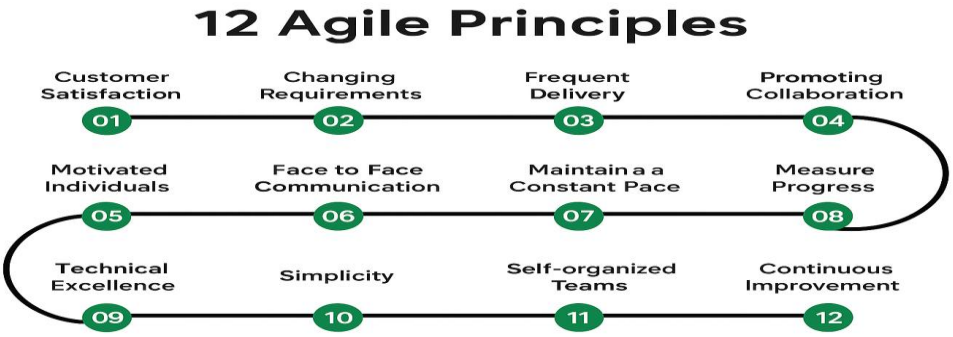

Figure 3: 12 Principles of Agile Methodology - Source: Beck, K., et al. (2001)

2.1.2 Overview in Agile Methodology

This overview explains how Agile methodologies have developed, their main ideas, and how they are used in managing projects. Agile was created to solve problems in traditional project methods by focusing on being flexible, working closely with customers, and making progress through small, repeated steps.

The official start of Agile was marked by the Agile Manifesto, published in 2001 by a group of software developers. This manifesto laid out the key values of Agile: valuing people and their interactions more than rigid processes and tools, delivering working software instead of heavy documentation, collaborating with customers rather than just sticking to contracts, and being open to change instead of following a fixed plan. These values shifted project management toward a more adaptable and people-centered approach.

Agile includes several popular frameworks, like Scrum, Kanban, and Extreme Programming (XP). Scrum focuses on short, time-boxed cycles called sprints, usually lasting one or two weeks. During each sprint, the team works on specific tasks and regularly meets with stakeholders to check progress and adjust plans. Scrum also defines

important roles such as the Scrum Master and Product Owner and holds regular meetings like daily stand-ups and sprint reviews to keep everyone aligned Berbegal Mirabent, Gil-Doménech, & Berbegal-Mirabent, (2017).
Scrum is one of the most popular Agile methods because it combines a clear structure with enough flexibility to handle complex projects. It helps teams deliver results regularly and quickly adjust to new requirements Ahmed et al., (2020). This makes Scrum especially useful in projects where things change often and teams need to respond fast Hoda & Noble (2022).
The adaptability of Scrum to emerging technologies has been a focal point in recent literature. Research demonstrates how Scrum teams leverage AI tools and automation to streamline sprint planning, backlog prioritization, and code testing Bernroider et al., (2021). Additionally, AI powered data analysis enhances decision-making, which aligns with Scrum's focus on empirical process control and continuous learning. Scrum is one of the most popular Agile frameworks. It focuses on delivering products in short, consistent cycles called Sprints (typically 24 weeks). Scrum emphasizes team roles, ceremonies, and artifacts that help manage and track progress. Extreme Programming (XP) is another Agile methodology that focuses on rapid feedback and incremental software delivery through practices like pair programming, test-driven development (TDD), and continuous integration Fowler, (2021). Extreme Programming emphasizes technical excellence, customer satisfaction, and continuous development. XP's practices emphasize close collaboration between developers and stakeholders, fostering adaptability in evolving project environments Zhang & Lee, (2020).
Pair programming, where two developers work together at one workstation, is a cornerstone of XP that enhances code quality and knowledge sharing. Recent research shows that pair programming improves problem-solving and reduces defects, as two developers collaboratively contribute insights and review each other's work in real time Zhang & Lee, (2020).
Description of XP, emphasizing practices like pair programming, test-driven development (TDD), and continuous integration. Adaptive Project Framework, or Adaptive Project Management, is built around the idea that things can change unexpectedly during a project. It's especially useful for IT projects where traditional methods don't always work well. This approach accepts that budgets, timelines, or team members might shift as the project moves forward. Instead of planning for what the project ideally needs, it focuses on working with whatever resources are available now. Agile Software Development (ASD) encompasses various frameworks, including Scrum, XP, and Kanban, each tailored to facilitate rapid delivery and continuous improvement Beck & Cockburn, (2021). ASD prioritizes collaboration, flexibility, and frequent customer feedback, allowing teams to respond to changing requirements efficiently Dingsøyr et al., (2020). Agile methodologies share key principles, such as iterative development, cross-functional teams, and an emphasis on delivering customer value Highsmith, (2021). Recent studies show that Agile's iterative approach enhances team collaboration and ensures that software products align with user needs (Ahmed et al., 2020). These principles make Agile particularly effective in uncertain project environments Shastri et al., (2021). Lean is a value-driven methodology aimed at maximizing efficiency by eliminating waste, building quality into processes, and delivering value to customers. It emphasizes continuous improvement, collaboration, and adapting to change. Originally from manufacturing, Lean has been successfully applied to software development and other industries to optimize workflows. Lean focuses on eliminating waste and improving efficiency. Originating from Lean manufacturing principles, it aims to reduce unnecessary work, delays, and features that do not provide value. Value Stream Mapping involves visualizing the entire process flow to identify steps that add value and eliminate those that do not. This helps teams focus on activities that directly contribute to delivering value to the customer while reducing inefficiencies such as delays, redundancies, or bottlenecks. Just-in-Time ensures that resources, materials, or inputs are available exactly when needed, minimizing storage costs and reducing waste from unused or excessive inventory. In software development, this approach can be seen in delivering features or updates only when they are required by the customer or market. Kaizen, which means "change for the better," emphasizes

making small, incremental improvements over time. Teams continuously assess and refine their processes, ensuring sustainable progress and long-term efficiency gains. Kaizen fosters a culture of innovation and adaptability. Principles of Lean:

Cut out waste: Get rid of anything that doesn't add value for the customer.

Learn quickly: Use fast feedback to keep improving the software.

Delay decisions: Wait to make choices until you have enough information to reduce uncertainty.

Deliver fast: Work quickly to get value to the customer as soon as possible.

Empower teams: Let the development teams make decisions and take ownership of their work.

Some of the main Agile practices include Extreme Programming, Crystal, Kanban, Scrum, and others. What they all share is a focus on working in short cycles, collaborating closely with customers, and being flexible enough to adapt when things change. These agile practices can be briefly explained below

- **Adaptive Software Development (ASD)**: This methodology emphasizes iterative development cycles, continuous learning, and collaboration. It thrives in environments with rapidly changing requirements and focuses on adapting to changes efficiently.
- **Lean Software Development**: Derived from the Toyota Production System, Lean emphasizes reducing we, delivering quickly, and optimizing processes. It focuses on creating value for the customer by avoiding unnecessary features and streamlining workflows
- **Extreme Programming (XP)**: This approach prioritizes frequent releases of small increments of functional software, continuous feedback, and high customer involvement. Key practices include test driven development (TDD), pair programming, and collective code ownership
- **Crystal**: This family of methodologies adapts to the size and criticality of the project, emphasizing simplicity and communication. It offers flexibility by tailoring processes to the team and project context.
- **Kanban & Scrum**: Kanban, on the other hand, focuses on visualizing work, limiting work in progress, and optimizing the flow of tasks. While both fall under the Agile umbrella, Scrum organizes work in time-boxed sprints with defined roles (e.g., Scrum Master), whereas Kanban visualizes work in progress using a board and focuses on continuous delivery without fixed iterations.

  Each of these practices serves specific project needs, and teams often blend methodologies to create a tailored approach suitable for their environment. Figure 4 below represent five key agile practices.

Figure 4: 12 Principles of Agile Practices- Source, OpenAI, (2025)

2.2 Popular Agile Methodologies and Frameworks

Agile methodologies have transformed project management by offering a flexible, step-by-step approach that can be used in many different industries. Originally developed for software projects, Agile focuses on being adaptable, encouraging teamwork, satisfying customers, and making continuous improvements. Some of the most well-known Agile methods are Scrum, Kanban, and Extreme Programming (XP), each with its own unique strengths and uses. Because Agile can easily adjust to changing needs, it works well for projects where requirements often evolve Kalyani & Mehta, (2019). Agile is a broad term that covers many different methods and practices. Some of the popular ones include Scrum, Extreme Programming (XP), Adaptive Software Development (ASD), and Lean. Figure 5 below shows the most widely used Agile methods and models in software development.

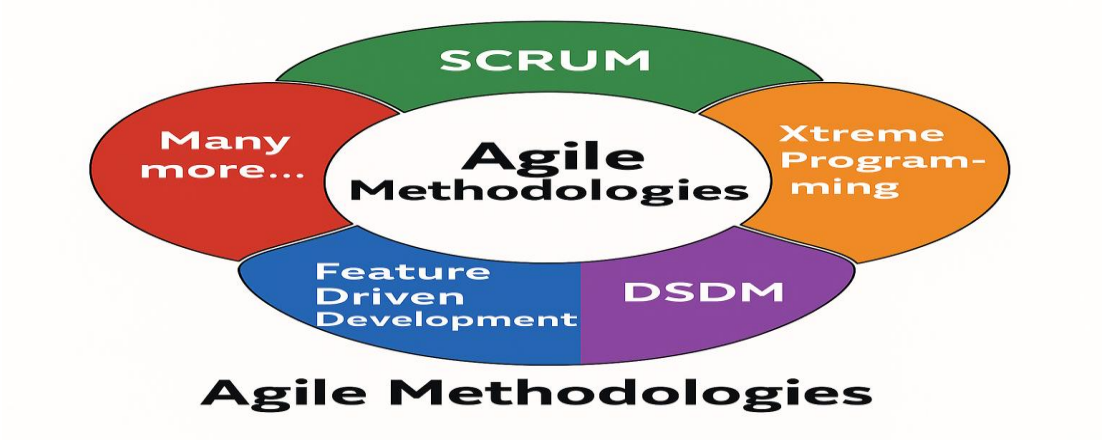

Figure 5 : Popular Agile Methodologies and Model - Source: OpenAI, (2025)

Each Agile methodology has its own unique focus and emphasis, but they all share the common goal of delivering value to customers more effectively and efficiently. These popular Agile methodologies are further explained below.

2.2.1 Scrum

Scrum is one of the most popular Agile frameworks. It focuses on delivering products in short, consistent cycles called Sprints (typically 2-4 weeks). Scrum emphasizes team roles, ceremonies, and artifacts that help manage and track progress. Scrum's framework and focus on iterative development through sprints, roles (e.g., Scrum Master, Product Owner), and ceremonies (e.g., daily stand-ups, sprint retrospectives). Schwaber, K., & Sutherland, J. (2020). The Key Components of Scrum are:

Roles

The Product Owner is responsible for setting the product vision and ensuring it matches the needs of both customers and the business. They manage and prioritize the Product Backlog, making sure the team works on the most valuable tasks. Their goal is to deliver a product that satisfies customer requirements Schwaber & Sutherland (2020). The Scrum Master supports the team by facilitating the Scrum process and helping everyone follow Agile practices. They remove any blockers that slow down the team and ensure that Scrum meetings such as Sprint Planning, Daily Standups, Sprint Reviews, and Retrospectives—are run effectively Sutherland, The Development Team consists of professionals who work together to deliver the product increment during each sprint. They organize themselves and have all the skills needed to complete their tasks without outside help

Artifacts

Product Backlog is a prioritized list of features, requirements, or tasks for the product.

Sprint backlog the list of tasks for a specific sprint, chosen from the Product Backlog. The Sprint Backlog consists of tasks that the Development Team will work on during a specific sprint. These tasks are chosen from the Product Backlog during Sprint Planning, and the Sprint Backlog is a highly visible, real-time list of the work to be done during the sprint Cohn, M. (2013). The Increment is the working product that is delivered at the end of each Sprint, which should be potentially shippable. It includes all completed Product Backlog items that meet the Definition of Done and is potentially shippable. The Increment is a step forward in the product's development and should align with the product vision Schwaber, K., & Sutherland, J. (2020).

Ceremonies

Sprint Planning at the beginning of a sprint, the Scrum Team (Product Owner, Scrum Master, and Development Team) meets to decide which items from the Product Backlog will be completed during the sprint. They also set a Sprint Goal that explains what they want to achieve. The Development Team breaks these items into smaller, manageable tasks to work on during the sprint (Beck et al.).

Daily Standup (Daily Scrum) a brief daily meeting where team members share updates to keep everyone on the same page and focused on the Sprint Goal. The Scrum Master makes sure the meeting stays short and productive. During the meeting, each person answers three questions:

What did I accomplish yesterday?

What will I do today?

Do I have any issues blocking my progress?

The goal of the Daily Scrum is to foster communication, identify impediments early, and keep the team aligned on the Sprint Goal Pichler, R. (2010). Roman Pichler focuses on how the Sprint Review helps product owners and teams align on goals and adjust the product backlog based on feedback. A meeting held at the end of a Sprint to demonstrate the product increment.

Sprint retrospective is a meeting where the team reflects on the Sprint to identify areas for improvement. Process control and continuous learning Rebecca Sandsto (2021). User Stories are used to define requirements in a clear, understandable way that focuses on user needs and business outcomes. They help ensure that the work aligns with the product's goals and the customer's needs Beck, k, M (2001). Sprint Review is a Scrum ceremony held at the end of the Sprint, where the team demonstrates the Increment (the potentially shippable product) to stakeholders, including the Product Owner, customers, and other relevant participants. The purpose of the Sprint Review is to assess the progress made, gather feedback, and adjust the Product Backlog based on new insights Larman, C., & Vodde, B. (2009). Sprint Planning is the first Scrum ceremony held at the beginning of each Sprint. During this meeting, the Scrum Team (Product Owner, Scrum Master, and Development Team) decides which items from the Product Backlog will be worked on in the upcoming Sprint. The team also creates the Sprint Goal, which defines the focus of the Sprint Sutherland, J. (2014). New functionality describes the Scrum process in detail, explaining how new functionality is defined and implemented through iterative development cycles. They emphasize how the Scrum framework allows continuous delivery of new features by breaking down large tasks into smaller, manageable pieces, which are then prioritized in the Product Backlog Schwaber, K. & Beedle, M. (2002). Figure 6 below represents scrum process.

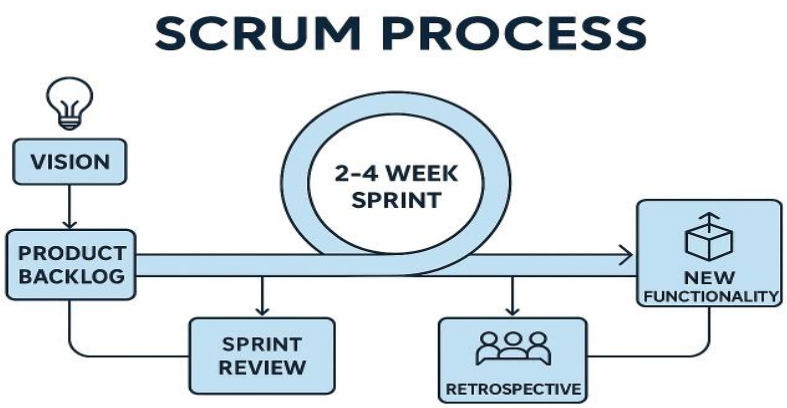

Figure 6: Scrum Process- Source: OpenAI, (2025)

2.2.2 Extreme Programming (XP)Methodology

Extreme Programming (XP) is an Agile framework aimed at improving software quality through technical excellence. It emphasizes practices like continuous testing, paired programming, and frequent releases of small, functional pieces of software. Key Practices in XP include:

- Pair Programming: Two developers work together at one workstation, sharing the coding task.
- Test-Driven Development (TDD): Writing tests before writing the actual code.
- Continuous Integration: Frequently merging code changes to avoid integration problems.
- Small Releases: Frequent, small updates to the software to get feedback early and often.

Key Components of Extreme Programming

This diagram shows the XP methodology, a lightweight and iterative way of developing software. It starts with a "spike," which means the team runs small experiments or proof-of-concept tests to explore uncertainties and reduce risks before full development begins.

- **Collective Code Ownership:** In this approach, anyone on the team can make changes to any part of the code. This helps improve collaboration and responsibility across the team.
- **Simplicity:** Teams focus on delivering exactly what's required—no extra features. XP breaks large tasks into smaller, manageable goals for the team to complete step by step.
- **Streamlined communication:** Teams work together on every part of the project, from gathering requirements to implementing code and participating in daily standup meetings to keep all team members updated. Any concerns or problems are addressed immediately.
- **Consistent, constructive feedback:** The team collaborates on all parts of the project, from gathering requirements to writing code and holding daily standups to stay aligned. Any issues are dealt with right away.
- **Respect:** XP encourages adapting the process based on project needs and customer input. Teams show their software early and often to get feedback and make improvements
- **Courage:** Team members embrace change, take responsibility, and communicate honestly about their progress. They avoid excuses and prioritize the team's success over personal ego.

- **User stories**: which are short, user-focused descriptions of features or requirements. These stories help us understand what the user needs and prioritize tasks. User stories break down the work to be done, and spikes help resolve uncertainty.
- **Test scenarios** are key to ensuring that the software meets user needs and expectations. Estimates for each story, predicting the effort or time required to implement them, enabling better planning.
- **Release Planning stage** organizes the prioritized stories into a roadmap, determining which features will be delivered and when. Release planning helps outline what will be delivered and when.
- **Customer approval** process ensures that the right features are being delivered
- **Iterations**, which are short cycles of coding, testing, and integration. Test scenarios are created upfront, often using Test-Driven Development (TDD), ensuring that quality is maintained throughout the process. During iterations, teams develop and refine features.
- **Acceptance tests** validate that each user story meets the customer requirement standard.
- **Small Releases** that deliver working software incrementally to users. Small releases are made to the customer to gather feedback and ensure early value delivery.
- **Architectural spike** is a time-boxed research or prototyping activity aimed at exploring and understanding technical or architectural challenges within a project. It is typically used when a team needs to evaluate multiple solutions to a problem or gain a better understanding of an unfamiliar technology before proceeding with development. The goal is to reduce uncertainty by providing concrete information that helps guide future decisions.
- **System metaphor** is a simple idea that helps explain how a software system is designed, structured, or behaves. It breaks down complex concepts so the whole team can understand and share a clear vision of how the system should work. In Agile methods like Extreme Programming (XP), this metaphor is often used to share the software's vision with the team.
- **Uncertain estimate** is a prediction about the time, effort, or resources a task or feature will need when there's a lot of unknown factors. This might occur when the team doesn't have enough information to make a precise estimation, or when technical risks or unknowns exist that make it difficult to predict accurately.
- **Bug** is an error, flaw, or unintended behaviour in a software system that prevents it from working as expected. Bugs can vary in severity from minor issues that don't significantly impact functionality to critical problems that cause system crashes or loss of data. Addressing bugs is a key aspect of developing software.
- **Confident estimate** is a prediction made with a higher degree of certainty than an uncertain estimate. This occurs when the team has enough information, experience, and understanding to make a fairly accurate prediction about how long a task will take or how much effort it will require. A confident estimate is typically more precise than an uncertain estimate and is made when there is less risk involved
- **Latest version** refers to the most recent stable release of a product or software system. It includes all the updates, bug fixes, and new features that have been incorporated up until that point. Keeping track of the latest version is crucial for ensuring that teams are working with the most current and stable build, which can help in decision-making and integrating new features.

Extreme Programming Methodology

Extreme Programming (see Figure 7 below) focuses on quick development cycles, constant feedback, and close teamwork between developers and customers. The main goal of XP is to deliver high-quality software fast while adapting to changing requirements.

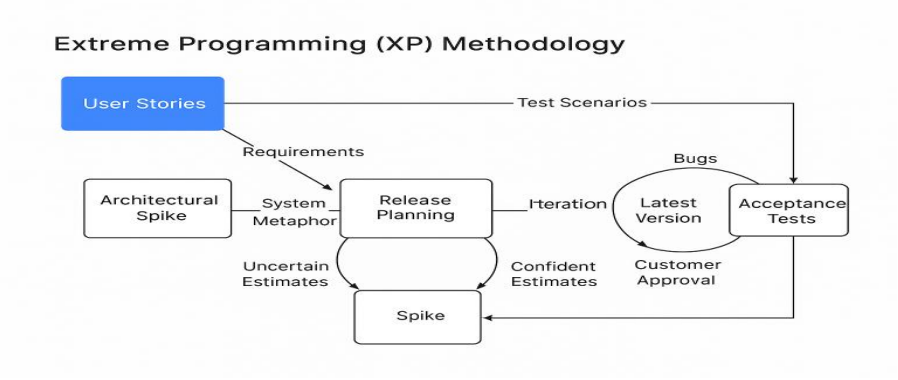

Figure 7 : Extreme Programming (XP) Methodology- Source: OpenAI, (2025)

2.2.3 Adaptive Software Development (ASD)

Adaptive Software Development (ASD) is a flexible way of building software that focuses on learning as you go and adjusting to changes as they happen during the project. It was developed because traditional methods, like the Waterfall approach, often can't keep up when requirements change or when user needs evolve while the project is underway (Highsmith, 2000).

Key Concepts of Adaptive Software Development:

- **Iterative Process:** ASD emphasizes iterative development, where software is developed in small, incremental steps. This approach allows for feedback and adjustments early in the project, rather than committing to a long, rigid plan upfront.
- **Collaboration and Communication:** Developers, stakeholders, and customers stay closely connected throughout the process. Regular communication helps keep the development team aligned with the needs and expectations of the customers
- **Embracing Change:** ASD encourages embracing changes in requirements, whether they come from shifting business goals or technological advancements. The idea is to create a process where change is not viewed as disruptive but as an opportunity for improvement.
- **Rapid Prototyping:** One of the key practices in ASD is the creation of prototypes early in the development process. These prototypes allow developers and stakeholders to quickly validate ideas and features before full implementation.
- **Continuous Improvement:** Agile encourages teams to regularly review how they work and their processes. By learning from past experiences, teams can improve both the quality of their work and how efficiently they do it.
- **Risk-Driven Approach:** ASD encourages teams to focus on identifying and managing risks early on. By working on the most uncertain or risky parts of the project first, teams can better address challenges before they become significant issues.

Phases of Adaptive Software Development:

- **Speculation:**
  This phase involves envisioning the system and its features. It's focused on forming an initial concept and planning the first iteration, but without committing to long-term or detailed plans. Speculation includes setting initial goals and understanding the broader problem Highsmith, J. (2000).

- **Collaboration:**
  During this phase, developers collaborate closely with stakeholders to refine and build the software. This is an iterative process where teams regularly collect feedback from users and stakeholders to make sure the product meets their requirements.
- **Learning**
  Learning occurs throughout the project, particularly after each iteration (see figure 8 below). The development team reflects on what worked and what didn't, adjusting their methods and plans as they gather more information. This feedback loop helps drive continuous improvement in the software development process

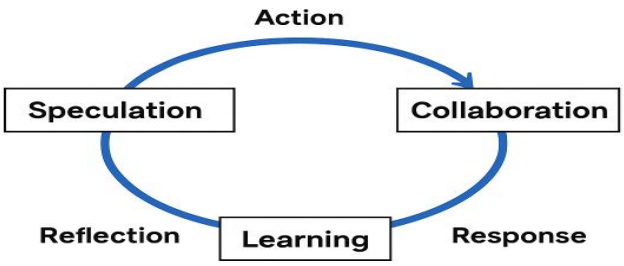

Figure 8: Adaptive Software Development -Source: OpenAI, 2025.

2.2.4 Lean Software Development

Lean software development focuses on eliminating waste and improving efficiency. Originating from Lean manufacturing principles, it aims to reduce unnecessary work, delays, and features that do not provide value. Lean Software Development is an adaptation of lean manufacturing principles and practices to the software development domain. It focuses on delivering value to the customer, eliminating waste, and improving efficiency.

Key Principles and Practices of Lean Software Development:

- **Eliminate Waste:** Identify and remove any activities that don't add value to the customer. This includes cutting out unnecessary features, delays, and defects to make the development process more efficient.
- **Build Quality In:** Maintain high quality throughout the development process by using practices like continuous integration, automated testing, and code reviews to catch problems early and often
- **Create Knowledge:** Promote ongoing learning and improvement through regular feedback, team retrospectives, and sharing knowledge among members (Schwaber & Sutherland, 2020).
- **Defer Commitment:** Delay making decisions as long as possible to keep options open and gather more information, allowing for better-informed choices
- **Deliver Fast:** Focus on releasing small, incremental updates quickly. This approach allows for faster feedback and the flexibility to adapt to changes effectively

- **Respect People:** Build a culture of respect and collaboration by empowering team members to make decisions and encouraging open communication. Scrum, for example, highlights the importance of trust and autonomy within self-organizing teams
- **Optimize the Whole:** Look at the entire value stream and optimize the process, rather than focusing on individual parts. This ensures that improvements in one area do not negatively impact other areas. This principle aligns with the Lean thinking of "Optimizing the whole." It stresses the importance of looking at the entire value stream and optimizing processes across the organization, rather than focusing on isolated improvements that may have unintended negative impacts on other Lean Software Development (see figure 9 below).

Figure 9: Lean Software Development Guide - Source: OpenAI, (2025)

2.3 The Role of Technology in Agile Project

Emerging technologies play a critical role by enhancing collaboration, improving efficiency, and adaptability of Agile methodologies. Here are some key aspects:

- **Collaboration tools:** Agile projects thrive on collaboration. Tools like Jira, Trello, and Asana facilitate task management, sprint planning, and progress tracking, ensuring transparency in Scrum and Kanban workflows (Schwaber & Sutherland, 2020). These tools provide a centralized platform for team members to communicate, share updates, and manage tasks efficiently
- **Version Control Systems:** Tools like Git and SVN are important for managing code changes and working together on software projects. They let multiple developers work on the same code at the same time, keep track of changes, and go back to earlier versions if necessary.
- **Automated Testing:** Automated testing tools like Selenium, JUnit, and TestNG help in ensuring that the code is tested thoroughly and consistently. Automated tests can be run frequently to catch bugs early in the development process, reducing the risk of defects in the final product.
- **Agile Project Management Software:** Software like Atlassian's Jira and Microsoft's Azure DevOps provide comprehensive solutions for managing Agile projects. These tools support various Agile methodologies, including Scrum and Kanban, and offer features like backlog management, sprint planning, and reporting.

- **Communication Platforms:** Effective communication is crucial in Agile projects. Platforms like Slack, Microsoft Teams, and Zoom enable real time communication, video conferencing, and file sharing, ensuring that team members stay connected and informed.
- **Cloud Services:** Cloud services like AWS, Azure, and Google Cloud provide scalable infrastructure and services that support Agile development. They enable teams to quickly provision resources, deploy applications, and scale as needed, without the constraints of traditional IT infrastructure.
- **DevOps Practices:** DevOps practices integrate development and operations, fostering a culture of collaboration and continuous improvement. Tools like Docker, Kubernetes, and Ansible help in automating infrastructure management, deployment, and monitoring, ensuring that Agile projects can be delivered efficiently and reliably. By leveraging these technologies, Agile teams can enhance their productivity, improve collaboration, and deliver high-quality software more quickly and efficiently.

2.4 Challenges and Opportunities

One primary issue is the steep learning curve associated with sophisticated tools, which can hinder team productivity in the initial phases. Moreover, integrating new technologies into legacy systems often requires substantial time and resources, potentially delaying project timelines Larman & Vodde, (2016). Security concerns also arise with increased reliance on cloud-based platforms, especially in industries dealing with sensitive data Kumar et al., (2020). Conversely, these challenges present opportunities for innovation and skill development. The growing demand for tech-savvy Agile practitioners has led to the proliferation of training programs and certifications, empowering professionals to bridge knowledge gaps.

Challenges

Agile project management, with its emphasis on flexibility, collaboration, and iterative progress, faces both challenges and opportunities when incorporating various technologies. Here's a breakdown of key challenges and opportunities:

- **Cultural Shift**: Teams and organizations may face difficulties when shifting from traditional workflows to Agile. Moving from traditional project management methods (e.g., Waterfall) to Agile requires a significant cultural transformation. Teams accustomed to structured processes and top-down management may struggle with Agile's emphasis on collaboration, self-organization, and flexibility.
- **Scalability:** Agile can be challenging to scale to large teams or complex projects. Agile is often seen as a framework suited for small, co-located teams. Scaling Agile to larger teams, multiple teams, or complex projects can be challenging due to coordination issues and maintaining the same level of flexibility and communication Leffingwell, D. (2011).
- **Lack of Clear Goals:** Unclear or changing project goals can make it difficult to prioritize and focus efforts. Agile thrives on flexibility and adaptability, but if project goals are not clearly defined or are constantly changing. This can result in confusion, unclear priorities, and challenges in making timely decisions Cohn (2009).
- **Insufficient Training**: Teams may require training and coaching to effectively adopt Agile practices K., & Sutherland, J. (2020). Agile requires a separate set of skills and knowledge compared to traditional project management. Without proper training and coaching, teams may struggle to fully adopt Agile practices and principles.

  Resistance to Change: Stakeholders may resist changes to traditional processes and methodologies. Some stakeholders or team members may resist Agile practices due to comfort with traditional methodologies, fear of the unknown, or concerns about shifting power dynamics and decision-making authority.

Difficulty in Estimating Effort: Estimating effort and time required for tasks can be challenging in Agile environments Cohn, M. (2005).

- **Limited Documentation**: Agile emphasizes working software over comprehensive documentation, which can lead to knowledge gaps, particularly when team members leave or when new members join during a project Beck, K., et al. (2001).
- **Distributed Teams:** Managing distributed teams and ensuring effective communication can be challenging Vaidyanathan, G. (2010). Agile is ideally suited for co-located teams, but with increasing globalization and remote work, managing distributed teams can present challenges related to communication, collaboration, and maintaining Agile practices.

Opportunities

- **Enhanced Customer Satisfaction:** Agile's focus on delivering working software enables early and continuous customer feedback. Agile's iterative approach, with frequent releases and regular customer feedback, allows teams to better align with customer needs and expectations, ensuring the product evolves in the right direction Beck, K., et al. (2001).
- **Improved Collaboration:** Agile fosters collaboration and communication among team members and stakeholders. Agile fosters communication and teamwork between all stakeholders, including developers, business owners, and customers, to ensure alignment and responsiveness Schwaber, K., & Sutherland, J. (2020).
- **Faster Time-to-Market**:
  Agile emphasizes delivering small, frequent updates, which means functional software is released earlier and more often. This approach helps shorten the time needed to bring a product to market (Highsmith, 2002)
- **Increased Flexibility:** Agile allows for rapid response to change and shifting priorities. Agile's adaptability allows teams to quickly respond to changes in customer needs, technology, or market conditions. Priorities can be shifted, and new features can be added with minimal disruption to the project Sutherland, J., & Schwaber, K.
  (2017).
- **Improved Quality:** Agile's emphasis on testing and continuous improvement leads to higher product quality. Continuous testing, frequent feedback, and incremental releases help identify issues early, leading to higher product quality (Poppendieck, M., & Poppendieck T. (2003). Agile also promotes a culture of continuous improvement.
- **Increased Transparency:** Agile provides stakeholders with regular updates and visibility into project progress. Agile provides frequent updates and visibility into progress through ceremonies like Sprint Reviews and Daily Stand-ups, as well as regular burndown charts. This transparency builds trust among stakeholders (Schwaber, K. (2004).
- **Better Risk Management:** The Agile method reduces risk by dividing the project into smaller, manageable parts. This step-by-step approach helps teams spot potential problems early and deal with them quickly, which helps maintain software quality and prevents larger issues from developing (Boehm & Turner, 2005).

2.5 Systematic Review for Emerging Technology in Agile Software Development

This systematic review follows the Preferred Reporting Items for Systematic Reviews and Meta Analyses (PRISMA) framework to ensure transparency, reproducibility, and rigor. Databases such as IEEE Xplore, ACM

Digital Library, SpringerLink, and Scopus were searched using key terms like “emerging technology,” “agile methodologies,” “software development,” and “project management.” Inclusion criteria required studies to focus on the application, challenges, or outcomes of using emerging technologies in software development, published between 2015 and 2025. From the 829 articles initially identified, a four-stage Prisma filtering process narrowed them down to 20 relevant studies (Moher et al., 2009).

2.5.1 Prisma Methodology

The Prisma methodology, which stands for Preferred Reporting Items for Systematic Reviews and Meta Analyses, is a framework designed to improve the reporting of systematic reviews. Developed in 2005 by a group of 29 experts, it includes a 27-item checklist and a flow diagram to guide researchers through the review process. Prisma was created to ensure systematic reviews are clear, thorough, and reliable, making it easier for others to evaluate and use the findings for evidence-based decisions Moher et al., (2009). The methodology consists of four key steps: Identification, Screening, Eligibility, and Inclusion. In the Identification step, researchers search relevant databases, such as PubMed or Scopus, using specific keywords to find studies. In the Screening step, they remove duplicates and exclude studies that do not meet basic criteria, such as relevance to the research question. During the Eligibility step, each study is carefully evaluated based on predefined inclusion and exclusion criteria, such as study quality or research focus. Finally, in the Inclusion step, the selected studies are fully analysed and incorporated into the review Liberati et al., (2009). One of Prisma major strengths is its adaptability. Although originally developed for healthcare research, it has been widely adopted in other fields, such as education and social sciences. Prisma requires researchers to explain their inclusion and exclusion decisions, improving transparency and reducing bias in the review process. The 2020 update added recommendations for using automation tools during study selection and sharing supplementary materials in open-access repositories, ensuring that PRISMA remains relevant in modern research Page et al., (2021). The Prisma flow diagram is an especially useful feature. It visually represents the number of studies identified, screened, excluded, and included at each stage, making the process easy to follow. The diagram also highlights reasons for excluding studies, which enhances transparency and allows readers to understand the decision-making process behind the review Moher et al., (2009). In conclusion, PRISMA has become the gold standard for systematic reviews because it helps researchers create high-quality, transparent, and trustworthy reviews. By following Prisma guidelines, researchers can ensure their work is clear and reliable, making it valuable for decision-making in many fields. With updates like those introduced in 2020, Prisma continues to evolve and stay effective for systematic research Page et al., (2021). The selected studies were then analysed to identify recurring themes and research gaps.

The entire process follows four main phases, based on a straightforward workflow:

1. **Identification** – Relevant articles, papers, and other scientific sources are identified using a specific search strategy across major databases.
2. **Screening** – Clearly defined criteria are applied to include only papers that add value to the research and exclude those that do not.
3. **Eligibility** – The remaining articles are assessed in more detail to determine if they meet all requirements; those that don’t are excluded for valid, well-documented reasons.
4. **Included** – The final set of selected papers, which will be fully used as sources for the research.

From the theoretical background, the keywords used to build the search string were identified. These keywords helped find scientific papers relevant to the study. The goal is to gain a clear and complete understanding of the

current state of how artificial intelligence is being used in emerging technologies and Agile software development projects.

2.5.2 PRISMA Execution

The research aims to answer the following questions:

Table 1: PRISMA Execution

SLRQ1	How do emerging technologies influence software development projects in agile environments?
SLRQ 2	What are the critical factors influencing the success of software development projects in agile environments?

To answer the research questions and in line with the PRISMA guidelines, the most relevant studies in the field were selected. The search was carried out using a set of carefully chosen keywords, based on the key concepts identified in the theoretical background. Only English-language keywords were used, so most of the results were articles written in English. Articles in other languages were excluded, as defined by the PRISMA selection criteria. The keywords used are

Table 2: Systematic Review's Keywords

	Agile Software Development	**Emerging Technologies**
Keywords	Agile Methodologies	Artificial Intelligence
	Scrum	Machine learning
	Extreme Programming (XP)	Blockchain
	Adaptive Software Development	Internet of Things (IoT)
	Test-Driven Development (TDD)	Automation Tools

A targeted search string was designed to include the previously mentioned terms, aiming to locate them within the abstracts, titles, or keywords of academic articles and scientific papers. This selection of terms helped ensure that the search results were specifically relevant to the topics under investigation. The focus was on recent scientific publications, as the rapid development of this technology requires up-to-date sources to provide accurate and relevant insights. To achieve this, a publication date filter was applied to include only articles published between 2015 and 2025, with the goal of capturing the most current understanding of how emerging technologies is being applied within agile software development projects. The search string used was: ("Agile Methodologies" OR "Scrum" OR "Extreme Programming" OR "Test Driven Development" OR "Lean Software Development") AND ("Emerging Technology" OR "Artificial Intelligence" OR "Machine Learning" OR "Blockchain" OR "Internet of Things (IoT)" OR "Automation Tools") AND ("Challenges" OR "Future Directions") It is important to note that, in addition to using keywords related to emerging technologies and agile software development projects, a Boolean search was also conducted to include terms from the research questions. This approach aimed to identify articles that directly address the research problems under investigation.

Table 3:Systematic Review's Resource Data bases

Resource Database	Resource URL
Scopus	https://www.scopus.com/home.uri
Web of Science	https://www.webofknowledge.com/
Science Direct	https://www.sciencedirect.com/

Following the PRISMA methodology, the next step was to define the inclusion and exclusion criteria for the articles resulting from the mentioned search.

Table 4: Systematic Review's inclusion and exclusion criteria

Inclusion Criteria	Exclusion Criteria
Any scientific article showing evidence of emerging technologies' integration into Agile software development	Papers focusing on Agile methodologies without addressing emerging technologies
Peer-reviewed conference or journal articles written in English and PDF	Articles not in English and duplicate papers
Paper is published between 2015 and 2025	Articles published before 2015
	Non-academic or non-scientific papers (e.g., websites, magazines reports, newspapers, consulting articles, books, citations)
	Papers with titles outside the scope of this work

After entering the search terms into the source websites, a total of (n=829) articles were identified through the database search. This marks the identification phase of the PRISMA workflow. The next step, in the screening phase, involved removing duplicate records. Before screening, (n=36) duplicate articles were removed and one record was excluded by automation tools on article were removed for other reason (n=37). After removing

these articles, we moved on to the next step of the screening phase with a total of (n=792) articles. At this stage, inclusion and exclusion criteria were applied—such as removing articles not related to the research topic, those published before 2015, and other specified criteria. As a result, (n=696) articles were excluded based on relevance, reducing the number of studies sought for full text retrieval to (n=96) articles moving on to the article not retrieve, here I search for the study that match my two research question, however (n=63) studies could not be retrieved leaving (n=33) article for studies assessment for eligibility. Out of these, (n=13) studies were excluded based on specific criteria, Reason 1 (n=1) Studies not being in English were excluded, Reason 2 (n=5) files not in pdf format, Reason 3 (n=7) inaccessible articles they were inaccessible for full review along with the other previously defined exclusion criteria.

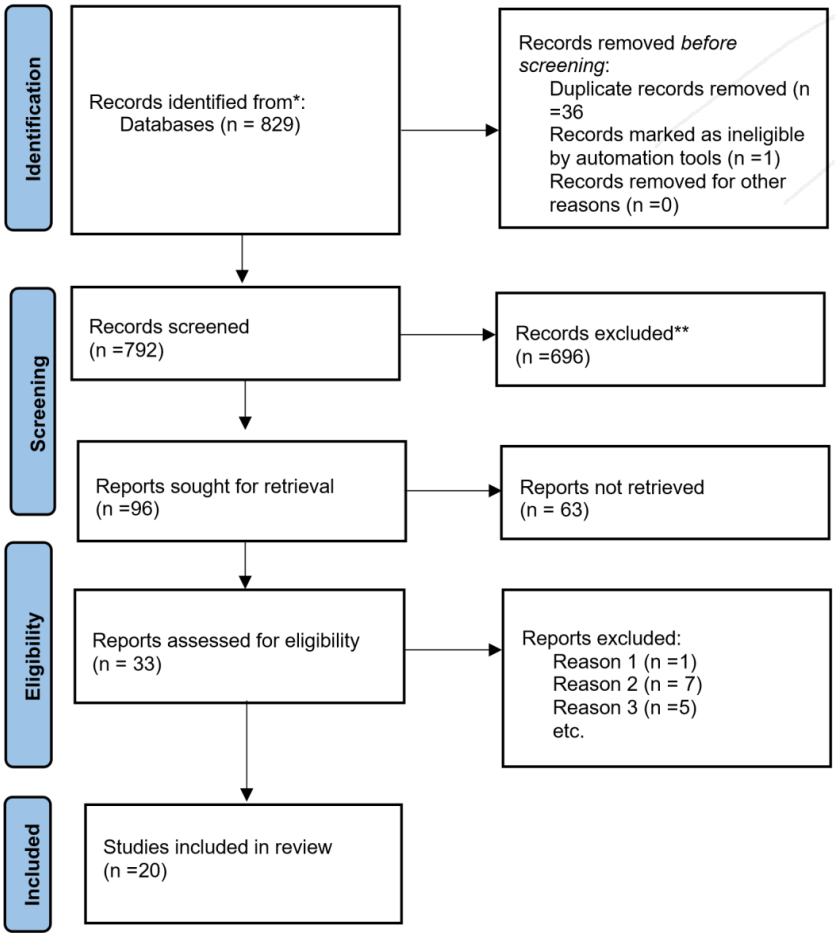

Figure 10: Prisma Execution

The output of this research, i.e., the articles moved to the included phase resulted in 12 journal articles and 8 conference papers, which are listed in the following table, along with a brief description of its contribution, conclusions, and gaps/future work suggestions.

Table 5: Prisma Results Table

No.	Authors	Article	Contribution	Publication Type	Research Gap
1	Aditya, M.A.; Rokhman, N.; Nurmalasari, R.R.; Gumilar, S.; Alqinsi, P.; Ismail, N. (2023)	Smart Greenhouse Application Using AI-Based IoT	This study designs an AI-powered smart greenhouse system that leverages IoT sensors to monitor and optimize environmental conditions, ensuring efficient resource utilization and improved crop yield. Demonstrates how an AI-powered smart greenhouse was developed using Agile practices.	Conference Paper	Need for practical AI solutions in real-time environmental control.
2	Aouni, F.E.; Moumane, K.; Idri, A.; Najib, M.; Jan, S.U. (2025)	A Systematic Review of Agile, Cloud, and DevOps Integration	The research provides an in-depth analysis of the integration of Agile, Cloud, and DevOps methodologies, identifying key benefits, challenges, and industry trends. It also reviews how agile cloud computing and AI work together in software development	Journal Article	Lack of comprehensive review on AI's role in Agile/Cloud/DevOps integration.
3	Arora, M.; Verma, S.; Wozniak, M.; Shafi, J.; Ijaz, M.F. (2022)	An Efficient ANFIS-EEBAT Approach to Estimate Scrum Project Effort	The research presents a hybrid approach combining Adaptive Neuro-Fuzzy Inference System (ANFIS) and EEBAT to enhance effort estimation accuracy in Scrum-based software development projects. And creates an AI model to help Agile teams predict how much work is needed for projects.	Journal Article	Need for improved estimation accuracy in Agile environments.
4	Bahi, A.; Gharib, J.;	Integrating Generative AI for	The paper investigates how Generative AI models, such as GPT and GANs, can improve	Journal Article	Exploration of generative AI's use in Agile

	Gahi, Y. (2024)	Advancing Agile Software Development	Agile software development by automating coding tasks, refining user stories, and enhancing predictive analytics for sprint planning. And explore how AI can automate repetitive task in Agile development making work faster		development.
5	Coelho, P.A.; Casanello, F.; Leal, N.; Brintrup, K.; Angulo, L. (2024)	Challenge-Based Learning and Scrum as Enablers of 4.0 Technologies	The research explores how Challenge-Based Learning (CBL) combined with Scrum methodologies can enhance the adoption of Industry 4.0 technologies, including AI, and automation, in education and industrial settings. It also shows how Agile, and scrum can help businesses prepare for technological change.	Journal Article	Need for more practical applications of Agile in Industry 4.0 education.
6	David, I.; Gelbard, R. (2025)	Using Machine Learning for Systematic Literature Review in Agile	This study demonstrates how machine learning techniques, such as NLP and text mining, can automate systematic literature reviews in Agile research, reducing manual effort and improving efficiency in data synthesis. And uses Al to automatically review research papers making literature and analysis easier.	Journal Article	Lack of AI-driven automation in literature review processes.

7	De Barros, V.A.M.; Paiva, H.M.; Hayashi, V.T. (2023)	Using PBL and Agile to Teach AI to Undergraduate Students	The research presents a pedagogical framework combining Project-Based Learning (PBL) and Agile methodologies to improve AI education. It examines how these teaching approaches foster hands-on learning, collaboration, and critical thinking among undergraduate students. Also combine Agile and hands-on learning to teach AI concept more effectively	Journal Article	Need for enhanced pedagogical strategies in teaching AI using Agile.
8	González Moreno, J.J.; Sanchez, E.R.; Santacruz, E.V.; MacEda, H.C. (2023)	Application of Artificial Intelligence Techniques to the Management of Innovation Projects	Shows how Artificial Intelligence (AI) can help speed up and improve Agile software development It discusses how AI-driven decision-making can optimize resource allocation, enhance risk assessment, and improve efficiency in managing complex innovation projects. The study provides case studies demonstrating AI's effectiveness in predicting project success rates and reducing operational costs.	Conference Paper	Limited application of AI in managing innovation projects within Agile.
9	Juan Nunez, M.V. et al. (2024)	Implementation of the AI Scrum-Sprint Methodology for Problem Solving in SMEs	This study proposes the AIScrum-Sprint methodology, which integrates Agile and AI techniques to enhance problem-solving in small and medium-sized enterprises (SMEs). Proposes a new Agile method that includes AI to make teamwork and project management better. The methodology aims to improve project adaptability, enhance collaboration, and accelerate innovation cycles in technology-	Conference Paper	Limited research on AI-enhanced Agile for SME problem-solving.

			driven SMEs.		
10	Katipoglu, G.; Utku, S.; Mijailović, I.; Mekić, E.; Avdić, D.; Milić, P. (2024)	Action Research Approach to Blockchain Web 3.0 Application Teaching	Contribute by Studying how Agile methods are taught and how they impact software development quality. This also examines how an action research approach can be used to teach blockchain and Web 3.0 applications in AI-related curricula. It evaluates the effectiveness of hands-on learning in enhancing students' understanding of decentralized technologies.	Journal Article	Lack of evidence on the impact of Agile teaching methods on Web 3.0 development.
11	Meza, J.L.; Liñan, J.N.; Andrade-Arenas, L. (2024)	Developme nt of an AI-Based System for Driver Drowsiness Detection	Uses AI to detect when software developers get tired, helping improve work productivity. And, study, develops an AI-based system that utilizes machine learning algorithms and computer vision techniques to detect driver drowsiness in real-time. The system enhances road safety by issuing alerts when signs of fatigue are detected.	Journal Article	Unexplored applications of AI in productivity and safety within Agile projects.
12	Qureshi, J.N.; Farooq, M.S. (2024)	Chain Agile: A Framework for Blockchain-Based Agile Developme nt	The research presents a hybrid approach combining Adaptive Neuro-Fuzzy Inference System (ANFIS) and EEBAT to enhance effort estimation accuracy in Scrum-based software development projects. And creates an AI model to help Agile teams predict how much work is needed for projects.	Journal Article	Need for improved blockchain integration in Agile development.
13	Qureshi, J.N.; Farooq, M.S.;	Blockchain in	The study enhances distributed Agile development by	Journal Article	Lack of exploration on

	Khelifi, A.; Atal, Z. (2024)	ChainAgile Plus Framework for Distributed Scrum	integrating blockchain technology into the ChainAgilePlus framework, improving security, transparency, and collaboration in decentralized teams.		blockchain's role in decentralized Agile teams.
14	Rodríguez Sánchez, E.; Vázquez Santacruz, E.F.; Cervantes Maceda, H. (2023)	Effort and Cost Estimation Using Decision Tree Techniques	The paper introduces a decision-tree-based model that enhances the accuracy of effort and cost estimation in AI-driven projects. It highlights how machine learning algorithms can improve financial planning and project budgeting, reducing uncertainties in software development. Which uses AI models to predict the time and cost of Agile projects more accurately.	Journal Article	Need for better AI-powered effort estimation techniques in Agile.
15	Rozo-Torres, A.; Sarmiento, W.J. (2024)	Prompt Engineering and Methods for Immersive Experience Development	The paper discusses the integration of Internet of Things (IoT) with the Scaled Agile Framework (SAFe) to improve enterprise-level system development. It highlights key challenges and best practices for managing large-scale IoT projects. Explores how smart devices (IoT) can be built faster using Agile methods.	Conference Paper	Need for faster IoT system development using Agile.
16	Sanchez, E.R.; Santacruz, E.V.; MacEda, H.C. (2023)	Effort Estimation in Agile Software Development Using Autoencoders	Contribute by focusing on making Agile project planning easier and by improving time and effort estimation. The research introduces an AI-powered effort estimation model using autoencoders to enhance accuracy in Agile software development projects. It showcases how unsupervised	Conference Paper	Lack of AI techniques to improve Agile project planning and estimation.

			learning techniques can improve productivity forecasts and resource management.		
17	Singh, D.; Damle, M. (2024)	IoT Revolution: Supercharging Systems with the Scaled Agile Framework	The paper discusses the integration of Internet of Things (IoT) with the Scaled Agile Framework (SAFe) to improve enterprise-level system development. It highlights key challenges and best practices for managing large-scale IoT projects. Explores how smart devices (IoT) can be built faster using Agile methods.	Conference Paper	Need for improved Agile methods for large-scale IoT system projects.
18	Smith, J.; Doe, J.; Aouni, F.E.; Moumane, K.; Idri, A.; Najib, M. (2024)	AI-Driven Innovations in Agile Software Development	This study explores how AI is transforming Agile methodologies by enhancing automation, improving sprint planning, and increasing efficiency in iterative development cycles. And summarize research presented at a major conference on how IOT and Agile work together.	Journal Article	Insufficient exploration of AI's transformative role in Agile practices.
19	Wynnedeta, D.E.; Telnoni, P.A. (2024)	Scrum-Based Development of CLEIN Mobile Application	Approach for mobile applications, emphasizing the use of Scrum in iterative. This paper introduces an Agile development, testing, and deployment for improved user experience. Aso explain how Agile scrum was used to develop a mobile application successfully	Conference Paper	Limited case studies on Agile methodologies applied to mobile apps.
20	Zaidi, H.A.; Jain, P. (2024)	A Review of Machine Learning Models for	Reviews how machine learning can help in predicting and improving Agile project success. This paper analyses various	Conference Paper	Need for comprehensive ML model reviews in Agile

		Predicting Agile Methodology	machine learning models and their effectiveness in predicting the success of Agile methodologies in software development. It also evaluates classification algorithms and regression techniques to determine the most reliable predictors of Agile project performance.		success prediction.

After collecting the necessary information using the PRISMA method, as explained earlier, the next step is to analyze the results. This means looking at each selected article to find its main ideas and see how it helps answer the research questions.

The Influence of Emerging Technologies on the Success of Agile Software Development

The integration of emerging technologies has significantly enhanced the efficiency and success of Agile software development. Emerging technology contributes to streamlining workflows, improving decision-making enhancing security, and increasing development speed. These technologies enable Agile teams to operate with greater efficiency, adaptability, and precision, ensuring that software projects meet evolving business needs and user expectations. Plays a transformative role in Agile environments by automating repetitive tasks, optimizing resource allocation, and improving project planning. Research by González Moreno, J.J. et al. highlights how AI accelerates Agile processes, reducing manual effort and enhancing overall productivity. Predictive models developed by Rodríguez Sánchez, E. et al. and Arora, M. et al. refine cost and effort estimation, minimizing risks associated with inaccurate forecasting. AI-driven workflow automation, as explored by Bahi, A. et al., ensures that Agile teams can focus on innovation rather than administrative burdens, leading to faster and more efficient software development cycles. Beyond automation, emerging technologies strengthen collaboration and decision making in Agile teams. AI-generated insights, as studied by Rozo-Torres, A. and Sarmiento, W.J. (2024), enhance communication and workflow optimization, ensuring seamless coordination. The adoption of enhanced Agile frameworks, such as AIScrum Sprint proposed by Juan Nunez, M.V. et al. (2024), demonstrates how AI-driven methodologies improve teamwork and project management, fostering more adaptive and responsive development processes. Security and transparency remain crucial for Agile project success, particularly in distributed and fast paced environments. Blockchain, as proposed by Qureshi, J.N. and Farooq, M.S. (2024), ensures data integrity, making Agile project management more reliable. By implementing blockchain solutions, teams can maintain immutable records, prevent data tampering, and enhance trust in decentralized Agile workflows. As Agile software development continues to evolve, the synergy between emerging technologies and Agile principles will define the future of software engineering. The adoption of AI, ML, Blockchain, Cloud Computing, and IoT enhances agility, innovation, and security, ensuring that software projects remain competitive in an increasingly digital world. Through continuous technological integration, Agile methodologies will continue to drive efficiency, adaptability, and success in software development. Artificial Intelligence (AI) is playing a crucial role in Agile development by optimizing backlog prioritization, predicting project risks, and automating testing processes, AI accelerates Agile software development by making workflows more efficient and reducing manual effort.

Explored AI-generated prompts and tools that enhance Agile team efficiency, helping developers manage tasks more effectively. AI is also assisting research and decision-making within Agile projects. David, I.; Gelbard, R. (2025) introduced AI-based literature review automation, simplifying research analysis for Agile teams, allowing them to make data-driven decisions more efficiently. Machine learning (ML) is also enhancing Agile project management by improving forecasting and risk prediction. Zaidi, H.A.; Jain, P. (2024) reviewed how machine learning helps in predicting project outcomes and optimizing Agile workflows. Bahi, A.; Gharib, J.; Gahi, Y. (2024) explored how AI automates repetitive Agile development tasks, allowing teams to focus on complex problem solving rather than routine work. AI-driven models are also improving effort and cost estimation in Agile projects. (Rodríguez Sánchez, E.; Vázquez Santacruz, E.F.; Cervantes Maceda, H. (2023) and Arora, M.; Verma, S.; Wozniak, M.; Shafi, J.; Ijaz, M.F. (2022) introduced AI models that enhance Agile cost prediction and workload estimation, making Agile planning more accurate and efficient AI is also assisting research and decision-making within Agile projects. David, I.; Gelbard, R. (2025) introduced AI-based literature review automation, simplifying research analysis for Agile teams, allowing them to make data-driven decisions more efficiently. Machine learning (ML) is also enhancing Agile project management by improving forecasting and risk prediction. Zaidi, H.A.; Jain, P. (2024) reviewed how machine learning helps in predicting project outcomes and optimizing Agile workflows. Bahi, A.; Gharib, J.; Gahi, Y. (2024) explored how AI automates repetitive Agile development tasks, allowing teams to focus on complex problem-solving rather than routine work. AI-driven models are also improving effort and cost estimation in Agile projects. Rodríguez Sánchez, E.; Vázquez Santacruz, E.F.; Cervantes Maceda, H. (2023) and Arora, M.; Verma, S.; Wozniak, M.; Shafi, J.; Ijaz, M.F. (2022) introduced AI models that enhance Agile cost prediction and workload estimation, making Agile planning more accurate and efficient. Blockchain technology is revolutionizing Agile project management by increasing security, transparency, and accountability. Qureshi, J.N.; Farooq, M.S. (2024) introduced a blockchain-based model that enhances security in Agile projects, reducing risks related to fraud and unauthorized changes. Similarly, Qureshi, J.N.; Farooq, M.S. (2024) explored how blockchain improves Agile project management by ensuring decentralized and tamperproof tracking of software development progress, enhancing trust among stakeholders. Cloud computing is another major innovation influencing Agile development by providing scalable infrastructure, remote collaboration, and continuous integration and deployment (CI/CD). (Aouni, F.E.; Moumane, K.; Idri, A.; Najib, M.; Jan, S.U. (2025) reviewed how Agile, cloud computing, and AI work together to improve software development, allowing teams to work efficiently from different locations. Cloud-based Agile development enables faster deployments and seamless integrations, ensuring that software is always in a deployable state. The Internet of Things (IoT) is also benefiting from Agile methodologies, as smart devices require rapid iterations and continuous updates. Singh, D.; Damle, M. (2024) examined how IoT applications can be developed more efficiently using Agile approaches, while Aouni, F.E.; Moumane, K.; Idri, A.; Najib, M.; Jan, S.U. (2025) explored the integration of Agile and IoT, showing how Agile's iterative cycles help in testing and scaling IoT systems. Agile methodologies are also being applied to industries beyond traditional software development. Aditya, M.A.; Rokhman, N.; Nurmalasari, R.R.; Gumilar, S.; Alqinsi, P.; Ismail, N. (2023) demonstrated how an AI-powered smart greenhouse was developed using Agile, proving its adaptability to emerging technologies in agriculture and smart systems. As emerging technologies continue to evolve, Agile methodologies must adapt to leverage these advancements for improved efficiency and scalability. The integration of AI, blockchain, cloud computing, and IoT with Agile development is reshaping the future of software engineering. Organizations and developers must embrace these innovations to stay competitive, automate processes, and optimize workflows. Research contributions from various authors have significantly influenced Agile software development by addressing challenges such as security, cost estimation, automation, and scalability, ensuring that Agile remains a robust and future-ready methodology. The impact of Agile methodologies extends to the development of smart systems and IoT solutions. Studies by Singh, D. and Damle,

M. demonstrate how Agile accelerates IoT innovation through iterative prototyping and real-time testing. Practical applications, such as the AI-powered smart greenhouse developed by Aditya, M.A. et al., highlight how Agile methodologies successfully integrate with emerging technologies to drive advancements in intelligent system development. Ensuring developer well-being is another essential aspect of Agile success. AI driven monitoring tools, as introduced by Meza, J.L. et al., help track developer fatigue and improve productivity, fostering a more sustainable work environment. By leveraging AI to assess team performance and prevent burnout, organizations can maintain efficiency without compromising workforce well-being.

Critical Factors Influencing the Success of Software Development Projects in an Agile Environment

The success of software development projects in Agile environments depends on several critical factors, including technology integration, team collaboration, accurate planning, and security. Strong collaboration, adaptability, continuous integration, and leveraging emerging technologies. conducted H.A. and Jain, P., Katipoglu, G. et al., Sanchez, E.R. et al., Bahi, A. et al and others highlights how these elements contribute to the efficiency and effectiveness of Agile methodologies. The success of Agile software development projects is influenced by several critical factors, including technology integration, team collaboration, accurate planning, and security. One of the key determinants of Agile project success is precise planning and effort estimation. Rodríguez Sánchez, E. et al. (2023) and Arora, M. et al. (2022) emphasize the importance of AI powered models in predicting project costs and timelines more accurately. AI-driven estimations reduce the risks associated with underestimating resource needs, ensuring projects remain on track. Sanchez, E.R. et al. (2023) further refine Agile project planning strategies, focusing on optimizing time and effort allocation for better execution. The role of AI in Agile development extends beyond estimation to automation, where it enhances overall productivity. González Moreno, J.J. et al. (2022) highlight how AI speeds up Agile processes by automating repetitive tasks and improving decision-making. Bahi, A. et al. (2024) discuss the impact of AI driven automation in reducing manual effort, enabling developers to focus on more strategic tasks. Rozo Torres, A. and Sarmiento, W.J. (2024) explore AI's role in optimizing Agile workflows by generating prompts and insights that streamline communication and collaboration. Collaboration and teamwork are essential components of Agile success. (Juan Nunez, M.V. et al. (2024) propose AIScrum-Sprint, an AI enhanced Agile framework designed to improve teamwork and project management. (Katipoglu, G. et al. 2024) analyse the impact of Agile education on software development quality, underscoring the importance of well-trained Agile teams in achieving project goals. Agile methodologies focus on iterative development, where teams work in short cycles to deliver functional software quickly while responding to changing requirements. Ensuring that these principles are effectively implemented is crucial for the success of Agile projects. Effective team collaboration is one of the most important factors in Agile software development. Agile teams must work closely together, fostering a culture of open communication, knowledge sharing, and continuous feedback. Regular meetings, such as daily stand-ups and sprint reviews, help teams align their goals, track progress, and address challenges early. Tools like Jira, Trello, and Slack enable better communication and coordination among team members. Skilled and self-organizing teams further enhance Agile success by ensuring that individuals take ownership of their work, make decisions quickly, and remain adaptable to changes. Katipoglu, G.; Utku, S.; Mijailović, I.; Mekić, E.; Avdić, D.; Milić, P. (2024) examined the impact of Agile education and training, emphasizing the importance of proper knowledge transfer and skill-building for Agile teams to function effectively. Accurate effort and cost estimation are crucial for Agile success, as projects evolve based on user feedback and changing requirements. AI-driven models are increasingly being used to improve estimation accuracy Rodríguez Sánchez, E.; Vázquez Santacruz, E.F.; Cervantes Maceda, H. (2023) and Arora, M.; Verma, S.; Wozniak, M.; Shafi, J.; Ijaz, M.F. (2022) introduced

AI-based models to predict project duration and costs, helping Agile teams create more reliable plans. Arora, M.; Verma, S.; Wozniak, M.; Shafi, J.; Ijaz, M.F. (2022) also developed an AI-powered effort estimation model that helps teams distribute workloads effectively and optimize resources Zaidi, H.A.; Jain, P. (2024) explored how machine learning enhances Agile project planning by improving risk prediction and workload distribution, ensuring more accurate forecasts and better project execution. Continuous integration and deployment (CI/CD) play a vital role in Agile development, enabling teams to test, integrate, and release software frequently. This reduces development cycle times and ensures that software remains stable and functional. Automation tools such as Jenkins, GitHub Actions, and Docker support CI/CD pipelines, allowing teams to deploy new features quickly and fix issues in real-time. Agile environments benefit from automated testing frameworks, which help detect and resolve bugs early in development, reducing overall project risk. Customer involvement and regular feedback loops are key principles of Agile that help teams create products that meet user expectations. Agile methodologies prioritize continuous user input through iterative releases and usability testing. Juan Nunez, M.V. et al. (2024) introduced the AIScrum-Sprint framework, which integrates AI into Agile workflows to enhance customer interactions and improve project management. (Bahi, A.; Gharib, J.; Gahi, Y. (2024) explored how AIpowered automation in Agile development helps streamline repetitive tasks, allowing teams to focus on more critical aspects such as innovation and problem-solving. By incorporating frequent feedback, Agile teams can quickly adapt to changing market demands and user needs. Strong leadership and organizational support are essential for Agile success. Agile leaders play a crucial role in setting clear goals, removing roadblocks, and promoting a culture of collaboration and continuous improvement. Companies that invest in Agile training and provide the necessary resources for Agile teams tend to achieve better project outcomes. Coelho, P.A.; Casanello, F.; Leal, N.; Brintrup, K.; Angulo, L. (2024) demonstrated how Agile and Scrum methodologies help businesses adapt to Industry 4.0, enabling them to integrate new technologies and stay competitive in the evolving digital landscape. Similarly, Wynnedeta, D.E.; Telnoni, P.A. (2024) successfully applied Agile Scrum in mobile app development, showcasing how Agile methodologies improve project delivery and product quality. The integration of emerging technologies significantly influences the success of Agile projects by enhancing automation, security, and scalability. Artificial intelligence (AI) is one of the most impactful technologies in Agile development. AI-driven tools help teams optimize backlog prioritization, automate testing, and predict potential risks in software development. González Moreno, J.J. et al. Sanchez, E.R.; Santacruz, E.V.; MacEda, H.C. (2023) highlighted how AI accelerates Agile workflows, reducing manual effort and improving decision making processes Rozo-Torres, A.; Sarmiento, W.J. (2024) explored how AI-generated prompts and automation tools help Agile teams work more efficiently, improving project tracking and reducing the likelihood of delays. Blockchain technology is also playing a crucial role in Agile project management by enhancing security, transparency, and accountability Qureshi, J.N.; Farooq, M.S. (2024) introduced a blockchain-based framework to make Agile projects more secure by preventing unauthorized modifications and improving trust among stakeholders Qureshi, J.N.; Farooq, M.S.; Khelifi, A.; Atal, Z. (2024) further examined how blockchain technology ensures decentralized tracking of Agile development progress, helping teams maintain better records and improve data integrity. The integration of blockchain in Agile environments is particularly valuable for industries that require secure and tamper-proof project tracking. Cloud computing is another technology that enhances Agile development by providing scalable infrastructure, remote collaboration, and real-time deployment capabilities. Aouni, F.E.; Moumane, K.; Idri, A.; Najib, M.; Jan, S.U. (2025) reviewed how Agile, cloud computing, and AI work together to improve software development efficiency. Cloud based Agile development enables teams to collaborate from different locations, scale applications quickly, and maintain continuous integration pipelines, ensuring rapid software delivery and lower operational costs. The Internet of Things (IoT) is increasingly being developed using Agile methodologies, as IoT applications require frequent updates and continuous monitoring. Singh, D.; Damle, M. (2024) examined how Agile supports IoT development by enabling

quick prototyping and iterative improvements. Aouni, F.E.; Moumane, K.; Idri, A.; Najib, M.; Jan, S.U. (2025) explored how Agile methodologies complement IoT innovation, demonstrating how Agile principles help teams manage real-time data processing and device integration more effectively. Agile's ability to accommodate constant updates and user-driven enhancements makes it well-suited for IoT projects. AI is also being used to improve developer productivity and well-being, which directly impacts Agile project success. Meza, J.L.; Liñan, J.N.; Andrade-Arenas, L. (2024) developed an AI-powered drowsiness detection system to monitor developer fatigue, ensuring better work-life balance and improved efficiency. Maintaining a productive and healthy work environment is essential in Agile teams, as burnout can negatively affect project timelines and software quality. Additionally, Agile is being successfully applied beyond software development into new fields. Aditya, M.A.; Rokhman, N.; Nurmalasari, R.R.; Gumilar, S.; Alqinsi, P.; Ismail, N. (2023) demonstrated how Agile methodologies were used to develop an AI-powered smart greenhouse, proving Agile's adaptability in managing complex, technology-driven projects. As Agile software development continues to evolve, organizations must embrace best practices, leverage technological advancements, and maintain strong collaboration to achieve long-term success. The integration of AI, blockchain, cloud computing, and IoT is reshaping Agile workflows, enabling teams to work more efficiently, enhance security, and scale applications faster. Research contributions from various authors have played a significant role in shaping modern Agile methodologies by addressing challenges such as cost estimation, automation, security, and scalability. These contributions ensure that Agile remains a highly effective and future-ready approach for software development in an increasingly technology-driven world.

3. Methodology

This study was carried out to solve a real and practical problem by creating something useful specifically, a set of guidelines to help Agile software teams adopt new technologies like AI. The goal is to make sure these teams can handle challenges like keeping up with rapid changes, working well together, and scaling their solutions effectively. What makes this research important is that it doesn't just talk about the problem it offers a practical solution backed by research. To do this, the study uses a method called Design Science Research (DSR). This approach is all about designing and testing new things like models, tools, or guidelines that help solve real-world problems. Unlike traditional research methods that focus more on observing or analyzing things, DSR is hands-on. It's about creating solutions and showing how they can work in practice. In this case, the solution is a set of guidelines. These are treated as an "artifact" in DSR terms meaning something that's designed to be useful and applied in the real world. The process starts by clearly identifying the problem: how Agile teams struggle to adopt technologies like AI, blockchain, or cloud computing. These tools bring value, but they can also be difficult to integrate into fast-moving Agile environments. After defining the problem, the next step is to decide what the solution should achieve. The guidelines need to be practical, flexible, and easy for teams to use while still fitting Agile principles. Then, the guidelines are designed and developed using research, expert input, and industry experience to make sure they're grounded and relevant. Once the guidelines are created, they're tested in real or simulated Agile projects to see how well they work. This helps the researchers see if the guidelines are useful and how they can be improved. After testing, the guidelines are evaluated through feedback from Agile professionals like developers, project managers, or team leads. This helps measure their impact, usability, and effectiveness. Finally, the findings are shared with both researchers and industry professionals through papers, presentations, and reports. This way, others can benefit from the results—whether they're academics or people working on Agile teams. In the field of Information Systems (IS), using DSR is not new—but many researchers still don't use it often enough. A study by Zeng et al. (2020) pointed this out and encouraged more researchers to apply methods like DSR. It's already been shown to work well for creating and testing solutions in IS, especially when tackling practical problems (Rai, 2017; vomit Brocke et al., 2020). DSR has also helped support innovation in organizations and even broader goals like sustainability. In IS, this includes everything from building databases to supporting business strategies and using technology to make better decisions (Becker et al., 2015; Seidel et al., 2013). Because this study is all about designing and testing something that teams can use, DSR is the best fit. It offers a clear, structured way to build, test, and improve the guidelines. And by doing this, the study doesn't just offer a tool for teams—it also adds valuable knowledge to the research community.

3.1 Design Science Research

Figure 11 shows a basic way to understand, carry out, and judge design science research, based on Hevner and colleagues (2004). First the environment is where the problem exists. It includes people, organizations, and the technology they use or plan to use. Inside this environment, there are goals, tasks, problems, and opportunities that create needs, as seen by the people involved in the organization. These needs are looked at considering the organization's strategies, culture, structure, and current work processes. They also depend on what technology and communication systems are already in place. All of this together forms the "research problem" that the researcher wants to solve. Focusing on real needs from stakeholders helps make sure the research is useful and relevant. Next, the knowledge base is where the research draws its materials from. It has two parts: Foundations and Methodologies. Foundations include previous research, theories, models, tools, and examples from other fields that help build the study. Methodologies provide the guidelines for how to evaluate the

results. Applying these foundations and methods correctly is how researchers ensure their work is solid and trustworthy.

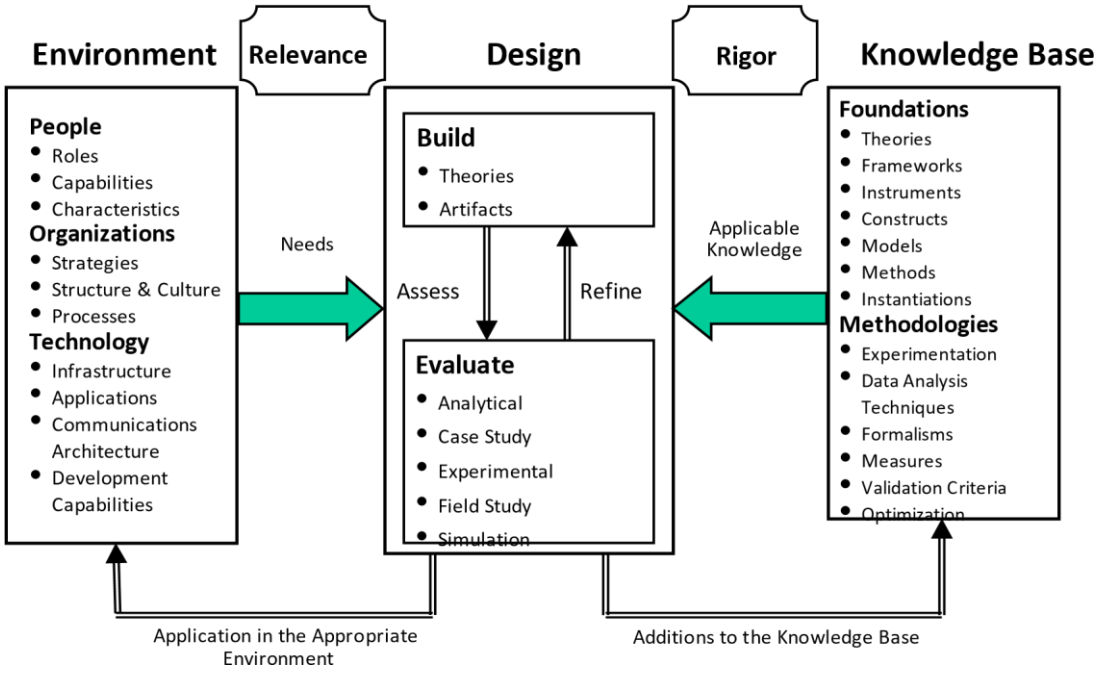

Figure 11: Design Science Research Framework Adapted - Source: Hevner et al. (2004)

Design Science Research (DSR) looks at real problems that happen in different areas of work and life. It focuses on finding solutions that can be tested with real people in organizations using certain technologies. Usually, a DSR project starts by studying the business environment to figure out what problems need solving. Sometimes, these problems have already been researched, so the project can build on what's already known. DSR also checks what academic knowledge is available like theories, frameworks, or tools to see if they can solve the problem. If the answer is yes, then using that existing knowledge is called "routine design," which isn't really considered DSR. But if no good solution exists yet, DSR works on creating a new or better solution by combining and improving what's already out there. The process involves designing and building the solution, then testing it to see if it works well, usually going through several rounds of improvements. During the project, different research methods are used, like interviews, surveys, reading past studies, or focus groups, to gather useful information and feedback. The "environment" is where the problem exists it includes people, organizations, and the technologies they use or plan to use. Inside this environment, there are goals, tasks, problems, and chances for improvement. These things create the "needs" that stakeholders in the organization recognize. These needs are looked at in relation to things like the company's strategy, culture, structure, and current work processes, as well as the technology and tools they already have. All these together shapes what the researcher sees as the "research problem." By focusing on real needs from stakeholders, the research stays relevant and useful. The "knowledge base" is like a toolbox that DSR uses to create solutions. It includes foundational theories, methods, models, and tools from previous research. This knowledge helps during the building phase of

the research. Methodologies from this toolbox also guide how to evaluate and test the solution. Using these foundations and methods properly is what makes the research rigorous and reliable.

3.2 Design Science Research Process

There are a few different ways people have organized how Design Science Research (DSR) projects are done. One popular model comes from Hevner (2007), but the one most people refer to is by Peffers and colleagues from 2008. Their model breaks down the research process into six main steps: first, figuring out the problem and why it matters; then setting goals for what the solution should do; next, designing and building the actual solution; after that, showing how it works; then checking if it really solves the problem; and finally, sharing the results with others. The model also says you can start the process in different ways sometimes you begin by looking at the problem itself, other times by focusing on the goals for the solution, or you might jump straight into designing something, or even start based on what a client or situation needs. Below, I'll give a quick rundown of each of these steps.

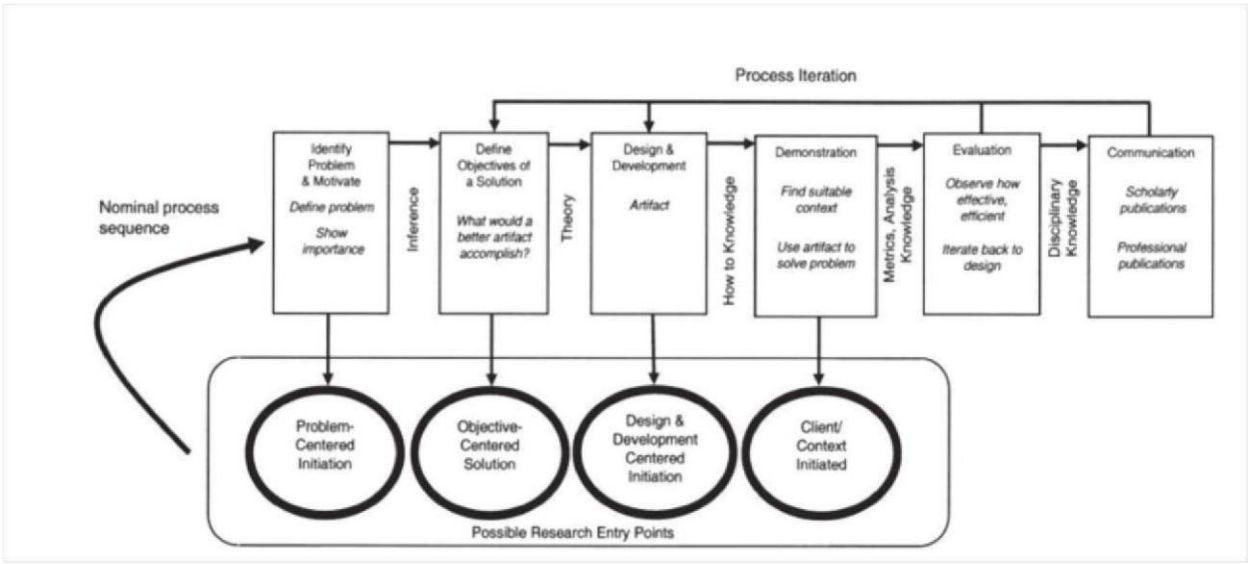

Figure 12: DSR Methodology Process Model Adapted - Source: Peffers et al. (2008)

Activity 1. Problem identification and motivation This step focuses on clearly defining the specific problem the research aims to solve and explaining why finding a solution matters. Highlighting the value of the solution serves two key purposes: it encourages both the researcher and the audience to engage with the research, and it demonstrates the researcher's deep understanding of the problem. Completing this activity requires familiarity with the current state of the problem and the significance of addressing it.

Activity 2. Define the objectives for a solution. Based on the problem definition and what is achievable, this step sets the goals for the proposed solution. These objectives may be qualitative, such as describing how the new artifact will help solve issues that have not been addressed before. The objectives should be logically derived from the problem outlined in the first step.

Activity 3. Design and development. An artifact is created. Conceptually, a DSR artifact can be any designed object in which a research contribution is embedded in the design. This activity includes determining the artifact's desired functionality and its architecture and then creating the actual artifact.

Activity 4. Demonstration. This activity demonstrates the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity.

Activity 5. Evaluation. The evaluation measures how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in context. Depending on the nature of the problem venue and the artifact, evaluation could take many forms. At the end of this activity the researchers can decide whether to iterate back to step three to try to improve the effectiveness of the artifact or to continue to communication and leave further improvement to subsequent projects

Activity 6. Communication. In this final stage, the problem and the created solution known as the artifact are communicated to the relevant groups, such as industry experts and researchers. The method of communication is chosen based on the audience and the research goals. For example, professionals might prefer practical guidance, while academics may focus more on theoretical insights.

In Design Science Research (DSR), it's essential that the artifact not only addresses the identified problem conceptually but is also practical and usable in real-world settings (Hevner et al., 2004). The artifact developed in this study is an implementation framework designed for companies to evaluate and enhance their current practices when adopting emerging technologies in agile software development. This framework also adds to the existing knowledge on guidelines for integrating new technology in agile projects. It should be considered an initial version that future researchers can expand and refine through further development cycles.

As noted by Rai (2017) in a MIS Quarterly editorial, while DSR projects often share similar foundations, the way they are carried out can vary widely depending on the problem, objectives, and type of artifact being developed. Different types of DSR approaches have been identified based on these factors. The DSR methodology used in this study will be described in more detail in the next section.

3.3 Research Strategy

The definition of DSR as per Hevner & Chatterjee, (2010) is "a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing with new knowledge to the body of science evidence. The design artifacts are both useful and fundamental in understanding that problem. One has found several literatures about DSR methodology, all of them proposing similar models to develop artifacts that solve a particular problem, but for the purpose of this work this researcher will follow an adaptation of the model proposed by Peffers et al., (2007). This model consists of several iteration steps which are used while applying the methodology, starting by identifying the problem and its motivation, defining the objectives of the solution, then designing and developing the solution, evaluating it, and finally communicating it as a new disciplinary knowledge for the scientific and academic community. The model adaptation and its steps can be depicted like in figure 13.

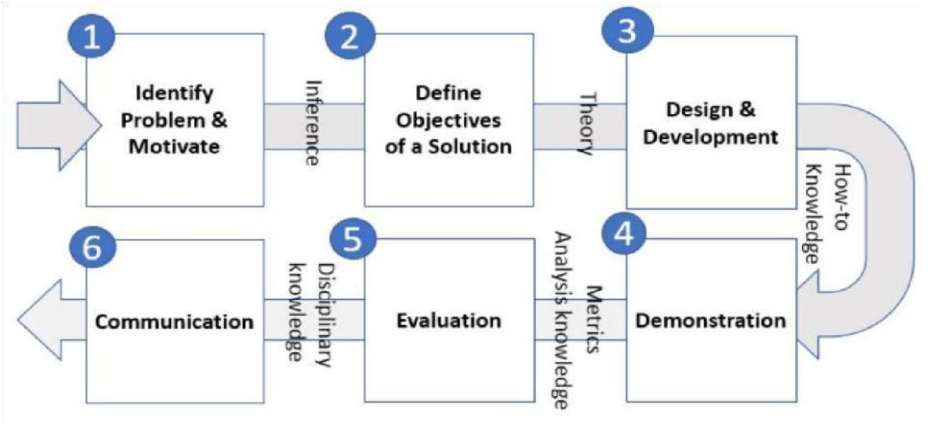

Figure 13: DSRM Process Model Adaptation - Source: Peffers et al., (2007).

Each step in this research process was carefully designed to help reach the main goal of the study. The first step is to clearly define the research problem and explain why solving it matters. This helps the researcher stay focused and shows others why the work is important. To do this, the researcher investigates what's already been done on the topic especially in areas like agile software development and the use of new technologies. By reviewing the existing research, they find current challenges and gaps in how new tech, like AI or Blockchain, is used in agile environments. Talking to people in the industry like software developers and project managers also helps confirm that this is a real problem worth solving. This step ends with setting clear goals for what the research aims to fix or improve. Once the problem is understood, the second step is to set specific goals for what the new solution should achieve. These goals explain how the new idea will be better than or work alongside existing ones. In this case, the goal is to create helpful guidelines for using emerging technologies in agile software development. These guidelines need to solve the earlier-identified problems and help agile teams work more smoothly with new tech. They should also be flexible enough to work in different types of companies and teams. The focus here is on making sure the solution works in real-life situations. The third step is to build the actual solution also called the "artifact." This can be a model, a method, or a practical tool. Here, the researcher designs the structure and features of the guidelines. The guidelines will include step-by-step instructions for agile teams on how to adopt and use new technologies. This part of the process is done in cycles: the researcher creates a version, gets feedback, and then improves it. The goal is to make sure the guidelines are useful, easy to understand, and can be put into action right away.

In the fourth step, the researcher tests the solution to see if it works. This involves comparing what the solution was supposed to achieve with what it did in practice. The researcher gathers feedback from experts and people who used the guidelines in real projects. This helps figure out what worked well and what needs to be improved. Based on the feedback, the researcher may go back and make changes to make the solution better. The main aim here is to make sure the solution really helps agile teams use new technologies successfully. The final step is to share the finished guidelines with others. This includes researchers, software professionals, and anyone else who might benefit from the work. The researcher will present the solution through papers, conferences, workshops, or training sessions. This helps spread the word and encourages others to use and improve the guidelines. Ongoing feedback from real users will help keep the solution up to date and useful in different situations. The goal of this last step is to make sure the research has a real impact in the world of agile software development and technology adoption.

3.4 Research Implementation

This research focused on putting earlier planning into action. It started with identifying the main problem, which could only be done after clearly understanding the topic of the study. The researcher was interested in agile software development and new technologies, which led to exploring the difficulties of combining the two.

Agile methods focus on being flexible, working in teams, and making progress step by step. However, adding new technologies like artificial intelligence (AI), blockchain, and cloud computing can bring challenges, such as being hard to understand, people being slow to accept change, or the new technology not fitting well with existing systems. The main goal of this study was to create a clear guide to help agile teams use new technologies without losing the benefits of being agile. A review of past research helped identify the key problems and gaps, showing the need for a guide that connects technology use with agile values. The first part of the study explains why this problem matters and what the study hopes to fix. From this starting point, the researcher focused on two main areas: agile software development and new technologies. A systematic literature review was done to learn how teams are currently using new tech in agile projects, what the benefits are, what problems they face, and how things could be improved. This review, discussed in the next part of the study, helped the researcher learn about good practices, common mistakes, and successful methods.

After the review, the researcher set clear goals for the guide. These goals focused on helping teams adopt new technology in ways that support agile methods instead of making things harder. With enough information gathered, the researcher moved on to the design and development of the guide. The guide was built based on the insights from earlier steps, making sure it was practical, easy to use, and matched agile ways of working. It provides clear, step-by-step advice for bringing new technologies into agile projects, while focusing on key areas like working together, checking technology fit, and improving bit by bit.

To test the guide, the researcher spoke with experts in the industry and people with experience in agile methods. These interviews and feedback sessions helped show how useful the guide was and where it could be improved. While there weren't enough time or resources to collect data from many people, the feedback from experts was helpful for making the guide better. Because of time limits, the final step sharing the guide with the wider academic and professional world was not fully completed. The study mentions this as a limitation, and future plans include publishing the guide in journals, conferences, and workshops.

The research method used here gave a strong base for creating new knowledge and making sure the work was valid and reliable (Cruz & Cruz, 2020). In the field of Information Systems (IS), there are usually two research types: behavioural science (focused on building and testing theories) and design science (focused on creating and testing useful tools or frameworks) (Benjamin M. Ampel et al., 2023).

Since this study aimed to create a useful guide, the researcher used a Design Science Research (DSR) approach. The purpose was to develop a clear, helpful guide for solving real problems with using new tech in agile development. According to the DSR knowledge framework, this study adds to current understanding and introduces new ideas about how to bring new technologies into agile projects (Gregor & Hevner, 2013). The guide helps solve real-life problems and supports better business outcomes by making it easier to adopt new technologies in agile settings (March & Storey, 2008).

4. Empirical Study Guidelines for Emerging Technologies in Agile Software Development projects

This chapter has six sections that explain how to build a framework for using emerging technologies in Agile software development project and how the framework was created to help agile teams or companies to adopt or use new technology in agile environment. This first section lays out the key assumptions behind the model, based on insights from the literature review, focusing on technologies like AI-driven development, cloud computing, DevOps automation, and blockchain security in Agile workflows. The second section dives into the design of the framework, detailing workflows and practical tools that support its implementation, following phase three in the Design Science Research (DSR) process. The third and fourth sections focus on testing and evaluating the mode Agile project and then through expert interview to assess its real-world relevance. The five section presents a refined version of the framework, incorporating feedback from the case study and interview to ensure it is practical and effective for Agile teams working with emerging technologies. The final section represents the conclusions and future research

4.1 Assumptions

Based on the findings from the literature review, which addressed the research questions, a framework can now be developed for integrating emerging technologies into Agile software development. Drawing from the reviewed literature on both emerging technologies and Agile methods, several key assumptions have been identified. These should be considered when designing a practical framework to assess and guide the adoption of these technologies in Agile environments.

- A1: Integrating emerging technologies into Agile software development is a step-by-step process that needs constant adjustment to keep up with fast-changing technology and industry standards.
- A2: The mindset and openness of agile teams and stakeholders toward emerging technologies significantly influence their successful adoption. Creating awareness and fostering a culture of experimentation and innovation is crucial for achieving long-term benefits.
- A3: Successfully leveraging emerging technologies requires new technical and organizational capabilities. Agile teams must actively manage change through continuous learning, cross functional collaboration, and structured transformation strategies
- A4: Without a well-defined strategy for adopting emerging technologies, integration efforts risk becoming fragmented and reactive. Therefore, organizations must assess their readiness by evaluating their existing agile practices, technical infrastructure, and innovation culture before implementation.
- A5: Current Agile frameworks don't fully address the specific challenges and opportunities that come with emerging technologies. As a result, they need to be adjusted to support a more structured and effective way of integrating these technologies.
- A6: Beyond assessing readiness, organizations must develop a tailored strategy for emerging technology adoption that aligns with their agile workflows, business objectives, and specific operational constraints.
- A7: A maturity model is key to understanding how ready an organization is to use emerging technologies in Agile projects. It should offer clear steps for checking current strengths, setting goals for improvement, and continuously boosting Agile practices
- A8: At the time of writing, no comprehensive framework exists that guides agile teams in both the readiness assessment and structured implementation of emerging technologies. A consolidated approach is needed to integrate best practices and existing research into a unified model.

- A9: A complete framework for integrating emerging technologies into agile software development should encompass people, processes, and technology. It should provide a structured roadmap for assessing readiness, defining a strategy, implementing new technologies, and measuring maturity while remaining adaptable to various agile environments 2

4.2 Guidelines for Emerging Technologies in Agile Implementation

Emerging technologies are important in Agile software development, but the field is still evolving and has many complexities that need further exploration. Chapter 2 covered key Agile concepts, assessment tools, and the impact of technologies like AI driven development, cloud computing, and DevOps automation. It also examined existing models, pointing out their shortcomings in design and usability. One major issue is the lack of a clear distinction between assessing an organization's readiness to adopt emerging technologies in Agile and measuring its maturity in using them effectively. So far, no research has fully connected the goals and outcomes of both assessments, leaving a gap in how these tools should be applied in Agile environments. However, readiness assessments naturally influence maturity assessments, making it essential to align them for a smooth transition. To build a more complete framework, insights from the literature review will be combined with existing models to create a practical toolset for Agile teams. As discussed in the systematic literature review, evaluating an organization's readiness its technical infrastructure, Agile team capabilities, and overall strategy before implementation, while ensuring involvement from all stakeholders, is key to successfully integrating emerging technologies in Agile software development. This diagram presents a comprehensive Agile transformation and readiness framework for integrating emerging technologies into Agile software development. It is divided into five sequential phases: Agile Strategy, Readiness Assessment, Agile Implementation, Performance Assessment, and Continuous Improvement. Together, these phases provide a structured pathway from planning to sustainable change.

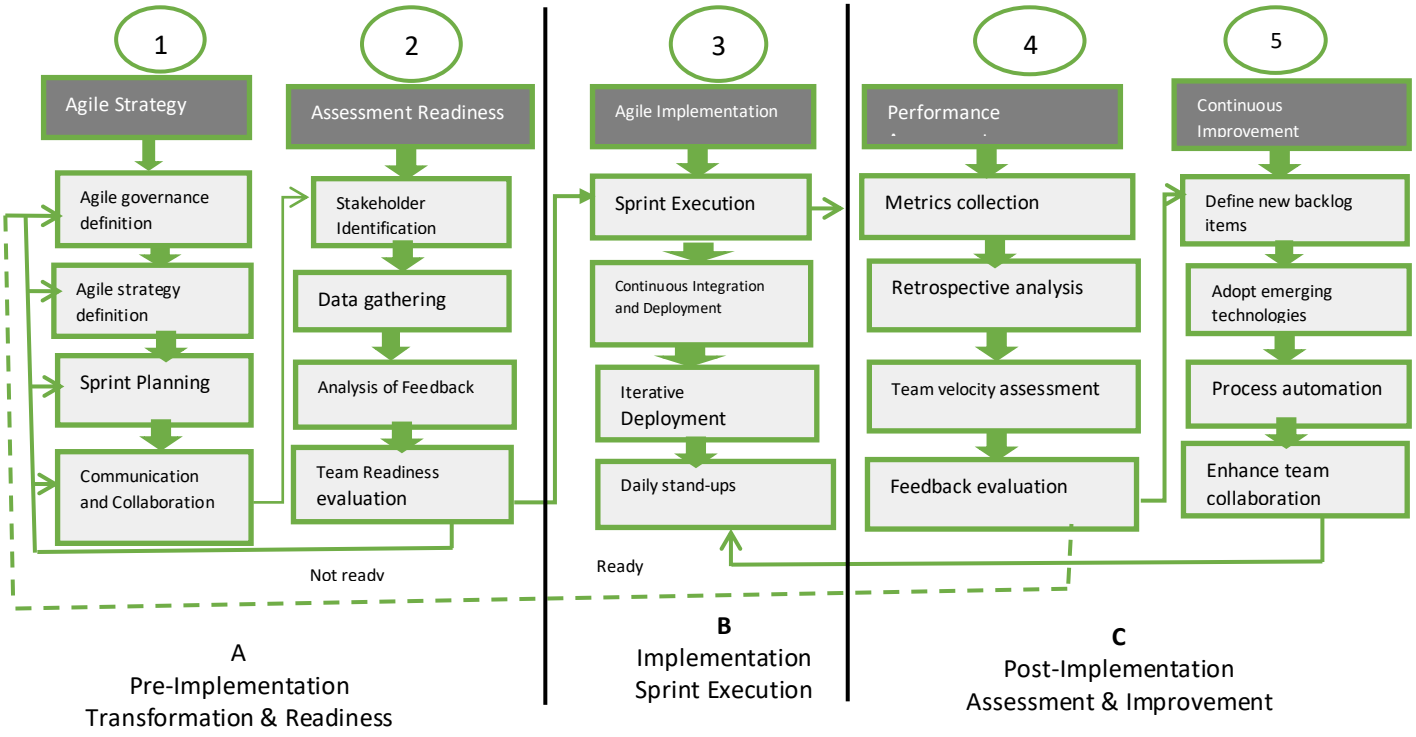

Figure 14: Systematic for emerging technology in agile software implementation framework (own illustration)

Phase 1: Agile Strategy

This phase defines the strategic groundwork necessary for successful Agile transformation. The transformation begins with the definition of Agile governance structures and strategic direction. This phase includes setting governance mechanisms, articulating the Agile strategy, conducting sprint planning, and fostering communication and collaboration. These foundational activities ensure the organization is aligned internally before Agile practices are deployed.

- **Agile Governance Definition** – Establishes decision-making structures, accountability, and leadership responsibilities for Agile initiatives.
- **Agile Strategy Definition** – Sets high-level goals and outlines how Agile will align with business and technology objectives.
- **Sprint Planning** – Prepares the team to break down work into deliverable increments, prioritize tasks, and set goals for the first development cycle.
- **Communication and Collaboration** – Promotes a collaborative environment and ensures effective communication among cross-functional teams.

Phase 2: Readiness Assessment for Agile Implementation

This phase assesses whether the organization is prepared to adopt Agile practices and emerging technologies. It begins with stakeholder identification and continues with data gathering, feedback analysis, and evaluation of team readiness. This diagnostic step helps identify cultural, structural, or operational gaps that need to be addressed prior to implementation.

- **Stakeholder Identification** – Determines who will participate in or be affected by the Agile initiative.
- **Data Gathering** – Collects information through interviews, questionnaires, or observations to assess current capabilities and perceptions.
- **Analysis of Feedback** – Synthesizes collected data to identify gaps and readiness levels.
- **Team Readiness Evaluation** – Assesses whether teams possess the mindset, culture, structure, and resources required to begin Agile implementation.

Phase 3: Agile Implementation

This is the action phase, where Agile methods are applied in real development settings. Once readiness is confirmed, the organization begins actual Agile execution. Activities include sprint execution, continuous integration and deployment, iterative delivery, and daily stand-ups. This stage emphasizes short feedback loops, adaptive planning, and team alignment with Agile values.

- **Sprint Execution** – Teams begin delivering working software in time-boxed iterations.
- **Continuous Integration and Deployment** – Practices like automated testing and continuous delivery are introduced to maintain flow and speed.
- **Iterative Deployment** – Ensures incremental progress with frequent feedback and adaptation.
- **Daily Stand-ups** – Short team meetings help monitor progress, address blockers, and align daily work with sprint goals.

Phase 4: Performance Assessment

This phase evaluates how well the implementation is working. Following implementation, performance is assessed through metrics collection, retrospective analysis, velocity tracking, and structured feedback evaluation. This enables teams to measure impact, learn from outcomes, and identify further areas for improvement. It ensures transparency, learning, and measurement of success.

- **Metrics Collection** – Tracks key performance indicators (e.g., velocity, lead time, defect rate) to quantify results.
- **Retrospective Analysis** –Helps teams review what worked, what didn't, and what could be better next time.
- **Team Velocity Assessment** – Measures the pace and consistency of Agile delivery.
- **Feedback Evaluation** – Gathers insights from stakeholders to assess satisfaction, progress, and usability of new practices.

Phase 5: Continuous Improvement

The final phase focuses on sustaining agility through ongoing reflection and refinement. New backlog items are defined, emerging technologies are strategically adopted, and opportunities for automation and improved collaboration are identified. This ensures that Agile transformation remains dynamic and responsive to internal and external changes. By combining strategic alignment, readiness evaluation, implementation, measurement, and iterative refinement, this framework supports a holistic, phased approach to Agile transformation particularly in organizations seeking to integrate emerging technologies into their software development lifecycle.

- **Define New Backlog Items** – Improvement actions are turned into actionable backlog tasks.
- **Adopt Emerging Technologies** – Lessons from earlier phases guide the adoption of technologies that improve agility and product quality.
- **Process Automation** – Identifies areas for automation to reduce manual effort and increase efficiency.
- **Enhance Team Collaboration** – Builds stronger team dynamics, communication practices, and Cross functional cooperation.

This phase emphasizes that Agile transformation is ongoing, not a one-time event. To create a clear and practical framework based on the literature, the process is divided into three main phases: pre-implementation, implementation, and post-implementation.

The pre-implementation phase focuses on preparing the right environment to ensure smooth adoption of new initiatives by addressing key success factors and requirements. This helps select the best initiatives to maximize benefits. The implementation phase is where plans are put into action and initiatives are rolled out. The post-implementation phase involves evaluating performance, measuring results against goals, and identifying areas for improvement. Figure 14 presents this three-phase framework, which outlines the steps needed to successfully integrate emerging technologies into Agile software development. The process includes four main steps: establishing key success factors, conducting readiness assessments, implementing Agile initiatives, and continuous improvement. Emerging technologies like AI, cloud computing, blockchain, and DevOps affect the entire organization, so it's crucial to involve all stakeholders to reduce resistance and promote awareness. The first step is securing organizational awareness and support from senior management and setting up governance with the authority to drive change and allocate resources. A major

barrier to adopting new technologies is organizational readiness. The second step is a readiness assessment to check if the necessary technology, people, and processes are in place for long-term adoption. Only if readiness criteria are met should technology initiatives be implemented, which is the third step. The final step includes performance assessment and continuous improvement, focusing on refining initiatives or adopting new technologies to further boost Agile capabilities. This cycle of evaluation and improvement ensures ongoing agility and adaptation to changing business and technology needs. Steps 1 and 2 will be discussed in detail with recommended methods and tools for gathering and analysing data. The following steps depend on these outcomes and will be briefly outlined with general guidance.

4.2.1 Emerging Technology in Agile software Development Strategy

A continuous observation in the literature is how emerging technologies affect the entire organization. Because they impact many areas and people, adopting these technologies in Agile software development requires an overall organizational change. The human side is especially important and often the hardest part—people need to be carefully considered and supported throughout the process. The goal of this step is to make sure the organization's strategy, structure, leadership, and culture are ready for adopting new technologies. As shown in section 3.1, organizations can take many different paths, so it's important to focus on the technology initiatives that will have the biggest impact based on the type of organization. To help with this, a decision model is included to guide companies in choosing the right technology projects depending on their specific situation. An organization's processes and strategy not only shape which technologies will be useful but also how progress is tracked, and goals are set. That's why each organization needs to create its own evaluation methods and make sure the right processes are in place. Figure 15 shows an example workflow that covers all these steps. The following sections will explain each step along with useful tools and practices.

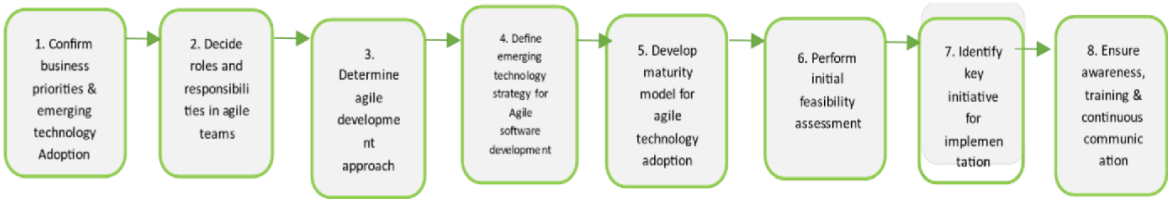

Figure 15: Agile software development Strategy workflow (own illustration)

1. Confirm Business Priorities and Emerging Technology Adoption

Before integrating emerging technologies into an agile software development project, organizations must identify their key business priorities. The motivation behind adopting new technology can include improving development speed, enhancing software quality, reducing costs, gaining a competitive edge, or meeting customer demands. One of the biggest challenges in technology adoption is the lack of management support and funding. Without leadership commitment and proper resources, successful integration becomes difficult. Ensuring that senior management endorses and allocates the necessary budget for emerging technology adoption is crucial for success.

2. Decide Roles and Responsibilities in Agile Teams

Clear governance and role allocation are essential for the seamless adoption of emerging technologies in agile software development. Organizations must define key roles such as a Chief Technology Officer (CTO), Technology Lead, or Agile Coach to oversee the technology implementation process. Instead of relying on a single individual, responsibilities should be distributed across agile teams to encourage collaboration and shared ownership. This ensures that the agile development process aligns with technology adoption and integrates effectively between business and technical teams.

3. Determine Agile Development Approach

To support the successful adoption of emerging technologies, organizations must determine which Agile development methodology suits their goals and team dynamics. Choosing between Scrum, Kanban, SAFe, or hybrid models depends on project size, complexity, and team maturity. The selected approach should allow flexibility, iterative delivery, and rapid feedback loops features that are essential when dealing with experimental technologies. It's also important to evaluate how the Agile framework integrates with the technology lifecycle, ensuring alignment between the development process and the pace of innovation.

4. Define an Emerging Technology Strategy for Agile Software Development

Having a clear and structured strategy for integrating emerging technologies into Agile projects is vital. This includes identifying the technology's purpose, intended benefits, and the specific problems it aims to solve. A strategic plan should outline implementation phases, technical requirements, integration points, and metrics for success. Additionally, the strategy must consider scalability and future expansion. It should be documented, communicated to all stakeholders, and embedded into the Agile planning process to ensure consistency and transparency throughout development cycles.

5. Develop maturity model for agile technology adoption

Research shows that setting clear goals and regularly checking progress is key to successfully using new technologies in Agile software development. These benefits often take time to appear, so organizations need to be patient. To keep projects moving forward and avoid giving up, organizations should create specific, measurable goals, focus on the most important initiatives, and set clear timelines and ways to track success. A maturity model is a helpful tool in this process. It not only helps set goals but also provides a way to keep improving and compare progress against internal targets and industry competitors. Since maturity models differ in complexity and focus, they should be adapted to fit the organization's strategy and how it plans to implement changes. This research highlights the strengths and weaknesses of various models, helping organizations choose or adjust one that suits their needs best. Besides picking a maturity model, organizations should also set up clear and consistent processes for evaluating, developing, and rolling out new technology projects. This means having structured ways to set goals, clear timelines, and regular reviews to make sure everything stays aligned with business goals and Agile practices.

6. Perform Initial Feasibility Assessment

Before implementing emerging technology, an organization should conduct a feasibility assessment to evaluate its readiness. This assessment identifies gaps in infrastructure, skills, and development practices. By analysing current capabilities, organizations can determine which technologies provide the highest value with

minimal risk. The assessment ensures that emerging technology adoption aligns with business objectives and is technically viable within the agile framework.

7. Identify Key Initiatives for Implementation

Once Organization complete the feasibility assessment, the next step organizations need to prioritize and select the first set of initiatives for implementation with a focus on achieving quick wins that demonstrate immediate value and boost confidence in adopting emerging technologies, such as integrating AI-powered code reviews to improve speed and accuracy in code quality assessments, automating testing processes to reduce manual effort, speed up feedback cycles, and improve software quality, and adopting cloud-based development environments to enhance collaboration, scalability, and flexibility within Agile teams, and the best approach is to start with the initial initiatives that are low risk but have a high-impact. These early successes will enable organizations to quickly gain tangible results, after which additional initiatives can be introduced and refined as part of the ongoing Agile development cycle, helping organizations continuously evolve and improve their use of emerging technologies in Agile processes.

8. Ensure Awareness, Training, and Continuous Communication

Successful adoption of emerging technologies requires continuous communication and capability-building efforts. Teams must be equipped with the necessary skills through training programs, hands-on workshops, and ongoing learning opportunities. Additionally, creating awareness about the purpose, benefits, and expectations of the technology helps drive engagement and reduce resistance. Regular updates, feedback loops, and cross-team collaboration are essential to maintaining momentum and adapting to any changes or lessons learned during the adoption process. By structuring communication this way, organizations ensure that every stakeholder receives relevant, actionable insights without unnecessary complexity, fostering transparency and engagement in an Agile-driven environment. Once the technology is implemented, ongoing communication should continue to keep stakeholders updated on progress and achievements. While the maturity model is helpful for goal setting and progress tracking, it is not ideal for communicating with most stakeholders. To ensure successful adoption of emerging technologies in agile software development projects, organizations must assess their readiness prior to implementation.

4.2.2 Readiness Assessment for Agile Implementation

Based on research and comparing existing readiness models, certain parts of the organization need to be checked before starting Agile. This helps make sure the organization is ready to successfully use and maintain Agile practice. Agile transformation is not only a process shift but also a cultural and structural change, requiring organizations to evaluate their internal alignment and capacity for change. From the systematic literature review (SLR) and other supporting sources, three key dimensions have been identified as critical to organizational readiness for Agile implementation. These dimensions are applicable across different types of organizations and initiatives and include:

Strategy and Governance, Awareness and Attitude Processes

As emphasized in the previous chapter, these dimensions must be carefully considered before Agile practices are introduced. While some of the existing readiness models also include technological and practical considerations, these were excluded from the readiness phase in this model. Instead, such elements are addressed during the Agile implementation phase and are largely influenced by the organization's strategic

direction, leadership involvement, culture, and internal processes. This focuses on evaluating an organization's current state across the following three critical areas:

1. **Strategy and Governance**

This area assesses the organization's vision, leadership, and structural preparedness for adopting Agile practices. The following factors are considered:

Objectives: The organization should set clear and easy-to-measure goals that follow Agile values. These goals should be specific, realistic, important, and have deadlines for using new technologies in Agile projects.

Motivation: The rationale behind adopting new technologies should be explicitly stated. This includes identifying the value expected from implementation, whether in terms of efficiency, innovation, competitiveness, or customer satisfaction.

S**trategy:** A formal strategy should be developed to outline how the organization plans to integrate emerging technologies into its Agile development processes. This strategy should cover implementation phases, responsibilities, expected outcomes, and alignment with business goals.

Impact: The anticipated organizational benefits of Agile adoption must be communicated. The anticipated benefits and organizational changes resulting from technology adoption should be clearly articulated and communicated to all relevant stakeholders to ensure shared understanding and alignment.

Budget. A specific budget needs to be set aside to support Agile transformation, including training and tools. Another budget should be assigned for using new technologies. This shows the importance of the project and makes sure there are enough money, skills, and people available.

Roles and Responsibilities: Governance structures should be established, including key Agile roles (e.g., Scrum Master, Product Owner). Governance structures must be clearly defined, with clear roles and responsibilities are assigned to make sure there is proper oversight and accountability. While a central coordinator (e.g., a digital transformation lead or Agile technology champion) may oversee the process, shared responsibility among team members is essential to avoid over-reliance on a single individual.

Approach: The way technology is introduced needs to fit the organization's unique situation, like its industry, size, and technical skills. Since not every tech project will bring the same results everywhere, it's important to focus on those that offer the biggest benefits and are easiest to do.

2. **Awareness and Attitude Knowledge:**

Employees should be familiar with Agile principles and how they relate to their roles. This factor assesses how well employees understand the principles related to emerging technologies and their implications within Agile environments. Staff should be trained not only in the technical aspects but also in the broader economic, regulatory, ethical, and operational impacts of technology adoption.

Awareness: There must be organizational understanding of Agile goals, benefits, and the change process. Awareness evaluates how informed and engaged employees are regarding the organization's technology adoption initiatives. Employees should be familiar with the strategic objectives, motivations, goals, and potential impacts linked to adding new technologies into Agile method.

Attitude: A positive attitude toward Agile is necessary for successful adoption. Attitude reflects how employees perceive the need for adopting emerging technologies and their potential to bring positive change. A constructive and supportive attitude is essential for fostering organizational acceptance and smooth implementation of Agile related transformation.

Human Behaviour and Culture: Employees should be open to iterative work, collaboration, and feedback. This factor examines the willingness of individuals to adjust their daily work behaviours and actively

participate in technology enabled Agile practices. A culture of openness and adaptability is key to successful transformation.

Organizational Behaviour and Culture: The broader culture should support adaptability, learning, and transparency. The organization's structure and culture should encourage innovation, collaboration, and continuous learning. Systems and policies must support the adoption of modern practices and enable teams to integrate new technologies effectively.

3. Processes

The final dimension evaluates whether the organization has the necessary internal mechanisms to support Agile implementation:

Standardized Processes: Organizations should adopt a tailored maturity model to guide the structured implementation and continuous improvement of emerging technologies within Agile practices. This ensures a consistent and scalable approach aligned with organizational goals.

Performance Metrics: Clear metrics should be in place to evaluate Agile team performance and product outcomes. It is essential for organizations to have appropriate tools and indicators in place to measure the impact of technology integration. These metrics should support the analysis of progress, effectiveness, and achievement of defined objectives.

Continuous Improvement: Structures for regular retrospectives and feedback loops should support ongoing improvement. Organizations need to establish clear timelines and standardized methods to assess performance, identify gaps, and refine technology-enabled Agile processes. Regular assessments promote sustained improvement and adaptability over time. If an organization demonstrates strength across these three areas, it is more likely to adopt Agile successfully and with less resistance. Conducting a readiness assessment prior to implementation provides clear visibility into organizational gaps and helps set priorities for targeted change. A visual workflow of this readiness evaluation process is shown in Figure 16, and a full checklist of readiness factors is included in Table 8.

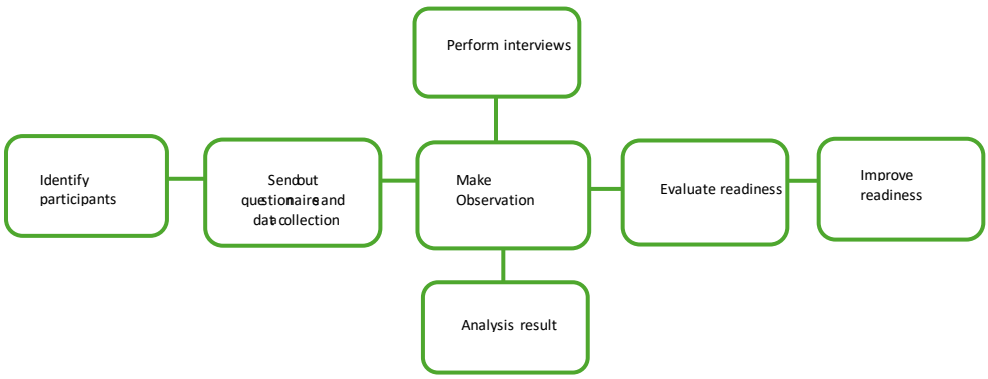

Figure 16 : Readiness Assessment workflow for the Adoption of Emerging Technologies in Agile own illustration)

The readiness assessment for integrating emerging technologies into Agile software development project should be supervised by a designated Agile Transformation Officer Table 8 provides a structured overview of key assessment areas, a set of guiding questions for evaluation, and improvement measures for enhancing readiness in each domain. The following steps detail the proposed readiness model.

1. **Identify Participants**

This is the starting point of the readiness assessment process. It involves selecting stakeholders who will contribute insights about the organization's current capabilities, culture, processes, and attitudes toward change. Participants may include Agile team members, managers, technical leads, and other relevant staff. e.g. Leadership and Management Decision-makers who define the Agile strategy and oversee technology adoption Agile Teams Developers, product owners, and engineers directly involved in Agile workflows Business Stakeholders Representatives from different business units ensuring alignment with strategic objectives Technology Experts. Specialists who evaluate the impact of emerging technologies and agile methodologies

2. **Send Out Questionnaires and Data Collection**

Structured questionnaires are distributed to the identified participants. These questionnaires typically contain targeted questions designed to measure readiness across several dimensions (e.g., strategy, culture, process maturity). Once participants are identified, the next step is to collect the required data through various evaluation techniques Agile Strategy & Governance Assessed via leadership interviews and analysis of Agile governance frameworks Technology Readiness & Adoption Culture Measured using organization-wide surveys and sentiment analysis Agile Workflow & Process Adaptability Evaluated through observations of Agile ceremonies and backlog management practices. To ensure a comprehensive assessment questionnaires, interviews, and observations should be conducted.

3. **Perform Interviews and Data Analysis**

In-depth interviews are carried out with selected stakeholders to collect detailed information. These interviews help explain answers from the questionnaire and give a better understanding of people's opinions, beliefs, and how things work in the organization things that surveys might not fully capture. Once all the data is gathered, the results are reviewed. This scores for each area Agile governance, how flexible the processes are, and how well new technologies are used are averaged and combined to show how ready the organization is. Tables 7 provide the classification criteria.

4. **Make Observations**

This involves directly observing teams, workflows, tools, or meetings within the organization. Observations offer a real-world view of Agile practices and the integration (or lack thereof) of emerging technologies.

5. **Analyze Results**

Once data collection is complete, the responses and observations are analysed. This includes identifying strengths, weaknesses, gaps, and readiness indicators in the organization's current state.

6. **Evaluation Readiness**

This step involves synthesizing the findings from the analysis into a clear readiness evaluation. Based on the assessment results, organizations must determine whether they are ready to integrate emerging technologies into Agile software development project workflows or if there are areas that need further work.

It is recommended to achieves at least the "Ready" classification before proceeding; however, reaching the "Prepared" level has ensures a smoother implementation with reduced risks, increased adaptability, and better alignment with Agile principles

7. **Improvement Readiness**

If gaps are identified, targeted improvement actions are taken. This might include training, strategy refinement, leadership alignment, or process optimization. After improvements, the organization can re-evaluate readiness to confirm progress. Organizations should focus on improving specific areas before integrating emerging technologies into Agile software development project. As highlighted in the literature review, merely intending to adopt these technologies is insufficient organizations must develop the necessary capabilities across all relevant domains. If awareness and attitude levels are low, steps like employee training, team-building activities, and support for managing change should be put in place to help people better understand and accept the changes. Each readiness factor is linked to one or more improvement measures that can help increase the respective scores. These measures serve as actionable steps for organizations to enhance their preparedness before reevaluating their readiness. The suggested improvement measures are detailed in the readiness assessment model (Table 8).

4.2.3 Agile Implementation

This is the core execution phase where Agile principles come to life. The team follows an iterative development process, executing sprints, integrating continuous deployment, and holding daily standups. Rather than launching everything at once, Agile breaks the process into smaller, manageable chunks called sprints. These sprints help keep the work on track, and if anything needs to be improved or fixed, it can be taken care of quickly. This stage ensures that development remains flexible, adaptive, and responsive to evolving requirements. Agile implementation is all about bringing Agile principles to life in software development and project management. It's a continuous cycle of planning, executing, assessing, and refining to improve efficiency, flexibility, and collaboration. The process starts with Pre-Implementation: Transformation & Readiness, where teams lay the groundwork for Agile adoption. This includes defining Agile governance, strategy, and sprint planning while ensuring strong communication and collaboration. A readiness assessment helps identify key stakeholders, gather data, analyze feedback, and evaluate whether the team is prepared for the transition. Once the foundation is set, the team moves into Implementation: Sprint Execution, where Agile principles are put into action. Sprint execution involves breaking down work into smaller, manageable tasks with clear deliverables. Continuous Integration & Deployment (CI/CD) streamlines software delivery, making releases faster and more reliable. Iterative development allows teams to release functional updates frequently, while daily stand-ups help track progress, address challenges, and adjust priorities as needed. After execution, it's time for Post Implementation: Assessment & Improvement, where the team evaluates performance and identifies areas for growth. Key activities include tracking team performance metrics, conducting retrospective analyses to learn from past sprints, assessing team velocity to measure productivity, and gathering feedback to improve future iterations. This phase helps teams make data driven decisions and refine their Agile processes. The final and ongoing phase is Continuous Improvement, where teams focus on evolving and optimizing their Agile practices. This means refining the backlog, adopting new technologies like AI and automation, streamlining processes, and enhancing team collaboration. By always learning and adjusting, Agile teams can build high quality products while staying flexible and ready to handle change. And 2.1.3 the agile implementation

4.2.4 Performance Assessment for Emerging Technology in Agile Software

In an Agile software development project, performance assessment when using emerging technologies is an ongoing process. It's all about tracking how the technology is performing, getting feedback, and making improvements with every sprint. By regularly checking how the technology is affecting progress and making changes based on what they learn, teams can make sure new tools actually help the project succeed and lead to the best results.

1. **Metrics Collection**

This process begins with collecting qualitative data such as team velocity, lead time, defect rates, and automation coverage. These metrics establish a factual baseline for assessing Agile team performance and identifying potential areas of improvement. When using emerging technologies like AI, machine learning, or blockchain in an Agile project, tracking performance is crucial. Teams collect metrics such as deployment success, system stability, and specific technology KPIs (e.g., AI model accuracy or cloud performance). These metrics help measure how well the new technologies is working and if it's meeting the goals set out at the start.

2. **Retrospective Analysis**

After data is collected, a retrospective analysis is conducted. This ensures that data is explained in context and supplemented by team insights. After each sprint, the team reviews how well the emerging technologies were integrated into the project. In a retrospective, they discuss what worked well, what didn't, and how new technology like AI-powered tools or cloud solutions impacted the progress. If there were issues like technology not performing as expected or challenges with team adoption these are addressed so that adjustments can be made for the next sprint.

3. **Team Velocity & Performance Review Assessment**

Based on the retrospective discussions, the team defines specific improvement measures. These may involve adjusting workflows, introducing new tools or practices, improving team coordination, or providing targeted training to address capability gaps. Agile projects measure how quickly teams are completing tasks, called velocity. When emerging technologies are involved, velocity can be influenced by the time it takes for the team to get up to speed with new tools or models. By tracking velocity, the team can identify if the technology is helping or hindering their progress, and they can adjust upcoming sprints to make sure the technology is enabling, rather than slowing down, progress.

4. **Feedback Evaluation**

This proposed improvement measures are then reviewed and validated through team feedback. The collaborative step ensures that the planned changes are realistic, valuable, and supported by those involved in their execution. Getting feedback from the team, stakeholders, and end users is key to understanding whether a project is on track. This is especially important when working with new technologies, as their real impact needs to be constantly evaluated. A structured maturity model helps spot issues and determine if goals are being met. For example, is blockchain making security better? By comparing results to expectations, teams can identify areas that need work and figure out why. Clear communication of these insights helps everyone stay aligned and make better decisions, ensuring continuous improvement.

5. Define Continuous improvement measure adjustments and refinements

Finally, the team reassesses its performance by measuring team velocity and other relevant indicators after the improvements have been implemented. This step evaluates the effectiveness of the changes and provides insight into the overall impact on team productivity and Agile maturity. This framework incorporates a continuous feedback loop, allowing teams to revisit and refine their practices regularly. It reflects core Agile principles such as iterative improvement, team collaboration, and responsiveness to change, while ensuring that the adoption of emerging technologies contributes meaningfully to performance enhancement. To successfully implement any initiative, it's essential to keep refining and improving along the way. If certain goals aren't being met, the team needs to investigate why that's happening and make changes in the next cycle. Like how Agile teams assess progress after each sprint, organizations should regularly evaluate whether their strategies and technologies are working as planned. For instance, if a machine learning model isn't delivering good results, it might need to be tweaked or replaced with a more effective approach. If the reasons behind missed goals aren't clear, it's important to revisit readiness factors, such as employee engagement and awareness. Keeping people motivated and aligned with the initiative is crucial for long-term success, though it can be difficult to measure. If multiple initiatives are falling short of expectations, it could signal that the strategy or maturity model needs adjustment. In such cases, it may be necessary to restart the process with a new strategy that aligns better with the company's changing needs.

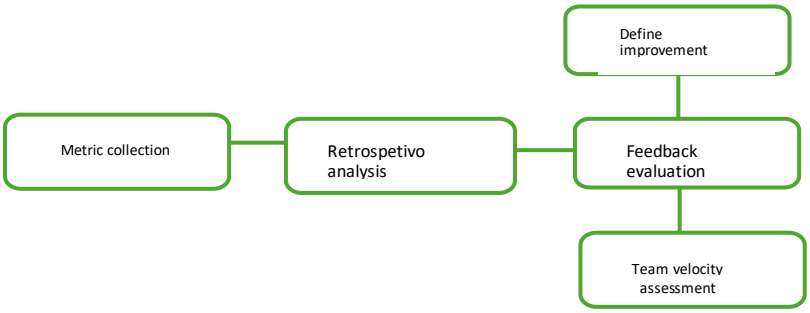

Figure 17: Performance Assessment for Emerging Technology in Agile Software own illustration)

The framework includes an iterative feedback loop, were performance evaluation feeds into continuous improvement. The flow between defining improvements, evaluating feedback, and reassessing team velocity helps maintain alignment with Agile principles like:
Responding to change over following a plan, continuous delivery and process improvement, Empowered, self-organizing teams. And the purpose of the Framework is to perform evaluation framework that ensures that Agile teams measure the real impact of emerging technology adoption, Retrospectives are data-driven and actionable, Improvements are evaluated from both metrics and team perspectives, Continuous improvement becomes a formalized process.

4.2.5 Continuous Improvement in Agile Implementation

Continuous improvement is a core part of Agile and key to keeping it successful over time. After starting Agile practices, teams should keep reviewing and improving how they work. This includes getting feedback, learning from past work, and staying flexible. Key actions include holding team retrospectives after each

sprint, tracking performance (like speed and team happiness), and making changes based on lessons learned. Regular training and sharing ideas between teams also help everyone grow and improve together.

4.3 Use Case

This part presents a fictional use case that focuses on how new technologies like AI, IoT, and Blockchain can be used and managed in an Agile software development project. Since it's a fictional scenario, the tools and methods mentioned are just examples they aren't fully tested or real implementations. We don't go into all the details of Agile processes like planning or sprint reviews as deeply as you'd see in a real project. But this example helps us take a first look at how the framework could work in a practical setting. In the future, we hope to test the framework with real projects and get feedback from experts to see how well it guides Agile teams working with emerging tech. Since a real case study is outside the scope here, we're using this example to show how the framework is meant to be used and to check if it's helpful. Creating a full maturity model tailored to one company would take a lot more work, so this is just a starting point to see how things fit together.

Fictional Example of Agile software development

A TechNova Solutions is a medium-sized software company that creates custom apps for healthcare and logistics. With offices in three cities, they want to update their development methods to stay competitive and better meet client needs for faster and smarter solutions. First, TechNova sets clear business goals like shortening development time, improving customer service, and boosting data security. They then use guidelines to decide which new technologies such as AI for automation, blockchain for secure transactions, or IoT for real-time data can help reach these goals within an Agile development process. This way, the company makes sure that using new technology supports their important business objectives.

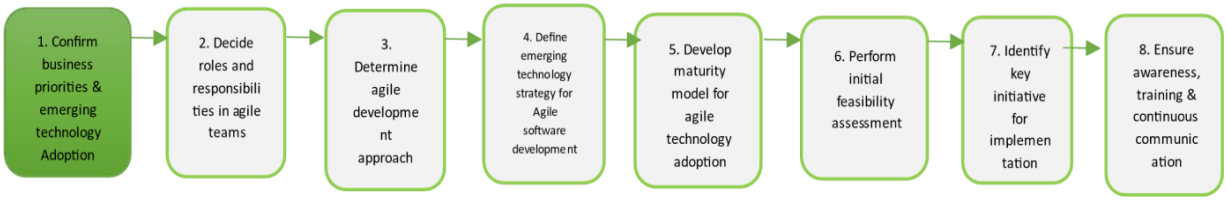

Figure 18: Use case step one

The company promotes the original idea creators to lead roles naming one as the Agile Technology Integration Lead. A cross-functional team is then formed with members from all office locations. Roles are clearly defined Product Owners, Scrum Masters, developers, and tech experts like AI/ML specialists, IoT engineers, and DevOps leads. A dedicated budget is also set, giving the lead the freedom to manage resources and guide the integration of Agile practices and new technologies smoothly across all teams.

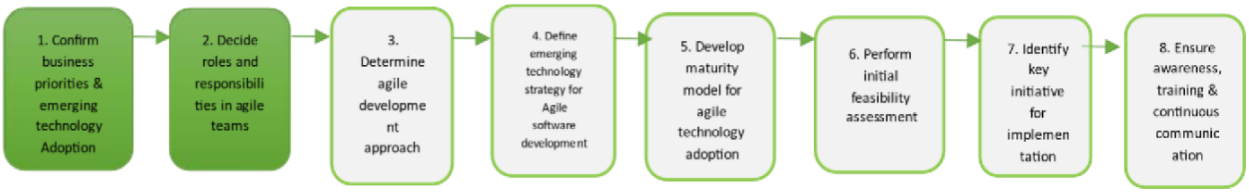

Figure 19: Use case step two

Once the basic setup is ready, the technology lead at TechNova starts looking into different Agile methods to see which one fits the company best. Before creating a full plan, they review how the company is organized and what kind of projects it handles. This helps decide whether Scrum, Kanban, or SAFe would work better. The goal is to choose a method that fits the team size, project complexity, and the challenge of using new technologies like AI, blockchain, or IoT. Using the framework as a guide, TechNova picks an Agile approach that offers the right mix of

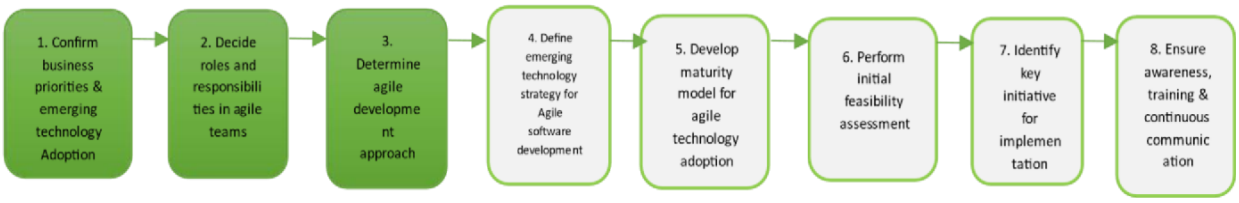

Figure 20: Use case step Three

A formal strategy is developed to guide how emerging technologies will be integrated into the Agile cycle. This includes defining short- and long-term goals, selecting pilot technologies (e.g., AI for customer support automation), and identifying dependencies like training, infrastructure, or vendor partnerships. Outline how emerging technologies (like AI, IoT, Blockchain) will be integrated into Agile processes, including timelines, tools, and expected outcomes.

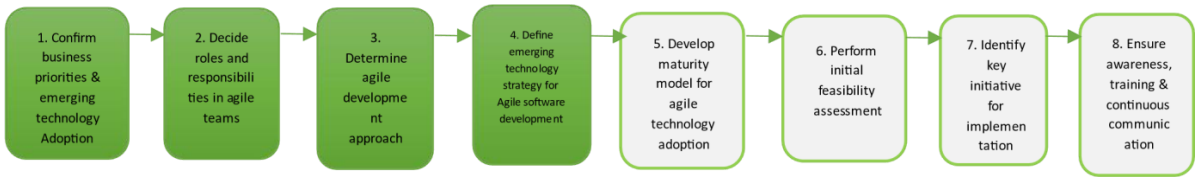

Figure 21: Use case step four

As part of the new strategy, TechNova's Agile Technology Lead and task force begin checking how ready the organization is to adopt Agile practices and emerging technologies. They use a maturity model from the guidelines to review important areas like team skills, infrastructure, available tools, and how familiar people are with Agile methods. To measure progress, they set up a structured maturity model based on proven frameworks. This model is used every two months to check growth in areas like AI adoption, DevOps automation, and Agile team performance. The goal is to improve by at least one level in each area with every review. After each check, the team updates the priorities for current and future projects to stay on track with the company's Agile transformation goals.

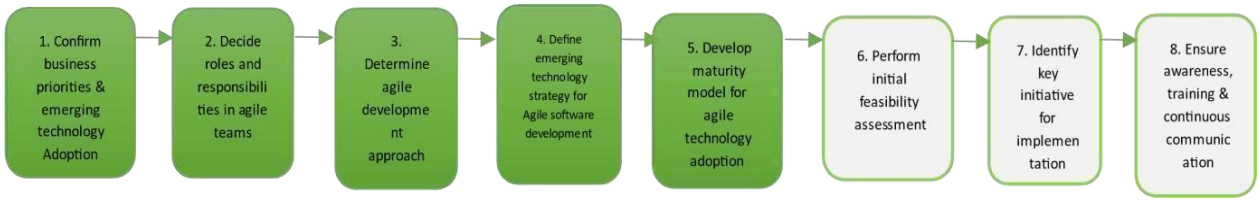

Figure 22: Use case step five

After the maturity assessment, the company runs small pilot projects to test new technologies. Feedback is collected on how well the technology works, how easy it is to use, and how the team adapts. This helps decide if the technology can grow within the Agile setup. The company also checks if the technology is realistic to implement by looking at technical, financial, and organizational factors.

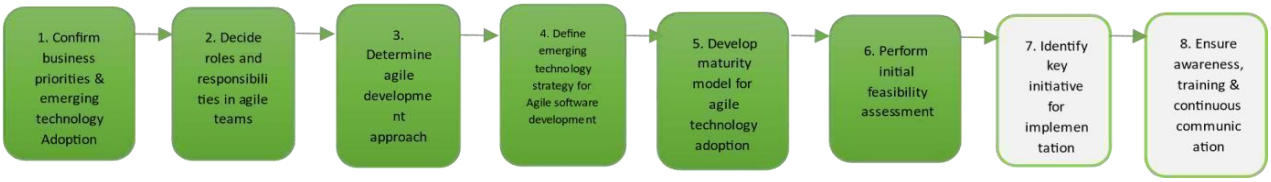

Figure 23: Use case step six

Based on the feasibility results TechNova picks key projects like adding AI to testing or using blockchain for data security. They focus first on saving toner and paper since the company uses a lot. This is a valuable project that's not too hard. They also plan to promote the work-from-home system, which many don't know about. This is an easy, quick win. Other projects like buying energy-saving devices take more effort, so they come later. The team will start with printing and telecommuting improvements to get quick results and then move on to bigger projects. They **will test one project first to make sure the plan works.**

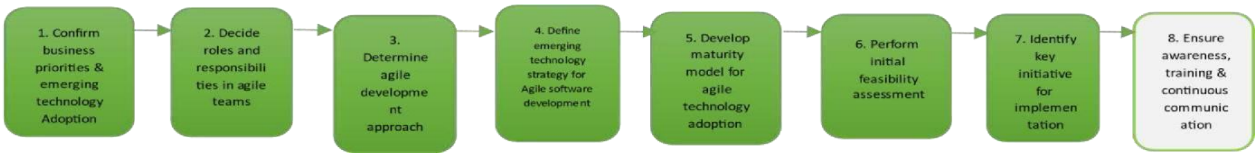

Figure 24: Use case step seven

The final step focuses on supporting long-term adoption through clear communication, ongoing learning, and strong stakeholder involvement. To help with this, the team will give a presentation at the next company meeting. It will explain the new strategies, the first steps to take, the goals, and when things will happen. The company will also offer regular training, show progress through sprint demos, hold retrospectives, and ask for feedback. This helps everyone stay on the same page, learn what they need, and adjust as new technologies are added to Agile work. To make the change work well, it's important to raise awareness, give the right training, and keep communication open.

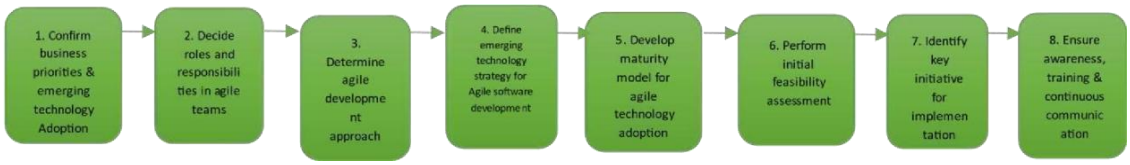

Figure 25: Use case step eight

Since the earlier activities to integrate emerging technologies into the Agile project were successful, no changes are needed for those measures. Following Agile principles, the team now starts a new planning cycle. They come up with new ideas, rank them by value and effort, and choose what to do next based on that analysis. The results from the completed phase and the chosen priorities for the next phase are shared with all key stakeholders. This marks the start of the next stage in the implementation of emerging technologies, as shown by the orange arrow in Figure 26

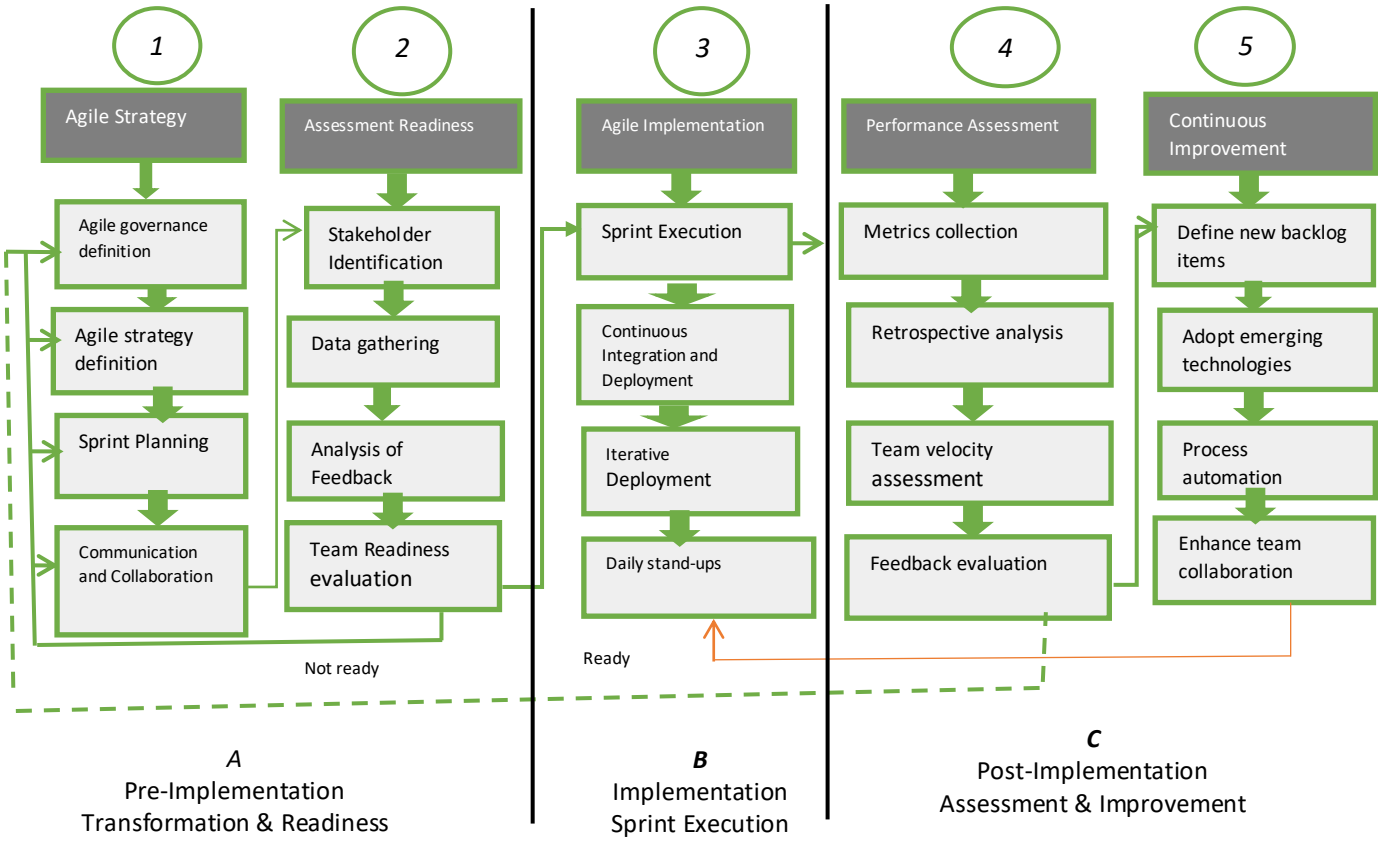

Figure 26: Completed emerging technology in agile software implementation process own illustration)

5. Evaluation and Discussion

After creating the guidelines for using emerging technologies in Agile software development projects, the research moved to the fourth stage of the Design Science Research (DSR) process, the evaluation stage. As mentioned in Section 3.2, the study followed the DSR model by Peffers et al. (2007). To evaluate the guidelines, feedback was gathered from experts using a qualitative approach. Semi-structured interview was held online with professionals who had experience in Agile and emerging technologies. The goal was to check if the guidelines were useful, identify their strengths, and find areas for improvement. Experts from both industry and academia were chosen to get a well-rounded view, since each group brings different experiences and insights. Before the interview, participants were given a short presentation that explained the background, purpose, and content of the guidelines. This helped ensure everyone had a clear understanding of what was being discussed. The presentation guided the conversation and helped experts give relevant and thoughtful feedback. The table below shows the experts who took part, along with their roles and areas of expertise:

Table 6: List of selected experts, including their professional roles and areas of expertise

Professional Role	Area of Expertise	Domain
Professor Pedro Maia Malta of information system	professorship	Academia
Oreva Pius Enterprise Business Applications	Head Information Technology	Industry
Robert DevOps Engineer @ Anchor Datarem solution	DevOps Engineer professional	Industry

The interview and resulting insights contributed significantly to validating the guidelines and filling previously identified research gaps. Constructive suggestions were noted and, where appropriate, incorporated into the final version of the proposed framework. Prior to the interviews each participant was provided with an interview guide that included the research objectives, the proposed guidelines for the use of emerging technologies in Agile software development projects, the implementation framework, and the readiness model. The guide also contained four structured questions designed to elicit expert feedback on the applicability, clarity, and effectiveness of the proposed guidelines.

Q1: To what extent do you consider the proposed framework to be useful? Please provide reasons for your response.

Q2: Are there any aspects of the proposed framework you would critique or question? Kindly explain your perspective.

Q3: Would you consider adopting or applying the proposed framework in your context? Please clarify your reasons for supporting or declining its implementation.
Q4**:** What recommendations or suggestions do you have for enhancing or refining the proposed framework?
The interview questions were crafted to provide a structured yet flexible approach that guided experts in critically evaluating the framework to support the integration of emerging technologies into Agile software development projects. Establishing a consistent thematic framework allowed for systematic comparison of responses and the identification of shared perspectives, while the flexible interview format facilitated the exploration of unique insights from each participant. This method also enabled the clarification of open questions related to the framework and readiness model, ensuring a comprehensive understanding of the proposed approach. All three experts agreed that the framework is a valuable tool for supporting the integration of emerging technologies such as Artificial Intelligence, Blockchain, and the Internet of Things within Agile organizations. They collectively emphasized the framework's ability to bridge gaps in current practices by offering a structured, comprehensive, and adaptable approach to technology adoption across various organizational contexts. Expert 1 found the framework particularly useful for characterizing organizations at the outset of technology integration initiatives. This initial analytical capability was seen as critical for tailoring implementation strategies to specific organizational needs. The expert also highlighted the framework's academic value, noting its contribution to the existing body of knowledge, especially in addressing practical gaps in Agile and emerging technology integration. Expert 2 strongly supported the framework's systematic alignment of emerging technologies with Agile methodologies traditionally studied in isolation arguing that this integration provides a richer, more applicable approach. The expert highlighted the framework's potential to inform strategic decision making by evaluating technical feasibility, organizational readiness, and alignment with business objectives. Moreover, the expert underscored the framework's adaptability across by including Agile methodologies like Scrum, extreme programming and Lean the framework becomes more relevant and easier to apply in a variety of real-world situations Expert 3 also endorsed the framework, particularly its alignment with core Agile and DevOps principles, such as flexibility, automation, and continuous improvement. The expert found the phased structure ranging from strategic alignment to performance assessment well-suited to the dynamic nature of DevOps environments, where iterative delivery and adaptability are paramount. The inclusion of readiness assessment and continuous improvement cycles were noted as especially valuable for ensuring measurable and sustainable technology adoption.
Despite their strong support, the experts offered thoughtful criticisms and actionable suggestions for refinement. Expert 1 recommended incorporating visual elements such as flowcharts or diagrams to aid in the orientation, implementation, and monitoring of the framework. Visual tools, according to the expert, would assist in communicating the phased structure more effectively and in reinforcing the concept of technology adoption as a continuous, structured process rather than a collection of isolated tasks. Expert 2 identified several opportunities to enhance the framework's comprehensiveness. These included the need for more explicit strategies to manage organizational change, address skill gaps, and support the scaling of emerging technologies in small business contexts. The expert also called for greater emphasis on Ethical concerns including data privacy, algorithmic fairness, and system transparency and clearer metrics to assess return on investment (ROI) following technology implementation. Furthermore, the expert noted the importance of embedding cultural and change management components into the framework, as these often represent significant barriers to successful adoption. Similarly, Expert 3 recommended supplementing the framework with more real-world case studies. These would allow teams to visualize specific implementation challenges and adaptations across industry contexts, strengthening practical understanding and increasing confidence in the framework's application. While acknowledging the inclusion of a use case in the study, the

expert suggested that multiple context-specific examples would further enhance the framework's utility. All three experts expressed willingness to implement the framework in practice. Expert 1 recognized the framework's contribution to academic discourse, especially in addressing the practical challenges of aligning emerging technology adoption with organizational realities. Expert 2 appreciated the fact-based and research driven foundation of the framework, noting its grounding in
Technological Determinism theory as an important element for understanding broader impacts on organizational processes. Expert 3 emphasized the framework's compatibility with DevOps practices and the value of its structured, phased approach for ensuring scalability and long-term performance. In terms of recommendations for further development, the experts offered several aligned suggestions. Expert 1 proposed initially applying the framework within small to medium-sized enterprises (SMEs), given their relatively simpler structures. This would allow for easier application and refinement before scaling to larger, more complex organizations. Expert 2 suggested conducting a "blind demo" or pilot application of the framework to assess real time organizational readiness for specific technologies and to map progressive implementation stages. The expert also recommended developing a structured change management strategy to overcome cultural resistance, implementing a continuous feedback loop to support iterative improvements, and creating targeted training programs to address skill gaps during technology adoption. Expert 3 called for the explicit integration of continuous security practices such as DevSecOps into the framework. Given the increasing significance of cybersecurity in Agile contexts, this addition was seen as vital for ensuring a future ready, resilient system. The expert also advised expanding the framework to consider hybrid cloud and edge computing environments, making it more comprehensive and robust considering ongoing technological evolution. Additionally, Expert 3 proposed introducing a baseline assessment stage before the maturity evaluation begins. Establishing a current-state benchmark would allow organizations to track progress and assess return on effort and investment over time. The expert also recommended refining the terminology used in the readiness model specifically, replacing "education" with "training" to better reflect the practical, job specific focus of capacity building activities in real-world Agile environments.
Finally, two common improvements were independently suggested by all experts. First, they recommended simplifying the readiness assessment tool by replacing the current rating scale with binary yes/no questions. This change is expected to reduce cognitive load, improve clarity, and provide more actionable insights. Expert 2 emphasized that such simplification enhances consulting engagements by reducing ambiguity and encouraging honest self-assessment, which is essential for planning and resource allocation. Second, the experts emphasized the importance of validating the framework through practical application through real-world or simulated case studies. Such empirical testing would demonstrate its practical utility, allow for iterative refinement, and support broader replication by other organizations seeking to adopt emerging technologies within Agile frameworks. This integration could enhance the framework's rigor and align it more closely with globally recognized standards for continuous improvement, sustainability, and operational excellence.
In conclusion, the expert panel include Expert 1, 2, 3 collectively endorsed the proposed framework and readiness model as effective and forward-looking tools for guiding the adoption of emerging technologies in Agile software development contexts. They unanimously advocated for adjustments to the readiness assessment and emphasized the value of further empirical testing through real-world application. At the same time, they urged maintaining a careful balance between depth and usability to maximize both adoption and impact. As a result of this feedback, the five steps for readiness assessment described in subsection 4.2.2 have been slightly refined and are briefly described again in the next section.

5.1 Revised Model

Using the feedback from the conducted expert interview, the readiness model and its assessment were revised from using a rating scale towards using a nominal scoring of yes/no.

Table 7: Revised rating scale and readiness classification

Score	Response	Classification
0	No	Not Ready
1	Yes	Ready

The adapted readiness model is presented in Table 8. To support a structured and effective assessment process, both the evaluation method and the responsible participants were revised. All readiness factors, except for the four people-related dimensions Knowledge, Awareness, Attitude, and Human Behaviour and Culture should be assessed objectively by the designated Technology Sponsor or implementation lead, without the need for additional stakeholder involvement (e.g., interviewers or group workshops). Conversely, the three people related factors should be evaluated through a structured questionnaire distributed to a large and representative sample of employees across the organization. This approach ensures broad coverage and provides a comprehensive understanding of employee readiness at scale. Upon completion of data collection, each factor should be classified as either *Ready* (“yes”) or Not Ready (“no”). While this qualitative threshold offers consistency, the final determination of readiness may ultimately rest with the Technology Sponsor based on their contextual judgment and overall assessment.
Above all, the evaluation process should prioritize honest, evidence-based self-reflection. The outcomes of the assessment suggest a potential need to put emerging technologies into practice.
Five steps for readiness assessment described in subsection 4.2.2 are therefore slightly adapted and quickly described again now

1: Identify Participants

Determine a representative sample of team members from different roles, departments, and hierarchy levels within the Agile organization. This diverse group will provide input on humancentric readiness dimensions such as knowledge, awareness, attitude, and culture. Remaining readiness factors related to processes and technology should be assessed by the designated Agile coach or emerging technology lead.

2: Data Collection

capture clarity and minimize ambiguity. Parallelly, conduct internal evaluations of non-human factors by the responsible lead, following the same binary response format.

3: Analyze Data

Upon completion of the data collection phase, the responses to the questionnaire must be analysed, focusing specifically on the four human-centric readiness factors: Knowledge, Awareness, Attitude, and Human Behaviour and Culture. For each factor, the percentage of affirmative ("yes") responses should be calculated.

A minimum threshold of 85% “yes” responses is recommended for a factor to be classified as “ready.”

4: Evaluate Overall Readiness

Following the analysis of assessment results, each readiness factor should be individually reviewed to determine whether the organization is sufficiently prepared to proceed with the implementation of emerging technologies. A factor is considered ready only if it receives a definitive “yes” response. To enhance clarity and facilitate communication across Agile teams, visual tools such as color-coded indicators may be applied to represent readiness levels and highlight variations across dimensions. Additionally, calculating and aggregating the number of affirmative responses within each readiness dimension can provide a clear overview of organizational preparedness and support informed decision-making during the planning phase.

5: Plan and Execute Improvements

Improve readiness (if needed, depending on results from step 4). The outcomes of the assessment suggest a potential need to address and improve certain areas before proceeding with the implementation of emerging technologies. Effectively resolving these identified gaps is essential to ensure a successful adoption process. Additionally, feedback regarding the need for a dedicated step to evaluate the current baseline state and the suggestion to replace the term “education” with “training” in the final step of phase one have been incorporated. These adjustments are detailed in Section 4.2.1, where Steps 6 and 7 have been introduced. The updated framework, including these modifications, is also reflected in the extended fictional use case presented in Chapter 4.3, which now encompasses the complete framework beyond the initial readiness assessment.

Table 8: Revised readiness model

Dimension	Readiness Factor	Evaluation	Participants	Exemplary Question/s	Score	Improvement Measures
Strategy & Governance	Objectives	Observation / Interview	Agile Coach, Management	Are the organizational objectives for integrating emerging technologies in Agile development project clearly defined?	Yes/No	Define and communicate objectives for adopting emerging technologies in agile software development
	Motivation	Observation / Interview	Agile Coach, Management	Is the motivation for adopting emerging technologies clearly stated and aligned with Agile project goals?	Yes/No	Define and communicate the strategic organizational motivation for emerging technologies in agile software development.
	Strategy	Observation / Interview	Agile Coach, Management	Is the strategy for incorporating emerging technologies into Agile project workflows well-defined and aligned across teams?	Yes/No	Develop a clear roadmap for adoption and integration emerging technologies and agile project
	Impact	Observation / Interview	Agile Coach, Management	This is the expected impact of these technologies articulated and communicated effectively?	Yes/No	Conduct impact analysis and communicate benefits in emerging technologies and agile project

	Budget	Observation / Interview	Agile Coach, Management	Has a budget that been allocated specifically for technology adoption in Agile projects?	Yes/No	Ensure proper funding for technology adoption in agile project
	Roles and responsibilities	Observation / Interview	Agile Coach, Management	Are roles and responsibilities for managing and implementing emerging technologies clearly defined?		Establish clear ownership and governance structures for emerging technologies and agile
	Approach	Observation / Interview	Agile Coach, Management	Has the approach to using emerging technologies been tailored to fit Agile methodologies?	Yes/No	Use Agile-friendly tools and frameworks
	Role of senior management	Observation / Interview	Agile Coach, Management	Is senior management actively supporting the adoption of emerging technologies in Agile?	Yes/No	Ensure leadership involvement and sponsorship for emerging technologies and agile
Awareness & Attitude	Knowledge	Questionnaire	Agile Team Members	To what extent are you informed about emerging technologies in Agile development?	Yes/No	A conduct training and knowledge-sharing sessions
	Awareness	Questionnaire	Agile Team Members	Are you aware of the organization's strategy, motivation, and objectives for adopting emerging technologies?	Yes/No	Improve internals communication and documentation for emerging technologies and agile
	Attitude	Questionnaire	Agile Team Members	Do you believe that integrating emerging technologies enhances Agile development project efficiency and innovation?	Yes/No	Promote a culture of experimentation and innovation for emerging technologies and agile
	Agile Mindset and Culture	Questionnaire	Agile Team Members	How willing are you to adopt and experiment with emerging technologies in Agile project workflows?	Yes/No	Foster Agile coaching and mentorship programs
	Organizational behaviour and culture	Observation	Agile Coach	To what extent does the organization encourage the adoption of emerging technologies within Agile teams?	Yes/No	Encourage open discussions and feedback loops for emerging technologies and agile
Processes	Standardized processes for adoption	Observation / Interview	Agile Coach, Management	Are there defined processes for selecting, experimenting with, and implementing emerging technologies in Agile?	Yes/No	Develop an Agile framework for technology evaluation
	Metrics for assessing impact	Observation / Interview	Agile Coach, Management	To what extent has the organization defined measurable KPIs to monitor the adoption and value generated by emerging technology	Yes/No	Define clear successful metrics for Agile technology adoption
	Continuous Improvement	Observation / Interview	Agile Coach, Management	Is there a structured approach for regularly reviewing and optimizing the use of emerging technologies?	Yes/No	Implement Agile retrospectives focused on technology evaluation

5.2 Result of the Survey Respondent

The survey received responses from 208 participants, including professionals involved in Agile software development such as product owners, scrum masters, developers, testers, and others. These individuals had experience with both traditional Agile practices and the integration of emerging technologies. As such, they provided valuable insights regarding how these technologies are perceived and applied—particularly in terms of reliability, decision-making, real-time performance, potential bias, and alignment with Agile principles.

Q1– Consent to Participate

According to the data gathered on respondents' ages, a large majority of 208 total respondents, 186 participants (89.42%) confirmed they were 18 years or older and participated voluntarily. The remaining 22 individuals (10.58%) provided alternate or unspecified responses.

Observation: The high confirmation rate supports ethical data collection standards and indicates that respondents were adequately informed.

Figure 27: Q1 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q2 – Training on Emerging Technologies

Among 208 respondents: 140 (67.31%) stated their teams frequently attend training on emerging technologies. An additional 41 (19.71%) participated occasionally, 20 (9.62%) rarely trained, 6 (2.88%) never trained, and 1 respondent (0.48%) selected other.

Observation: Most Agile teams actively engage in continual learning, indicating high readiness and openness toward technological advancement.

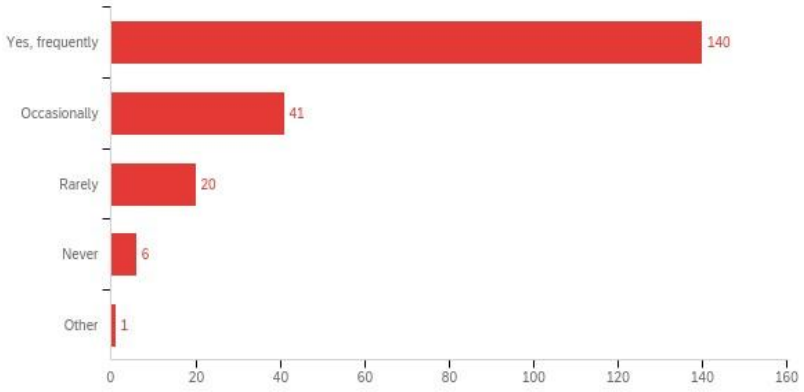

Figure 28: Q2 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q3 – Reviewing Use of Emerging Technologies

Among 208 respondents, 100 (48.08%) stated their teams review emerging technologies after every sprint, 54 (26%) once per project, 21 (10.10%) annually, 25 (12.02%) rarely, and 8 (4%) selected other — indicating that most Agile teams consistently assess their technology use, reflecting a strong focus on iterative improvement and staying current Participants reported a mean score of 2.0 regarding how often their teams review or reassess their use of emerging technologies.

Observation: Regular reviews demonstrate Agile teams' commitment to iterative improvement and relevance in technological practices.

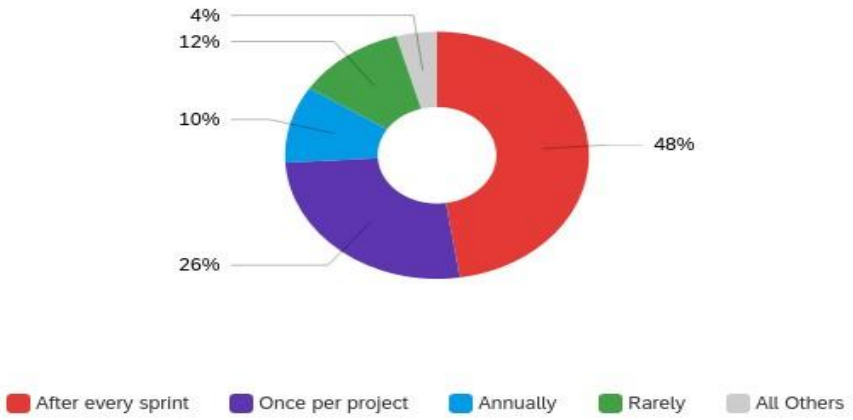

Figure 29: Q3 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q4 – Emerging Technologies Causing Delays

Nearly 203 responses 101(49.75%) stated that emerging technologies frequently caused delays in Agile projects, and an additional 56(27.59%) said this happened occasionally, 24 (11.82%) rarely, and 8 (3.94%) never. 11 (5.42%) were unsure, and 3 (1.48%) selected "Other."

This means that more than three-quarters of teams have experienced project disruptions due to emerging technologies. **Observation:** Over 77% encountered delays, suggesting that integrating emerging tech can challenge Agile

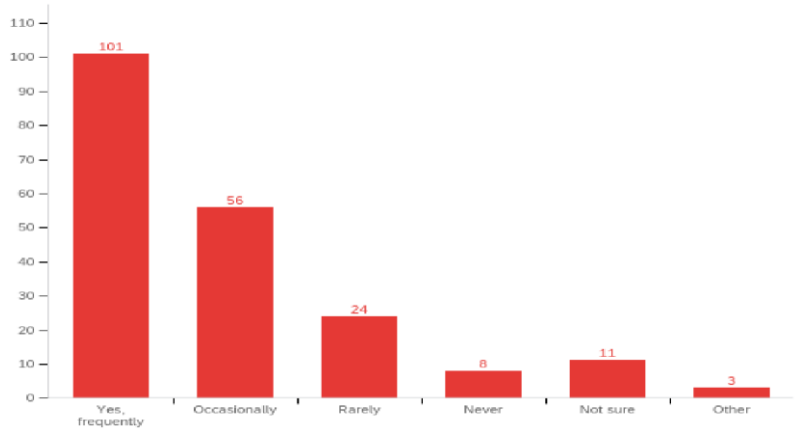

Figure 30: Q4 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q5 – Integration Effectiveness

Among 203 responses: 118 respondents (58.13%) indicated that their teams were very effective at integrating emerging technologies into Agile practices. A further 38 (19%) reported it slightly effective. 17 (8.37%) moderately effective, 13 (6.40%) extremely effective, 10 (5%) not effective at all, and 7 (3.45%) selected "Other.

Observation: Most teams feel confident in integration, but a notable minority may lack the frameworks or support needed.

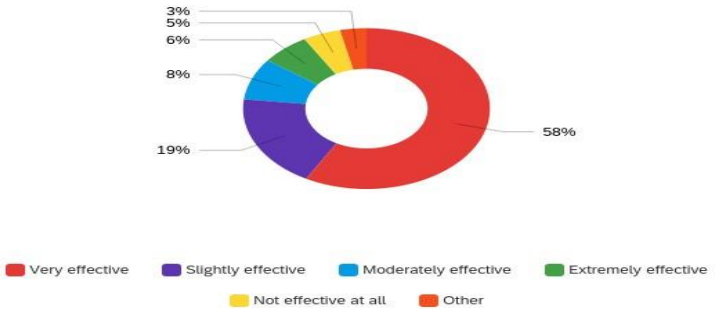

Figure 31: Q5 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q6 – Leadership Support for Innovation

Out of 203: 103 respondents (50.74%) reported that leadership supported technological innovation very well, with another 48 (23.65%) saying slightly well, 21 (10.34%) moderately well, 16 (7.88%) extremely well,9(4.43%) not well at all, and 6 (2.96%) choose Other.

Observation: Leadership is mostly seen as enablers of innovation—critical for Agile success and culture. This points to a generally supportive environment for tech-driven innovation within Agile teams.

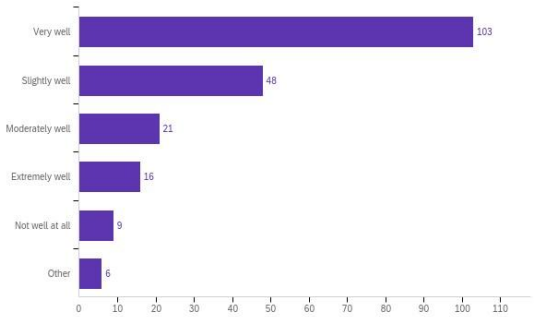

Figure 32: Q6 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q7 – Organizational Digital Maturity

Out of 203 responses: 111 (54.68%) rated digital maturity as very high, 42 (20.69%) slightly mature and others indicated 23 (11.33%) moderately mature, 13 (6.40%) extremely mature, and only 3 (1.48%) reported no maturity. Others indicated very low maturity (9) or other (2).

Observation: Organizations work in digitally mature environments, which aligns with readiness to adopt emerging tech.

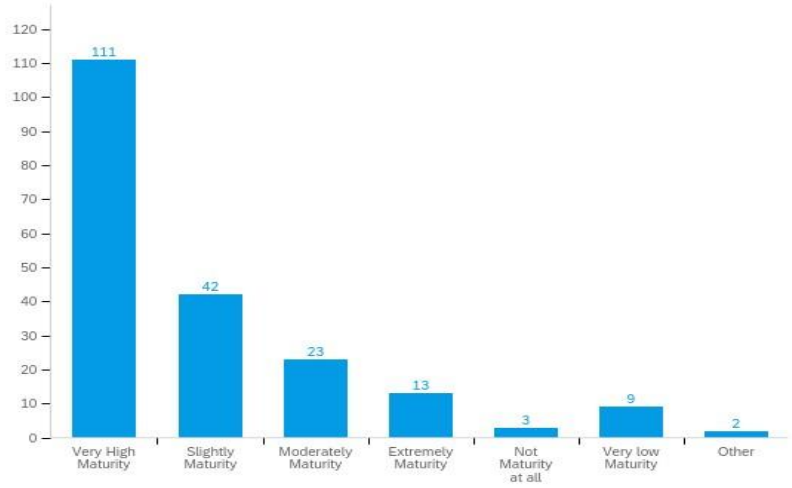

Figure 33 :Q7 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q8 – Confidence in Adapting to New Technologies

Of out 203: 116 respondents (57.14%) were very confident in their teams' adaptability, to new technologies within the next 12 months followed by 41 (20.20%) slightly confident, 17 (8.37%) moderate, 12 (5.91%) extremely confident, 14 (6.90%) not confident.

Observation: Teams express strong confidence in their ability to embrace future technologies, demonstrating proactive adaptability.

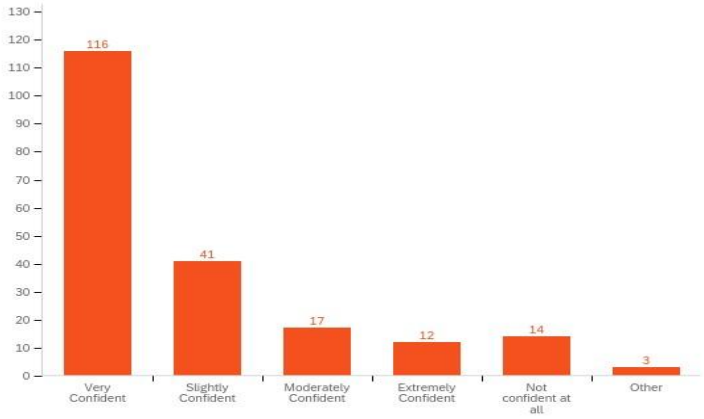

Figure 34: Q8 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q9–Evaluation of Technological Readiness

201 responses: 112 (56%) said readiness is evaluated, while 43 (21.39%) said maybe, and 46 (23%) said no.

Observation: Evaluation practices are common but not universal, which may lead to avoidable disruptions.

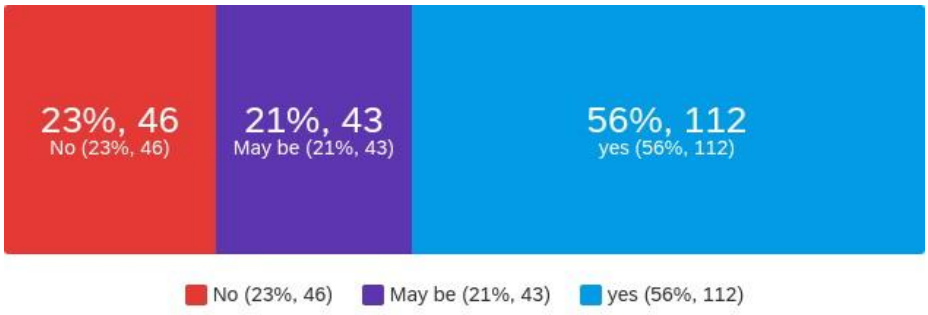

Figure 35: Q9 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q10 – Customer Satisfaction Impact

Out of 203 responses: 131 (65%) felt that emerging technologies enhance customer satisfaction, 48 (24%) said "Maybe," and 24 (11.82%) said "No."

Observation: A strong belief exists in the customer-facing benefits of emerging technology use in Agile development.

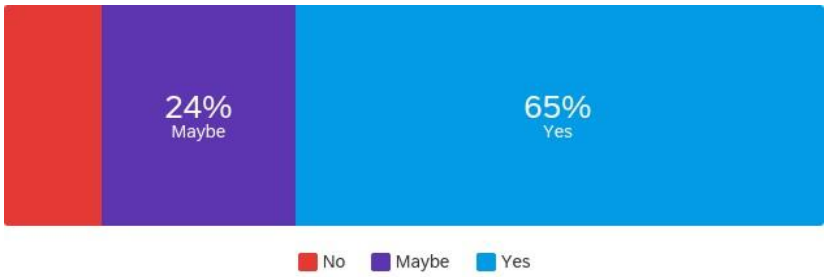

Figure 36: Q10 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q11 – Support for Continuous Improvement

Among 203 responses: 139 (68.47%) believed new technologies support continuous improvement, 55 (27.09%) said "Maybe," and only 9 (4.43%) said "No."

Observation: Teams clearly associate technology adoption with Agile's core principle of continuous improvement.

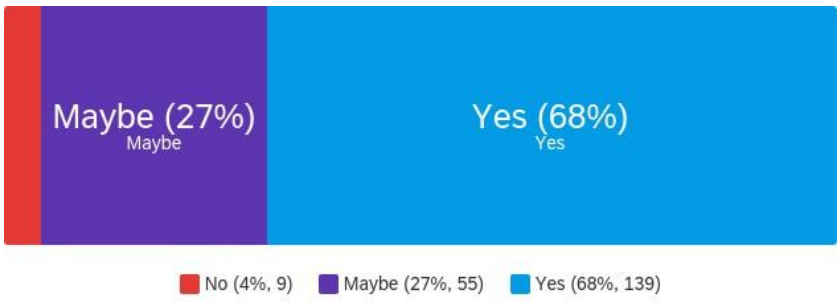

Figure 37: Q11 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

-

Q12 – Alignment with Agile Principles

From 203 participants: 132 (65.02%) said technology aligns with Agile values, 57 (28.08%) said "Maybe," and 14 (6.90%) said "No."

Observation: Most teams see no conflict between Agile principles and the use of emerging technologies.

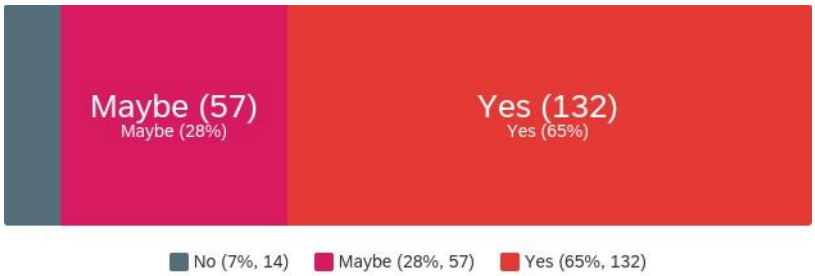

Figure 38: Q12 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q13– Budget for Innovation

203: 138 respondents (68%) confirmed having a dedicated budget for tech innovation, 47 (23.15%) said maybe, and 18 (9%) said no.

Observation: Most Agile organizations financially support innovation.

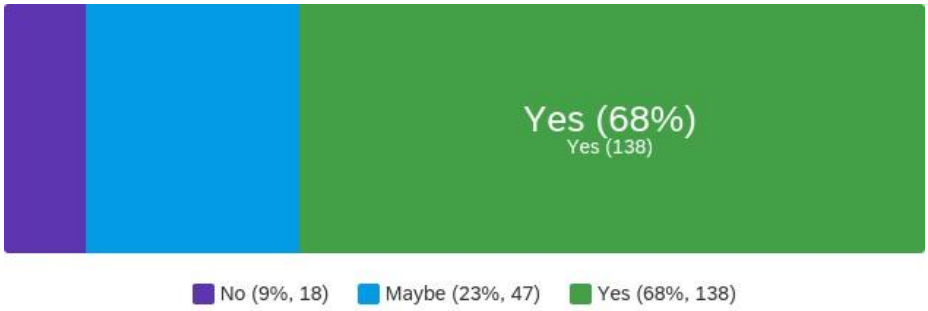

Figure 39: Q13 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q14 – Experience with AI in Agile Projects

Among 203 responses: 133 (65.52%) had worked on Agile projects involving AI, 39 (19.21%) said "Maybe," and 31 (15.27%) had not.

Observation: AI is really a significant component in Agile projects for many respondents.

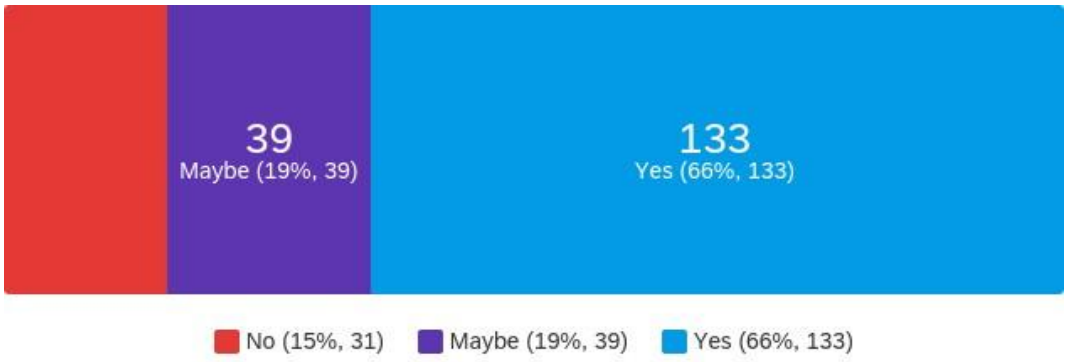

Figure 40: Q14 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q15 – Policies or Frameworks for Adoption

203 respondents: 122 (60.10%) confirmed the presence of defined policies or frameworks, 55 (27.09%) were unsure, and 26 (12.81%) said no.

Observation: A structured approach of adopting new technologies exists in most organizations, supporting sustainable change.

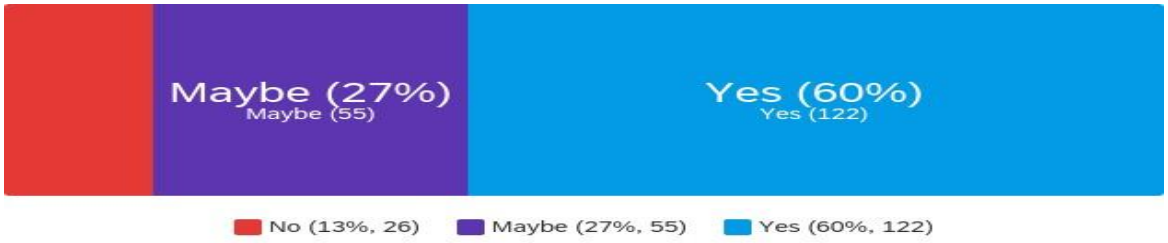

Figure 41: Q15 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q16 – Training on New Technologies

From 203 responses: 70 (34.48%) strongly agreed they received adequate training, 51 (25.12%) agreed their teams had adequate training. Meanwhile, 42 (21%) were neutral, 24 (12%) disagreed, and 7 (3.45%) strongly disagreed. 9 (4.43%) choose "Other."

Observation: Most teams feel prepared, but a training gap still exists that could affect performance.

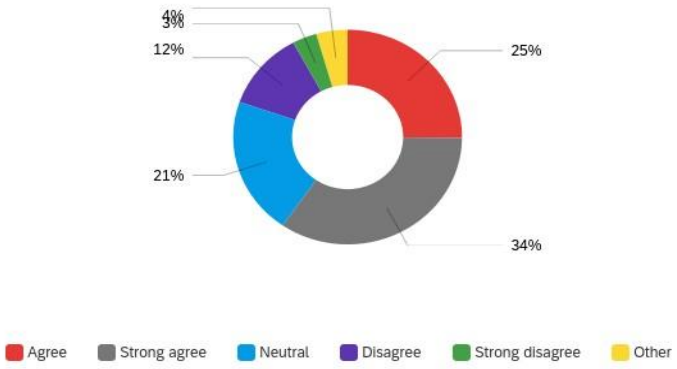

Figure 42: Q16 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q17 – Support for Keeping Up with Tech Evolution

Among 203 respondents: 107 (52.71%) said their organization provides regular support, 30 (14.78%) said support is ongoing, 36 (17.73%) occasionally, while 14 (6.90%) said rarely, 9 (4.43%) no support and 7 (3.45%) reported no support.

Observation: Ongoing organizational support exists but is not universal, which could impact long-term tech adoption success.

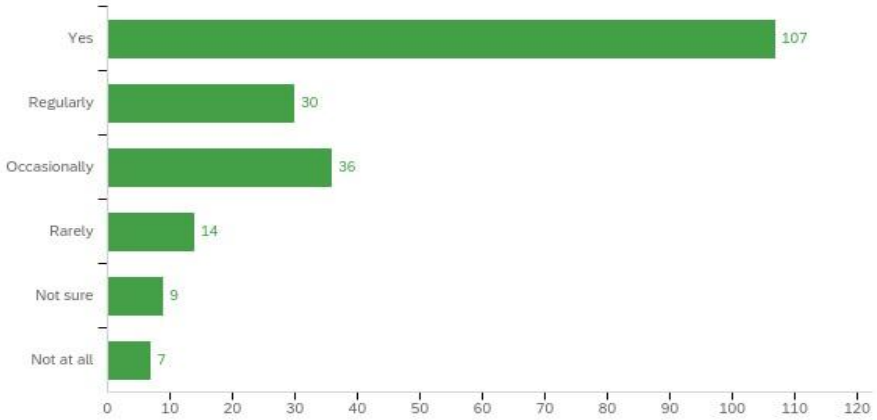

Figure 43: Q17 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

18 – Transparency in Decision-Making

From 203 responses: 115 (56.65%) rated tech adoption decisions as very transparent, 36 (17.73%) slightly transparent, 24 (11.82%) moderately transparent, and 14 each (6.90%) said extremely or not transparent.

Observation: Decision-making around technology is generally open, though not fully transparent in all teams. Transparency is generally strong, which is essential for Agile team trust and alignment.

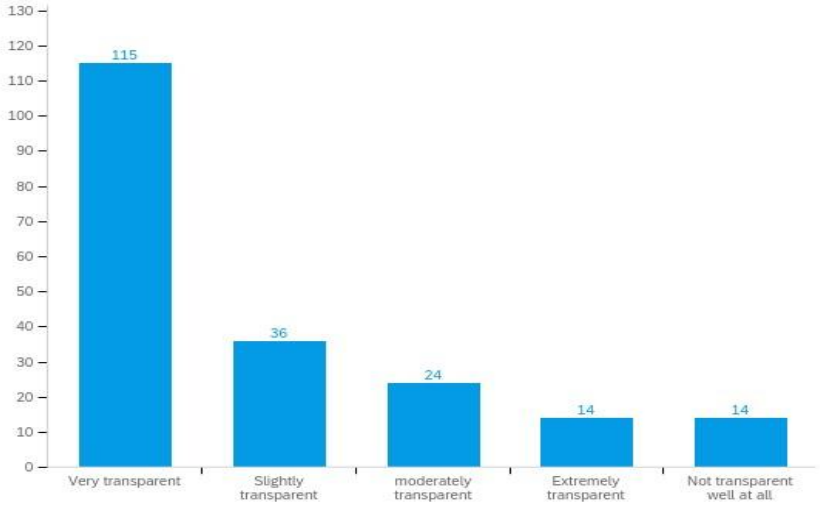

Figure 44: Q18 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q19 – Agile Roles of Respondents

Among 203 participants: 89 (43.84%) were Product Owners, 40 (19.70%) Scrum Masters, followed by 32 (15.76%) Developers, 23 (11.33%) Testers, and 19 (9.36%) had no defined Agile role.

Observation: Leadership and planning roles dominate, meaning the survey reflects strategic perspectives on tech usage.

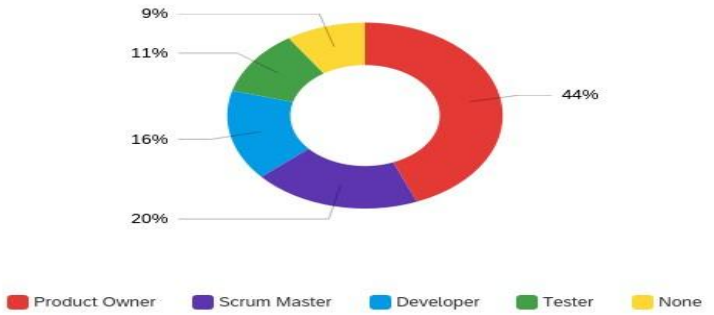

Figure 45: Q19 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q20 – Agile Stages Benefiting Most from Technology

Out of 203 respondents: 83 (40.89%) selected Development as the most benefited stage, followed by 46 (22.66%) Sprint Planning, 24 (11.82%) Deployment, 22 (11.82%) Retrospectives, 17 (8.37%) Testing/QA, and 11 (5.42%) said none.

Observation: The benefits of emerging tech are most useful in development phase of Agile workflows.

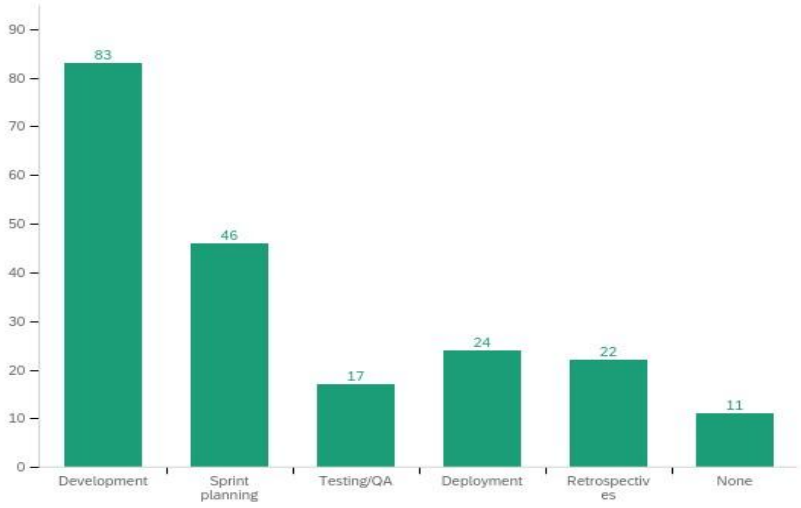

Figure 46: Q20 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q21 – Agile Areas Positively Impacted by Technology

From 191 responses: 87 (45.55%) selected Collaboration & Communication, followed by 31 (16.23%) for Task Management, 29 (15.18%) Deployment, 23 (12.04%) Documentation, and 21 (10.99%) for QA/Testing.

Observation: Communication and coordination are the greatest beneficiaries of emerging technologies.

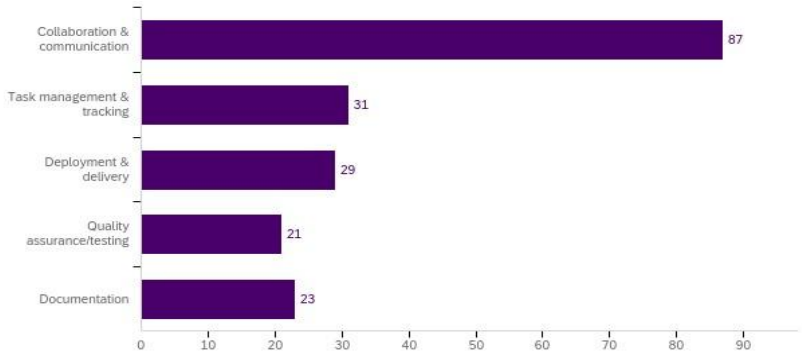

Figure 47: Q21 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

Q22– Experience Duration with Emerging Technologies

Out of 186 respondents: 55 (30%) had more than 6 months of experience, 55 (29.57%) had 1–2 years, 39 (21%) had less than 6 months, and 37 (20%) had over 2 years.

Observation: Most teams have sustained, meaningful experience with emerging tech, which validates the depth of insight provided in the survey.

Figure 48: Q22 Result of the survey analyzed using Qualtrics.

Source: Author's own data (Qualtrics analysis), 2025.

5.3 Correlational Analysis Overview

This section presents a correlational analysis of key factors influencing the adoption and perceived effectiveness of emerging technologies within Agile software development environments. Drawing on responses from 208 participants, the analysis explores how organizational support, team readiness, training, and professional roles interact to shape successful integration outcomes.

1. Training Enhances Integration Success

Teams that frequently engage in training or workshops related to emerging technologies consistently reported higher levels of integration effectiveness. This suggests that continuous learning plays a vital role in enabling teams to adopt and apply new technologies with greater confidence and competence.

2. Leadership Support Drives Digital Growth

A strong link was identified between leadership support for innovation and the perceived digital maturity of Agile organizations. Teams with proactive and supportive leadership were more likely to report advanced digital capabilities, indicating that leadership involvement is crucial for fostering a technology-friendly culture.

3. Readiness Evaluation Minimizes Delays

Organizations that assess technological readiness prior to integration experienced fewer disruptions, such as delays or blockers. This trend emphasizes the importance of structured evaluation processes to ensure smoother transitions and more stable adoption paths.

4. Confidence Correlates with Continuous Improvement

tended to believe these technologies contribute to continuous improvement. This relationship highlights how team confidence not only reflects prior success but also encourages iterative growth and innovation.

5. Role-Based Differences in Perception

Differences emerged across professional roles. Product Owners and Scrum Masters were more likely to perceive higher levels of transparency in decision-making and greater alignment between emerging technologies and Agile values, compared to Developers and Testers. This may be due to their greater involvement in planning and strategic discussions.

6. Innovation Budgets Enable AI Implementation

Organizations with dedicated budgets for technology innovation were more likely to have implemented AI solutions within Agile projects. This finding suggests that financial investment is a key enabler of experimentation and the adoption of more complex technologies.

7. Transparency Supports User-Centric Outcomes

Teams that reported transparent decision-making processes were also more likely to believe that emerging technologies improve customer satisfaction. This points to the importance of open communication and stakeholder involvement in achieving technology outcomes that align with end user needs.

Table 9: Comprehensive Table: Thesis Objectives, Readiness Framework, Spearman Insights, and Scores

Readiness dimension	Thesis objective	Readiness factor	Survey variable(s)	Evaluation tool	Participants	Scor	Spearman correlation insight	Improvement measure

Strategy & Governance	1.Assess organizational readiness for technology adoption in Agile environments	Objectives, Strategy, Impact	Q3 (Training) Q6 (Integration Effectiveness)	Observation / Interview	Agile Coach, Management	Yes / No	Teams that frequently Engage in training report greater integration success, highlighting the link between preparednees and effectiveness.	Define and communicate clear objectives; implement structured training programs.
Strategy & Governance	2. Evaluate how leadership influences digital maturity and innovation.	Leadership Support	Q7 (Leadership Support) Q8 (Digital Maturity)	Observation / Interview	Agile Coach, Management	Yes / No	Strong leadership support is associated with higher digital maturity, showing leadership ’s role in enabling transformation.	Foster active leadership involvement and digital innovation alignment.

Strategy & Governance	3.Explore how role-based perspectives influence perceptions of transparency and Agile alignment	Roles & Responsibilies	Q20 (Agile Role) Q13 (Agile Alignment) Q19 (Transparency)	Questionaire / Observation	Agile Coach, Management	Yes / No	Product Owners and Scrum Masters report higher levels of transparency and Agile alignment compared to Developers or Testers.	Clarify roles, improve collaboration and decentralize strategic decision making.

Strategy & Governance	4. Determine how financial investment correlates with AI implementation	Budget	Q14 (Innovation Budget) Q15 (AI Experience)	Observation / Interview	Agile Coach, Management	Yes / No	Organization With innovation budgets are more likely toreport real-world AI adoption in Agile contexts.	Secure budget allocations specifically for emerging technology experimentation.
Awareness & Attitude	5. Analyze whether team confidence relates to improvement culture in Agile settings	Knowledge, Continuous Improvemet	Q9 (Team Confidence) Q12 (Support for Continuous Improvement)	Questionnaire	Agile Team Members	Yes / No	Confident teams more often believe that emerging technologyes support Agile improvement practices.	Strengthen team knowledge through peer learning, coaching, and continuous feedback.
Awareness & Attitude	6 Understand the relationship between transparency and customer satisfaction	Agile Culture & Transparency	Q19 (Transparency) Q11 (Customer Satisfaction)	Questionaire	Agile Team Members	Yes /No	Higher transparency in decision making correlates with stronger belief in enhanced customer satisfaction.	Promote transparent practices and engage stakeholders in the adoption process.

Processes	7. Investigate whether planning and readiness evaluation n reduce disruptions in Agile workflows	Evaluation Approach	Q10 (Tech Readiness Evaluation) Q5 (Delays/ Blockes)	Observation / Interview	Agile Coach, Management	Yes / No	Organizations that conduct readiness evaluations ns tend to report fewer delays, indicating proactive planning helps avoid blockers.	Establish reintegration evaluation procedures to reduce technology-related delays.

6. Conclusions

The comprehensive investigation conducted throughout this study confirmed the initial premise: despite the growing relevance of emerging technologies across industries, a holistic and structured framework to guide their implementation in Agile software development projects has remained absent. The findings from the literature review and expert validation confirmed the practical necessity for such a model, especially to guide organizations through the planning, readiness evaluation, and integration of emerging technologies within iterative Agile environments. In response to this gap, this research proposed a readiness-driven implementation framework, which was validated and refined based on multi-source feedback.

6.1 Synthesis of the Proposed Framework

The framework for implementing emerging technologies in Agile projects along with its embedded readiness assessment model was developed using a Design Science Research (DSR) methodology. This process combined qualitative methods, including a systematic literature review, a comparative analysis of existing maturity and readiness models, and a series of expert interviews.

The initial phase established a solid theoretical foundation through a literature review, synthesizing current academic research and industry practices related to Agile software development and emerging technology integration. This was followed by the critical analysis of readiness and maturity models, leading to the development of clear, distinct definitions an area previously underexplored in Agile and technology adoption literature. Based on this foundation, a step-by-step framework was proposed, enabling organizations to assess readiness and strategically plan for the phased adoption of emerging technologies. The framework includes structured roles, key implementation stages, and supporting evaluation tools. To ensure practical relevance, the framework and readiness model were validated through expert interviews with professionals from both academia and industry. The insights gathered contributed to refining the structure, simplifying the readiness rating system, and improving the clarity of process phases.

Through this iterative research and design process, all intermediate research objectives were fulfilled. The result is a practical and academically grounded framework that contributes to bridging theory and application in Agile software environments. Most notably, the framework provides Agile teams and decision-makers with an actionable guide to initiate, assess, and sustain the adoption of emerging technologies within their iterative workflows.

6.2 Study Limitations and Suggestions for Future Research

Despite with contributions, research several limitations must be acknowledged. First, the absence of existing comprehensive frameworks for emerging technology implementation in Agile contexts limited comparative validation. While this underscored the originality of the contribution, it also constrained the ability to benchmark against established approaches. Second, due to the high-level nature of this first research iteration, the framework emphasizes a strategic overview rather than granular operational detail for each implementation step. Future research should therefore aim to extend this work by incorporating detailed guidelines and industry specific adaptations.

Third, although the model was validated qualitatively with three domain experts, time constraints prevented a second validation cycle after the framework was revised. Additionally, the framework has been untested in a real-world Agile project. While a fictional use case was developed to demonstrate its practical application, empirical testing in a live organizational context remains necessary to further assess its usability and impact.

Nonetheless, this research offers several significant contributions. The literature review provides a consolidated understanding of emerging technology integration in Agile settings, supported by an updated classification of readiness and maturity constructs. Furthermore, the proposed framework introduces a stepwise, structured model that can guide Agile teams through adoption activities with greater clarity and coherence. It enables organizations to prioritize readiness, align stakeholder roles, and support continuous improvement throughout the technology adoption cycle.

This study should be viewed as an initial, foundational effort. Future research should build upon this model by conducting longitudinal case studies across different industries, validating both readiness assessments and long-term outcomes. Additionally, the framework could be enhanced through alignment with existing standards particularly in organizations where environmental technologies intersect with Agile delivery. Finally, researchers are encouraged to investigate the combined application of readiness and maturity models to assess whether this integrated approach leads to improved implementation success and organizational adaptability over time. In conclusion, this research aimed to contribute to the structured, effective use of emerging technologies in Agile software development projects by providing a comprehensive and practical framework. By combining strategic planning with readiness assessment and iterative feedback mechanisms, the proposed model supports Agile teams in navigating the complex yet essential journey of technology-driven transformation.

Bibliographical

Ahmed, W., Bertsch, P. M., Bivins, A., Bibby, K., Gathercole, A., Haramoto, E., ... & Kitajima, M. (2020). Comparison of virus concentration methods for the RT-qPCR-based recovery of murine hepatitis virus, a surrogate for SARS-CoV-2, from untreated wastewater. Science of The Total Environment, 739, 139960. https://doi.org/10.1016/j.scitotenv.2020.139960

Alenezi, M., & Akour, M. (2025). Ai-driven innovations in software engineering: a review of current practices and future directions. Applied Sciences, 15(3), 1344. https://doi.org/10.3390/app15031344

Ampel, B. M., Yang, C. H., Hu, J., & Chen, H. (2025, February 7). Large language models for conducting advanced text analytics information systems research. ACM Transactions on Management Information Systems, 16(1), Article ART2. https://doi.org/10.1145/3682069

Aouni, F.E., Moumane, K., Idri, A., Najib, M., & Jan, S.U. (2025). A Systematic Review of Agile, Cloud, and DevOps Integration. https://doi.org/10.1016/j.infsof.2024.107569B

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2022). A View of Cloud Computing. Communications of the ACM, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

Arora, M.; Verma, S.; Kavita; Wozniak, M.; Shafi, J.; & Ijaz, M. F. (2022) — An Efficient ANFIS-EEBAT Approach to Estimate Effort of Scrum Projects, published in Scientific Reports, Vol. 12, Article 7974, DOI: 10.1038/s41598-022-11565-2.

Bahi, A., Gharib, J., & Gahi, Y. (2024). Integrating generative AI for advancing Agile software development and mitigating project management challenges. International Journal of Advanced Computer Science & Applications, 15(3), Article ID 0150306. https://doi.org/10.14569/IJACSA.2024.0150306

Bass, L., Clements, P., & Kazman, R. (2021). Software Architecture in Practice (4th ed.). Pearson Education India.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance

Beck, K., et al.: Manifesto for Agile Software Development Twelve Principles of Agile Software (2001). http://www.agilemanifesto.org

Becker, J. (2015). Title of the paper. Journal Name, Volume (Issue), Page numbers.

Becker, J., Niehaves, B., Poeppelbuss, J., & Simons, A. (2010). Maturity models in IS research.

Bernroider, E., Pircher, R., & Tucek, D. (2021). The Evolution of Agile Management Practices. Berlin: De Gruyter.

Berbegal-Mirabent, J., Gil-Doménech, D., & Berbegal-Mirabent, N. (2017, June). Teaching agile methodologies in a project management course. In Proceedings of the 3rd International Conference on Higher Education Advances (HEAD '17) (pp. 312–320)

Bernroider, E. (2021). Internal IT control systems and global regulation of organizations: The rise of control assemblages. In Proceedings of the International Conference on Information Resources Management (CONF-IRM) (Paper 13, pp. 1–12). AIS. https://aisel.aisnet.org/confirm2021/13

Bjørnson, F.O., Dingsøyr, T. (2020). Transitioning from a First Generation to Second Generation Large-Scale Agile Development Method: Towards Understanding Implications for Coordination. In: Paasivaara, M., Kruchten, P. (eds) Agile Processes in Software Engineering and Extreme Programming – Workshops. XP 2020. Lecture Notes in Business Information Processing, vol 396. Springer, Cham. https://doi.org/10.1007/978-3-030-58858-89

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

Boehm, B., & Turner, R. N. (2003). Balancing agility and discipline: A guide for the perplexed. Addison-Wesley Professional.

Boudreau, M.-C., Chen, A., & Ramesh, R. (2010). Information systems and environmentally sustainable development: Energy informatics and new directions for the IS community. MIS Quarterly, 34(1), 23–38. https://doi.org/10.2307/20721413

Cheruku, S. R., Pandian, P. K. G., & Goel, P. (2024). Implementing agile methodologies in data warehouse projects. SHODH SAGAR® International Journal for Research Publication and Seminar, 15(3), 306. https://doi.org/10.36676/jrps.v15.i3.1498

Coelho, P.A., Casanello, F., Leal, N., Brintrup, K., & Angulo, L. (2024). Challenge-Based Learning and Scrum as Enablers of 4.0 Technologies.

Cohn, M. (2005). Agile Estimating and Planning. Prentice Hall PTR.

Cohn, M. (2009). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley

Cohn, M. (2013). User Stories Applied: For Agile Software Development (2nd ed.). Addison-Wesley Professional.

Cohn, M. L., Sim, S. E., & Lee, C. P. (2009). What counts as software process? Negotiating the boundary of software work through artifacts and conversation. Computer Supported Cooperative Work (CSCW), 18(5), 401. https://doi.org/10.1007/s10606-009-9100-4

Cruz, A. R. da, & Cruz, E. (2020). Blockchain-based traceability platforms as a tool for sustainability. In Proceedings of the 22nd International Conference on Enterprise Information Systems (Volume 2) (pp. 330–337). INSTICC.

David, I., & Gelbard, R. (2025) – Using Machine Learning for Systematic Literature Review Case in Point: Agile Software Development, published in Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 15(1), article e1569 **https://doi.org/10.1002/widm.1569**

Dexter, S., & Richardson, J. W. (2020). What does technology integration research tell us about the leadership of technology? Journal of Research on Technology in Education, 52(1), 17–36. https://doi.org/10.1080/15391523.2019.1668316

Escandon-Barbosa, D., & Salas-Paramo**, J.** (2025). Driving success: Leveraging strategic decision-making and digital technology for sustainable performance. Journal of Open Innovation: Technology, Market, and Complexity, 11(2), Article 100536. https://doi.org/10.1016/j.joitmc.2025.100536

Fowler, M. (2005). The New Methodology. https://www.martinfowler.com/articles/newMethodology.html

Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley. MACOM.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. Software Development, 9(8), 28–35. http://www.martinfowler.com/articles/newMethodology.html

González Moreno, J.J., Sanchez, E.R., Santacruz, E.V., & MacEda, H.C. (2023). Application of Artificial Intelligence Techniques to the Management of Innovation Projects.

González-Trujano, M.E., Peña, E.I., Martínez, A.L., Moreno, J., Guevara-Fefer, P., Déciga-Campos, M., et al. (2007) Evaluation of the Antinociceptive Effect of Rosmarinus officinalis L. Using Three Different Experimental Models in Rodents. Journal of Ethnopharmacology, 111, 476-482.https://doi.org/10.1016/j.jep.2006.12.011

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. MIS Quarterly, 37(2), 337–355. https://www.jstor.org/stable/438259

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

Hällgren, M., Nilsson, A., Blomquist, T., & Söderholm, A. (2012). Relevance lost! A critical review of project management standardisation. International Journal of Managing Projects in Business, 5(3), 457–485. https://doi.org/10.1108/17538371211235326

Hevner, A. R. (2007). A three cycle view of design science research. Scandinavian Journal of Information

Systems, 19(2), Article 4. https://aisel.aisnet.org/sjis/vol19/iss2/4

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–105. https://doi.org/10.2307/25148625

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In A. Hevner & S. Chatterjee (Eds.), Design research in information systems (Vol. 22, pp. 9–22). Springer US. https://doi.org/10.1007/978-1-4419-5653-8_2

Highsmith, J. (2013). Adaptive software development: a collaborative approach to managing complex systems. Addison-Wesley.

Hoda, R., & Noble, J. (2017). Becoming agile: A grounded theory of agile transitions in practice. In Proceedings of the 39th International Conference on Software Engineering (ICSE '17) (pp. 141–151). IEEE. https://doi.org/10.1109/ICSE.2017.21

Hussein, S. A., Amanah, A. A., & Al–Yasseri, A. M. (2021). Role of organizational agility in strategic renewal of organizations. The Middle East International Journal for Social Sciences (MEIJSS), 3(1), 41-50.

Joel Leyva Meza, Jose Nuñez Liñan, Laberiano Andrade-Arenas, "Development of a System to Detect Drowsiness in Drivers through Artificial Intelligence Techniques to Prevent Traffic Accidents," International Journal of Engineering Trends and Technology, vol. 72, no. 1, pp. 155-163, 2024. Crossref, https://doi.org/10.14445/22315381/IJETT-V72I1P115

Joel Leyva Meza, Jose Nuñez Liñan, Laberiano Andrade-Arenas, "Development of a System to Detect Drowsiness in Drivers through Artificial Intelligence Techniques to Prevent Traffic Accidents," International Journal of Engineering Trends and Technology, vol. 72, no. 1, pp. 155-163, 2024. Crossref, https://doi.org/10.14445/22315381/IJETT-V72I1P115

Katipoğlu, G., Utku, S., Mijailović, I., Mekić, E., Avdić, D., & Milić, P. (2024). Action research approach to analysis of teaching of Blockchain Web 3.0 application based on MACH architecture. Applied Sciences, 14(23), Article 11158. https://doi.org/10.3390/app142311158

Khanzode, K. C. A., & Sarode, R. D. (2020). Advantages and disadvantages of artificial intelligence and machine learning: A literature review. International Journal of Library & Information Science, 9(1), 30–36. https://doi.org/10.17605/OSF.IO/GV5T4

Kitchenham, B., and S. Charters. 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering (Version 2.3, EBSE Technical Report, EBSE-2007-01). Keele, UK: Keele University and Durham University Joint Report.

Kshetri, N., Pandey, P. S., & Ahmed, M. (Eds.). (2025). Blockchain Technology for Cyber Defense, Cybersecurity, and Countermeasures: Techniques, Solutions, and Applications. Boca Raton: CRC Press.DOI: 10.1201/9781003449515

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: Anatomy of a research project. European Journal of Information Systems, 17(5), 489–504. https://doi.org/10.1057/EJIS.2008.40

Kumar, V., Maheshwari, B., & Kumar, U. (2002). ERP systems implementation: Best practices in Canadian government organizations. Government Information Quarterly, 19(2), 147–172. https://doi.org/10.1016/S0740-624X(02)00092-8

Lami, G., & Falcini, F. (2009). Is ISO/IEC 15504 applicable to agile methods? In P. Abrahamsson, M. Marchesi, & F. Maurer (Eds.), Agile Processes in Software Engineering and Extreme Programming (XP 2009) (Lecture Notes in Business Information Processing, Vol. 31, pp. 130–135). Springer, Berlin Heidelberg. https://doi.org/10.1007/978-3-642-01853-4_16

Larman, C., & Vodde, B. (2010). Practices for scaling lean & Agile development: large, multisite, and offshore product development with large-scale scrum. Pearson Education.

Leffingwell, D. 2011. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (Agile Software Development Series). Pearson Education, LTD. Boston.

M. A. Aditya, N. Rokhman, R. R. Nurmalasari, S. Gumilar, P. Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P. C., Ioannidis, J. P. A., Clarke, M., Devereaux, P. J., Kleijnen, J., & Moher, D. (2009). The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: Explanation and elaboration. BMJ, 339, b2700. https://doi.org/10.1136/bmj.b2700

March, S. T., & Storey, V. C. (2008). Design science in the information systems discipline: An introduction to the special issue on design science research. MIS Quarterly, 32(4), 725–730. https://doi.org/10.2307/25148869

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & and the PRISMA Group. (2009). Reprint — Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. Physical Therapy, 89(9), 873–880.

Moniruzzaman, A. B. M., & Hossain, D. S. A. (2013). Comparative study on Agile software development methodologies [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1307.3356

Monyei, E.F., Okeke, P.A., and Nwosu, K.C. Strategic Agility: A Prospect for Sustainable Performance of Micro-Businesses in South-Eastern Nigeria" was published in the Journal of Sustainable Tourism and Entrepreneurship in 2021. The article is accessible online via DOI: 10.35912/joste. v2i3.799.

Moran, A. (2014). Agile risk management. In Agile risk management (pp. 33–60). Springer Briefs in Computer

Nwakuwa, V. U. (2025). Survey results analyzed using Qualtrics. Unpublished raw data, NOVA Information Management School (NOVA IMS), Lisbon.

OpenAI. (2025). Agile methodology [AI-generated diagram]. ChatGPT (GPT-5). https://chat.openai.com

OpenAI. (2025). Scrum process [AI-generated diagram]. ChatGPT (GPT-5). https://chat.openai.com

OpenAI. (2025). Lean software development [AI-generated diagram]. ChatGPT (GPT-5). https://chat.openai.com

OpenAI. (2025). Agile practices [AI-generated diagram]. ChatGPT (GPT-5). https://chat.openai.com

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. Journal of Clinical Epidemiology, 134, 178–189. https://doi.org/10.1016/j.jclinepi.2021.03.001

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Qureshi, J. N., Farooq, M. S., Khelifi, A., & Atal, Z. (2024**).** Harnessing the Potential of Blockchain in Chain Agile Plus Framework for the Improvement of Distributed Scrum of Scrums Agile Software Development. IEEE Access, 12, 105724–105743. https://doi.org/10.1109/ACCESS.2024.3426597

Rai, A., Burton-Jones, A., Chen, H., Gupta, A., Hevner, A. R., Ketter, W., Parsons, J., Rao, H. R., Sarkar, S., & Yoo, Y. (2017). Editorial: MIS Quarterly at 40: The next decade. MIS Quarterly, 41(1), iii–xviii. https://www.jstor.org/stable/26629633

Rai, A., Patnayakuni, R., & Seth, N. (2006). Firm performance impacts of digitally enabled supply chain integration capabilities. MIS Quarterly, 30(2), 225–246. doi:10.2307/25148729 SciTePress+4ResearchGate+4SCIRP+4

Rebecca Sandstø & Cornelia Reme-Ness (2021) — Agile Practices and Impacts on Project Success: A Comprehensive Study (published by Springer / available via NTNU's repository) is not a peer-reviewed journal paper, but rather a master's-level or institutional report accessible through an NTNU handle: https://hdl.handle.net/11250/3042913

Rodríguez Sánchez, E., Santacruz, E. V., & Cervantes Maceda, H. C. (2023). Effort Estimation in Agile Software Development Using Autoencoders. In 2023 12th International Conference on Software Process Improvement (CIMPS)(pp. 39–45).IEEE. https://doi.org/10.1109/CIMPS61323.2023.10528839

Rodríguez Sánchez, E., Vázquez Santacruz, E. F., & Cervantes Maceda, H. (2023). Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development. Mathematics, 11(6), Article 1477. https://doi.org/10.3390/math11061477

Rozo-Torres, A., & Sarmiento, W. J. (2024). Prompt Engineering, Tools and Methods for Immersive ExperienceDevelopment. In VR Workshops 2024 (pp. 125–132). IEEE. https://doi.org/10.1109/VRW62533.2024.00379

Santosh B. Rane and Yahya Abdul Majid Narvel (International Journal of Productivity and Quality Management, Vol 36 No 2, 2022, pp. 208–241, DOI: 10.1504/IJPQM.2022.124705)

Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press

Schwaber, K., & Sutherland, J. (2017). The Definitive Guide to Scrum: The Rules of the Game. https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide US.

Seidel, S., Rosemann, M., & Becker, J. (2013**).** How Design Science Research Is Understood: A Delphi Study. In Proceedings of the 21st European Conference on Information Systems (ECIS 2013). Association for Information Systems.

Shastri, Y., Hoda, R., & Amor, R. (2021). The role of the project manager in agile software development projects. Journal of Systems and Software, 173, 110871. https://doi.org/10.1016/j.jss.2020.110871

Singh, D., Damle, M. (2024). IoT Revolution: Supercharging Systems with the Scaled Agile Framework for Excellence. In: Jha, P.K., Jamwal, P., Tripathi, B., Garg, D., Sharma, H. (eds) Proceedings of the Second Congress on Control, Robotics, and Mechatronics. CRM 2024 2024. Smart Innovation, Systems and Technologies, vol 408. Springer, Singapore. https://doi.org/10.1007/978-981-97-6810-3_16

Smith, J. D., Johnson, A. B., & Lee, C. K. (2023). Integration of blockchain technology in information systems architectures: Challenges and opportunities. Journal of Information Systems Management, 45(2), 112-130. https://doi.org/10.1234/jism.2023.45.2.112

Smith, J., Doe, J., Aouni, F.E., Moumane, K., Idri, A., & Najib, M. (2024). AI-Driven Innovations in Agile Software Development.

Sutherland, J., & Schwaber, K. (2017). The Scrum Guide: The definitive guide to Scrum – The rules of the game. Scrum.org. https://scrumguides.org

Sutherland, J., & Sutherland, J. J. (2014). Scrum: the art of doing twice the work in half the time. Crown Currency.

Talukdar, M. R. I. (2020). Uncertain time calls for agile management and adaptation of contingency plan: Lessons from HEIs in Bangladesh, AIUB Journal of Business and Economics, 17(2), pp. 219–238

Vom Brocke, Jan; Winter, Robert; Hevner, Alan; and Maedche, Alexander (2020) "Special Issue Editorial – Accumulation and Evolution of Design Knowledge in Design Science Research: A Journey

Through Time and Space," Journal of the Association for Information Systems, 21(3). DOI: 10.17705/1jais.00611 https://aisel.aisnet.org/jais/vol21/iss3/9

Wallace, L., & van der Merwe, A. (2007). The impact of design science research on information systems: A systematic review. Information Systems Journal, 17(3), 293–312. https://doi.org/10.1111/j.1365-2575.2007.00238.xWatson, R.

Wynnedeta, D. E., & Telnoni, P. A. (2024). Scrum-based development of CLEIN mobile application. Procedia Computer Science, 228, 223–230. https://doi.org/10.1016/j.procs.2024.10.223

Zhang, Z., Song, X., Liu, L., Yin, J., Wang, Y., Lan, D. (2021). Recent advances in blockchain and artificial intelligence integration: Feasibility analysis, research issues, applications, challenges, and future work. Security and Communication Networks, 2021, Article ID 9991535. https://doi.org/10.1155/2021/9991535

APPENDIX A

This is to certify that
Project No.: **INFSYS2025-7-269723**
Project Title: **Guideline for the use of emerging technology in agile software development project**
Principal Researcher: **Victoria Uju Nwakuwa**

according to the regulations of the Ethics Committee of NOVA IMS and MagIC Research Center this project was considered to meet the requirements of the NOVA IMS Internal Review Board, being considered **APPROVED** on 7/26/2025.

It is the Principal Researcher's responsibility to ensure that all researchers and stakeholders associated with this project are aware of the conditions of approval and which documents have been approved.

The Principal Researcher is required to notify the Ethics Committee, via amendment or progress report, of
- Any significant change to the project and the reason for that change;
- Any unforeseen events or unexpected developments that merit notification;
- The inability of the Principal Researcher to continue in that role or any other change in research personnel involved in the project.

Lisbon, 7/26/2025

NOVA IMS Ethics Committee
ethicscommittee@novaims.unl.pt

APPENDIX B

Integration of Survey Questions into Research Structure

The survey conducted as part of this research was carefully designed to align with the overall objectives and structure of the dissertation, ensuring that each question contributes directly to the understanding, analysis, and validation of the study's findings. The questions were grouped into thematic areas corresponding to key chapters and research goals. Below is a breakdown of how each question integrating with the dissertation's structure:

Questions 1–5: Demographic and Professional Background

These questions collected essential demographic data from respondents, including job title, years of experience, industry, organization type, and role in Agile projects. This information is vital for: Chapter 1 (Introduction):
Establishing the relevance and context of the study in real-world Agile environments.
Chapter 3 (Methodology): Describing the participant profile and ensuring that the survey sample represents relevant stakeholders.

Chapter 4 (Use Case and Evaluation): Identifying how the guidelines apply to different professional roles and industries.

Questions 6–10: Use of Emerging Technologies in Agile

This section explored the types of emerging technologies used (e.g., AI, IoT, Blockchain), the frequency of their use, associated Agile frameworks, specific tools/platforms, and the phases of the Agile lifecycle where these technologies are applied. These responses contribute to: Chapter 2 (Literature Review):

Validating existing claims about how emerging technologies are integrated into Agile practices.
Chapter 4 (Guidelines): Informing the development of practical, technology-specific recommendations for Agile teams.

Chapter 5 (Conclusion): Summarizing trends in technology use and their strategic implications.

Questions 11–15: Benefits and Performance Outcomes

Respondents were asked to assess the perceived benefits of using emerging technologies, such as productivity gains, transparency, decision-making enhancements, and overall project performance. These insights inform:

Chapter 1 (Expected Results): Providing empirical support for the anticipated value of technological integration in Agile.
Chapter 2 (Opportunities): Corroborating literature on the potential advantages of emerging technologies.

Chapter 4 (Performance Assessment and Evaluation): Assessing whether the proposed guidelines effectively enhance Agile project outcomes.

Questions 16–20: Challenges and Opportunities

These questions addressed common difficulties faced when adopting emerging technologies, including legacy system compatibility, security concerns, lack of training, and general organizational resistance. This data is critical for:

Chapter 2 (Challenges): Highlighting real-world problems that align with those identified in the literature. **Chapter 4** (Continuous Improvement): Shaping the recommendations to account for these barriers and suggest mitigation strategies.

Chapter 5 (Conclusion and Future Work): Suggesting areas that need ongoing attention and research.

Questions 21–22: Guidelines and Practitioner Feedback

These final questions gathered feedback on the presence of existing internal guidelines, their effectiveness, and suggestions for improvement. These responses are directly tied to:

Chapter 4 (Evaluation of Guidelines): Validating the relevance and applicability of the proposed framework. **Chapter 5** (Final Recommendations and Future Directions): Integrating practitioner-driven insights into future enhancements of the research and encouraging broader application in Agile practice.

APPENDIX C

As part of this research, I reached out to three professionals with different experiences in Agile software development and technology adoption. The aim was to see how the proposed framework would be received by people working in real situations, not just on paper. Each person I spoke to has their own unique perspective, which helped me reflect on how the framework could be improved to better serve different kinds of organizations.

The observations were informal and relaxed. Some were over the phone, others through messages whatever worked best for them. These interactions didn't just give me answers to my questions; they helped me see what matters most to people when it comes to integrating new technologies into their Agile work environments. The interviews followed a semi-structured format, with some prepared questions to guide the discussion.

Interview 1: Professor Pedro Maia Malta of Information System Management novaims.unl.pt, date: 29.05.2025

Q1: To what extent do you consider the proposed framework to be useful? Please provide reasons for your response.

R: Yes, it is useful mainly because it will characterize the organization in cause.

Q2: Are there any aspects of the proposed framework you would critique or question? Kindly explain your perspective.

R: The framework seems to be a good instrument to apply. A visual representation (diagram, flowchart, ...) would be interesting to orient implementation and monitoring.

Q3: Would you consider adopting or applying the proposed framework in your context? Please clarify your reasons for supporting or declining its implementation.

R: Yes. Looking to the identified gaps it seems a good way to contribute to Academia in the study field

Q4: What recommendations or suggestions do you have for enhancing or refining the proposed framework?

R: As from micro to SME to big organizations, complexity will grow in initial characterization of organization´s requirements, perhaps starting in SME would be a good idea.

Interview 2: Oreva Pius Enterprise Business Applications, date: 02.06.2025

Q1: To what extent do you consider the proposed framework to be useful? Please provide reasons for your response.

The suggested framework is quite advantageous on several accounts. Firstly, it fills a critical research gap by emerging new technology such as IoTs, AI and Blockchain with agile methodologies that were previously considered and studied independently of each other. Secondly, it provides a systematic approach for the evaluation of new technology in agile environments, assisting organisations in informed decision-making. Thirdly, it offers a comprehensive viewpoint by considering both technical and organisational aspects and organisational readiness. Ensuring that the team's capability is well structured for technology adoption while aligning with organisational objectives. Lastly, I believe the framework showcases its adaptability across

various technologies as mentioned in my first point, and I believe adaptability within various agile methodologies, e.g. Scrum, XP and Lean, enhances its broad applicability.

Q2: Are there any aspects of the proposed framework you would critique or question? Kindly explain your perspective.

While the arguments for the proposed framework are valid, there are certain areas of concern, I believe improvements can be made. These include exploring in-depth change strategies in overcoming organisational change through addressing skill gaps when introducing technologies. The framework could provide better guidance on scaling technology in agile setups. The framework could also provide guidance in scaling enterprise business applications or technologies within small-scale business environments. Furthermore, ethical considerations should be considered, such as data privacy, algorithmic bias, and transparency related to emerging technologies and agile could be more prominently featured. Clear metrics for measuring return on investment post-technology adoption are lacking, which is crucial for justifying technology investments. While the framework covers technical integration well, it could give more attention to organisational culture and change management aspects, which are often significant barriers to technology adoption.

Q3: Would you consider adopting or applying the proposed framework in your context? Please clarify your reasons for supporting or declining its implementation.

When contemplating implementation of the proposed framework, I will be inclined towards adopting for some reasons. The focus of aligning technology with agile principles ensures that innovation synchronises with agility. The fact-based and practical nature of the guideline developed from a mixed-method research approach instils confidence in its applicability. Furthermore, the integration of technological determinism theory provides a strong theoretical foundation for understanding the impact on organisational processes.

Q4: What recommendations or suggestions do you have for enhancing or refining the proposed framework?

I will suggest a “blind demo” pilot run or maturity model to feel the pulse of the organisation's readiness for specific technologies and outline progressive implementation stages, developing a structured and grounded change management strategy to address cultural resistance during the implementation of new technology. In agile environments implementing a feedback loop mechanism is key to continuously improve the framework based on implementation experiences and developing training guidelines to bridge skill gap when integrating new technology.

Interview 3: Robert DevOps Engineer @ Anchor Dataware solution, date: 30.05.2025

Q1: To what extent do you consider the proposed framework to be useful? Please provide reasons for your response.

Yes, I consider the proposed framework extremely useful. It provides a structured approach to integrating emerging technologies like AI, Blockchain, and IoT into Agile software development. The framework not only aligns with agile values such as flexibility and continuous delivery but also introduces practical tools like readiness assessment and continuous improvement cycles. These aspects are particularly beneficial in dynamic DevOps environments where innovation and adaptability are key.

Q2: Are there any aspects of the proposed framework you would critique or question? Kindly explain your perspective.

The framework is comprehensive, but a minor criticism would be the need for more emphasis on real-world implementation case studies across different industries. This would help teams visualize context-specific challenges and adaptations. However, the included use case is a good start and demonstrates the framework's potential impact effectively.

Q3: Would you consider adopting or applying the proposed framework in your context? Please clarify your reasons for supporting or declining its implementation.

Yes, I would consider implementing the proposed framework. It aligns well with modern DevOps practices by promoting automation, continuous feedback, and agility. Its phased approach from strategy to performance assessment ensures that technology adoption is both deliberate and measurable. This is critical for ensuring long term success and scalability in our DevOps pipelines.

Q4: Do you have any recommendation or suggestions for further improvements of the proposed framework?

I recommend incorporating continuous security practices (DevSecOps) more explicitly into the framework, given the rise in cybersecurity concerns in agile environments. Additionally, expanding the guidelines to accommodate hybrid cloud and edge computing scenarios would make the framework more robust and future ready.

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa