

Factors influencing the effective integration of secure development practices in agile teams

Kaylieh Manikam

**A research report submitted to the Faculty of Commerce, Law and
Management, University of the Witwatersrand, in partial fulfilment of the
requirements for the degree of Master of Management in the field of Digital
Business**

Wits Business School

Johannesburg, 2025

ABSTRACT

The integration of secure development practices (SDPs) within agile teams is critical to mitigating security risks in the software development lifecycle (SDLC), particularly in industries such as South African financial services, where cybersecurity threats are prevalent and ever-evolving. This qualitative study explored the factors influencing the effective integration of SDPs, focusing on organisational structure and culture, leadership commitment and support, and tools and technology.

Using the Technology-Organisation-Environment (TOE) framework and validated by the OWASP Software Assurance Maturity Model (SAMM), this study established a conceptual framework that comprehensively analysed and assessed the integration of SDPs.

Semi-structured interviews were conducted with a diverse range of IT professionals working in agile teams. These individuals, including software engineers, quality assurance (QA) engineers, architects, product owners, project managers, and IT managers, play a role in delivering working software. The research identified key themes through thematic data analysis: the need for a security-conscious organisational culture, proactive leadership support and commitment, and effective deployment of security tools integrated within CI/CD pipelines.

The TOE framework provided a holistic lens for understanding how technological, organisational, and environmental factors collectively influence the effective integration of SDPs. At the same time, OWASP SAMM offered a structured approach to measure and validate the maturity of these practices across the *Governance, Design, Implementation, Verification, and Operations* pillars.

While organisations emphasised cybersecurity, inconsistencies in training, communication gaps, and prioritisation of feature delivery over security often hinder SDP integration. Proactive leadership that allocates resources and fosters collaboration and continuous learning positively impacts outcomes. Similarly, when effectively integrated, modern security tools enhance agility and security but face challenges such as legacy system compatibility and the risks of overreliance.

This study contributes to the theoretical and practical body of knowledge on secure software development by using the TOE framework and aligning practical recommendations with OWASP SAMM.

KEYWORDS

Secure Software Development, Secure Development Practices, Security, DevOps, Agile, Scrum, SDLC, Cybersecurity, OWASP, SAMM

DECLARATION

I, Kaylieh Manikam, declare that this research report is my own work except as indicated in the references and acknowledgements. It is submitted in partial fulfilment of the requirements for the degree of Master of Management in the field of Digital Business at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in this or any other university.

Name: Kaylieh Manikam

Signature: 

Signed at ...Arnhem.....

On the ...11th..... day of ...February..... 2025..

DEDICATION

This research report is dedicated to agile teams, software practitioners, and leaders who continuously innovate, collaborate, and drive the pursuit of excellence in software delivery.

ACKNOWLEDGEMENTS

I want to extend my sincere gratitude to all those who have contributed to the completion of this research report.

To my husband, Deshan, thank you for your steadfast support and encouragement, which have been instrumental in helping me navigate the challenges of this endeavour. Your understanding and patience have been invaluable in enabling me to focus on my studies. I am grateful to my family for their unwavering support and encouragement throughout this journey.

I am greatly appreciative of my supervisor, Dr Pius Oba, whose guidance, expertise, and constructive feedback have been essential to the successful completion of this research report. Your insights and mentorship have greatly enriched this work, and I am grateful for your support throughout the research process.

To the participants of this study, I extend my sincere thanks for your time, insights, and contributions. Your willingness to share your experiences provided the foundation for this research and significantly enriched its outcomes.

Finally, I thank my employer for providing the resources and flexibility necessary to pursue this academic undertaking. Your support has been critical in facilitating the successful completion of this research.

TABLE OF CONTENTS

ABSTRACT	II
KEYWORDS	III
DECLARATION	IV
DEDICATION	V
ACKNOWLEDGEMENTS	VI
LIST OF TABLES	X
LIST OF FIGURES.....	X
LIST OF ACRONYMS.....	XI
CHAPTER 1. INTRODUCTION	1
1.1 STATEMENT OF PURPOSE	1
1.2 BACKGROUND OF STUDY	1
1.3 RESEARCH PROBLEM	2
1.4 RESEARCH QUESTIONS.....	4
1.5 RATIONALE	4
1.6 DELIMITATIONS OF THE STUDY	5
1.7 DEFINITION OF TERMS	6
1.8 ASSUMPTIONS.....	7
1.9 CHAPTER OUTLINE	8
CHAPTER 2. LITERATURE REVIEW AND THEORETICAL FRAMEWORK	9
2.1 INTRODUCTION	9
2.2 BACKGROUND DISCUSSION	9
2.2.1 AGILE METHODOLOGIES AND DEVOPS	9
2.2.2 SECURE DEVELOPMENT PRACTICES (SDPS)	10
2.3 ORGANISATIONAL STRUCTURE AND CULTURE.....	11
2.3.1 PROPOSITION 1.....	13
2.4 LEADERSHIP SUPPORT AND COMMITMENT	13
2.4.1 PROPOSITION 2.....	15
2.5 SECURITY TOOLS AND TECHNOLOGY.....	15
2.5.1 PROPOSITION 3.....	16
2.6 ANALYTIC FRAMEWORK.....	17

2.6.1	THEORETICAL FRAMEWORK	17
2.6.2	SOFTWARE SECURITY MATURITY MODELS.....	20
2.6.3	CONCEPTUAL FRAMEWORK	22
2.7	CONCLUSION OF LITERATURE REVIEW.....	24
CHAPTER 3. RESEARCH METHODOLOGY		25
3.1	RESEARCH PARADIGM	25
3.2	RESEARCH APPROACH	26
3.3	RESEARCH DESIGN	26
3.4	DATA COLLECTION METHODS	27
3.5	POPULATION AND SAMPLE	27
3.5.1	POPULATION.....	27
3.5.2	SAMPLING AND SAMPLING METHOD	28
3.6	THE RESEARCH INSTRUMENT	28
3.7	PROCEDURE FOR DATA COLLECTION.....	29
3.8	DATA ANALYSIS STRATEGIES AND INTERPRETATION.....	29
3.9	POSSIBLE LIMITATIONS AND CHALLENGES	32
3.10	QUALITY ASSURANCE	32
3.10.1	TRANSFERABILITY	32
3.10.2	CREDIBILITY	32
3.10.3	CONFIRMABILITY	32
3.10.4	DEPENDABILITY	32
3.11	ETHICAL CONSIDERATIONS	33
CHAPTER 4. PRESENTATION OF FINDINGS		33
4.1	INTRODUCTION	33
4.2	THEMATIC ANALYSIS	34
4.3	FINDINGS OF PROPOSITION 1	39
4.3.1	ORGANISATIONAL CULTURE	40
4.3.2	ORGANISATIONAL STRUCTURE.....	44
4.3.3	ENVIRONMENTAL FACTORS.....	49
4.4	FINDINGS OF PROPOSITION 2	51
4.4.1	LEADERSHIP COMMITMENT	52
4.4.2	LEADERSHIP SUPPORT	55
4.5	FINDINGS OF PROPOSITION 3	57
4.5.1	AUTOMATION AND INTEGRATION	58
4.5.2	TOOLS AND TECHNOLOGY.....	61

4.6 SUMMARY OF FINDINGS	64
CHAPTER 5. DISCUSSION OF FINDINGS.....	65
5.1 INTRODUCTION	65
5.2 DISCUSSION OF PROPOSITION 1	65
5.2.1 ORGANISATIONAL CULTURE	65
5.2.2 ORGANISATIONAL STRUCTURE.....	67
5.2.3 ENVIRONMENTAL FACTORS.....	70
5.3 DISCUSSION OF PROPOSITION 2	72
5.3.1 LEADERSHIP COMMITMENT	72
5.3.2 LEADERSHIP SUPPORT	74
5.4 DISCUSSION OF PROPOSITION 3	76
5.4.1 AUTOMATION AND INTEGRATION	76
5.4.2 TOOLS AND TECHNOLOGY	78
5.5 CONCLUSION	81
CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS.....	82
6.1 INTRODUCTION	82
6.2 CONCLUSION REGARDING RESEARCH QUESTION 1.....	82
6.3 CONCLUSION REGARDING RESEARCH QUESTION 2.....	84
6.4 CONCLUSION REGARDING RESEARCH QUESTION 3.....	85
6.5 RECOMMENDATIONS	86
6.6 CONTRIBUTION OF THE STUDY.....	88
6.7 SUGGESTIONS FOR FURTHER RESEARCH	89
REFERENCES.....	91
APPENDIX A: PARTICIPANT INFORMATION FORM	101
APPENDIX B: PARTICIPANT CONSENT FORM.....	102
APPENDIX C: CONSISTENCY TABLE.....	103
APPENDIX D: RESEARCH INSTRUMENT.....	105
APPENDIX E: ETHICS CLEARANCE CERTIFICATE	107

LIST OF TABLES

Table 1. Sample Population.....	28
Table 2. Participant Information.....	34
Table 3. Thematic Analysis Results	35
Table 4. Consistency Table - RQs, Propositions, Data Collection and Analysis	103
Table 5. Interview Questions	105

LIST OF FIGURES

Figure 1. Technology-Organisation-Environment Framework Diagram. Adapted from Tornatzky & Fleischer, 1990	18
Figure 2. Software Assurance Maturity Model (SAMM) [Source: (OWASP, 2024)] ..	22
Figure 3. Conceptual Framework	24
Figure 4. Data collection and analysis overview.....	30
Figure 5. Thematic Analysis Procedure	30
Figure 6. Word Cloud related to findings for RQ 1.....	40
Figure 7. Theme, category and code hierarchy - Organisational Culture	41
Figure 8. Theme, category and code hierarchy - Organisational Structure	45
Figure 9. Theme, category and code hierarchy - Environmental Factors	49
Figure 10. Word Cloud related to findings for RQ 2.....	52
Figure 11. Theme, category and code hierarchy - Leadership Commitment	53
Figure 12. Theme, category and code hierarchy - Leadership Support.....	55
Figure 13. Word Cloud related to findings for RQ 3.....	58
Figure 14. Theme, category and code hierarchy - Automation and Integration	59
Figure 15. Theme, category and code hierarchy - Tools and Technology.....	61

LIST OF ACRONYMS

4IR: Fourth Industrial Revolution

AI: Artificial Intelligence

BSIMM: Building Security in Maturity Model

CI/CD: Continuous Integration/Continuous Delivery

CMMI: Capability Maturity Model Integration

DAST: Dynamic Application Security Testing

DDoS: Dedicated Denial of Service

DOI: Diffusion of Innovation

FICA: Financial Intelligence Centre Act

FSP: Financial Services Provider

IoT: Internet of Things

IT: Information Technology

OWASP: Open Web Application Security Project

PO: Product Owner

POPIA: Protection of Personal Information Act

PM: Project Manager

QA: Quality Assurance

SAMM: Software Assurance Maturity Model

SAST: Static Application Security Testing

SDP: Secure Development Practice

SDLC: Software Development Lifecycle

TAM: Technology Adoption Model

TOE: Technology-Organisation-Environment

CHAPTER 1. INTRODUCTION

1.1 Statement of Purpose

The purpose of this qualitative study was to explore the factors influencing the effective integration of secure development practices (SDPs) in agile software development teams.

1.2 Background of Study

The fourth industrial revolution (4IR) has accelerated digitalisation, necessitating businesses to be agile and adapt swiftly to change. This is particularly evident in South Africa's financial services industry, which has undergone significant transformations driven by technological advancements. The financial industry faces unique challenges and opportunities as it moves to digital innovations such as cloud technology, mobile banking, fintech solutions, and blockchain. These innovations have the potential to enhance customer experiences and streamline operations, but they also expose South African Financial Services Providers (FSPs) to cybersecurity threats (FSCA, 2022). Regarding cloud technology, it was reported that South African organisations are the most concerned about cloud-related threats (PWC, 2024).

This underscores the urgent need for FSPs to bolster the confidentiality, integrity, and accessibility of their digital assets (hardware and software) amid an ever-evolving threat landscape. Cyberattacks exploiting software vulnerabilities can result in data breaches, financial losses, reputational damage, and operational disruptions. In 2023, IBM reported that the global average cost of a data breach was \$4.45 million, which was a 15% increase over the last three years, indicating the financial impact organisations could potentially face if a breach occurs (IBM, 2023).

The ENISA Annual Report (2023) highlighted ransomware, Dedicated Denial of Service (DDoS) attacks, threats against data, and malware as the top prime threats. Malware and phishing attacks were also reported as a top threat in the South African public sector in 2024 (Dawood et al., 2024). As software becomes more ubiquitous and technology advances, the threat landscape will evolve and become even more sophisticated, emphasising the need for robust security measures through the

Software Development Lifecycle (SDLC). According to a 2024 report by Veracode, 80% of all software applications scanned using Veracode tools contained one or more security vulnerabilities in active applications (Veracode, 2024).

Agile methodologies emphasise iterative development, rapid feedback, and customer collaboration. Teams are enabled to deliver software more quickly and respond to changing requirements. DevOps, an extension of agile, bridges the gap between development and operations teams by fostering collaboration, automation, and continuous integration/continuous delivery (CI/CD) through various integrated tools. DevOps enables agile teams to continuously develop, test, deploy and monitor their software applications (Atlassian, 2024). Agile methodologies and DevOps practices within the SDLC have driven a transformative shift in the software development landscape.

Recent studies have highlighted the need to integrate SDPs into agile and DevOps workflows since the rapid pace of development can inadvertently create security vulnerabilities in software applications. Research has explored various approaches, such as DevSecOps and integrating security tools into CI/CD pipelines, although this does not always lead to efficiency and rapid delivery.

The emphasis on rapid delivery conflicts with the thoroughness and rigour required for security, leading to a trade-off between agility and security. Additionally, organisational factors such as team structures, culture, leadership, and the availability of tools and technologies play a crucial role in determining the successful integration of SDPs.

As the South African financial services industry navigates the complexities of digital transformation, the imperative to incorporate security throughout the SDLC becomes increasingly urgent. Addressing this challenge can enhance the security posture of the developed software applications and mitigate potential risks and their implications for the broader economy.

1.3 Research Problem

McGraw (2004) defines software security as the process of engineering software so that it will function correctly even under a malicious attack. The need for security measures throughout the SDLC against potential threats and vulnerabilities has never been more critical. By increasing the security posture of the software applications

being developed, the risk of an attack can be mitigated, and the possible effects of an attack can be minimised.

Agile methodologies focus on the rapid development and deployment of functional requirements. However, security is commonly considered a non-functional requirement, a property of quality, and is usually performed in the final steps of the development process (Tøndel et al., 2022). This is often too late to catch high-ranking threats and can lead to delivery delays. (Kreitz, 2019). According to McGraw (2004), the most effective approach to building secure software involves considering security early and continuously in the SDLC. Implementing security early in the SDLC can improve the software application's security posture.

Agile teams face unique challenges in integrating SDPs into their current processes. These challenges, such as compromised agility and flexibility to deliver working software, are well-documented (Mihelič et al., 2023). Security considerations include gathering security requirements, threat modelling, secure coding practices, and design patterns. They must be thorough and well-documented, which requires more time and specialised knowledge (Khan et al., 2022). Other studies have indicated that the technical complexity of security, organisational culture, and leadership can hinder the integration of SDPs (Umeugo, 2023), further highlighting the complex nature of this problem.

Existing software security frameworks, such as OWASP Software Assurance Maturity Model (SAMM) (OWASP, 2024), Building Security in Maturity Model (BSIMM) (Synopsys, 2024), McGraw's Secure SDLC (McGraw, 2004), and the Capability Maturity Model Integration (CMMI) (ISACA, 2024), offer guidance on integrating SDPs into the SDLC. However, their generic nature often poses challenges for organisations seeking to adapt these frameworks within their own contexts, constraints, and risk profiles (Khan et al., 2022). The complexity of implementing and adapting these frameworks in the face of evolving threats and technologies can hinder their effectiveness. The lack of clear metrics and evaluation mechanisms makes it difficult to measure the impact of these frameworks on an organisation's overall security posture. In some cases, the use of frameworks can lead to a box-ticking exercise for the sake of compliance (Ryan et al., 2023).

This study addressed these challenges by exploring the cultural, leadership and technological factors that influence the effective integration of SDPs within agile teams. Agile teams often face challenges identifying and integrating security tools that align with their fast-paced, iterative workflows. The impact of tool choices on development efficiency and security posture must be thoroughly examined before implementation.

By understanding how these factors influence each other, and the integration of SDPs, organisations and agile teams can adopt a practical and tailored approach. Validating the data collected against real-life maturity models also provided an objective ranking of the effectiveness of the explored factors by measuring the security posture.

1.4 Research Questions

This study answers the following research questions:

- 1. RQ1. How do organisational structure and culture influence the effective integration of SDPs in agile teams?**
- 2. RQ2. How does leadership support and commitment influence the effective integration of SDPs in agile teams?**
- 3. RQ3. How does the use of tools and technology influence the effective integration of SDPs in agile teams?**

1.5 Rationale

The rapidly evolving threat landscape has made the development of secure software imperative for an organisation's performance and reputation. The literature review indicates that recent studies have been systematic literature reviews and quantitative studies involving a wide range of IT professionals. Few qualitative studies have explored the organisational, leadership, and technological factors that influence and successfully enable the integration of SDPs in agile teams in a South African context.

Despite the recognised importance of security in software development, comprehensive strategies and practices that effectively integrate security into the agile

SDLC are lacking. Studies have primarily focused on technical aspects and the adaptation of development frameworks to integrate security, leaving a gap in understanding the role of organisational structures and culture, leadership and the effectiveness of tools and technologies in effectively integrating SDPs.

This research presents a valuable contribution to the existing body of knowledge, offering practical insights into integrating SDPs in agile teams. Although security culture on an organisational level has been studied, specific factors affecting agile teams have not been adequately addressed.

By understanding these factors, organisations can develop strategies to integrate SDPs into their processes effectively, ensuring secure software delivery without compromising speed or agility. This research has significant potential benefits, as it can pave the way for more secure and efficient software development practices, thereby mitigating the risks posed by the ever-evolving threat landscape.

In South Africa, the financial services sector employs strict regulatory requirements, such as those enforced by the Financial Sector Conduct Authority (FSCA), the Protection of Personal Information Act (POPIA), and various industry guidelines, which emphasise the need for robust security controls. However, many financial institutions are rapidly adopting agile methodologies to respond quickly to market demands. Failure to effectively integrate SDPs into agile processes can leave critical financial systems, personal data, and customer trust at risk.

This study's focus on the organisational, leadership, and technological factors that influence successful SDP integration in agile teams addresses a vital need within South Africa's financial services industry. Its findings can guide organisations in implementing a stronger security-conscious culture, leadership involvement, and tool adoption to protect the interests of consumers, meet regulatory requirements, and preserve the industry's reputation. By identifying best practices that balance compliance, innovation, and speed-to-market, this research can significantly strengthen the security posture in South Africa's financial services industry.

1.6 Delimitations of the Study

This research focused on agile teams within the South African financial services industry, specifically targeting banks and insurance companies. The study examined

the organisational, leadership and technology factors influencing the effective integration of SDPs. However, it did not extend to broader operational or software development teams outside the South African financial services industry.

While the study acknowledged regulatory frameworks such as POPIA, it did not assess how effectively these regulations enforce security practices within organisations. Additionally, the research did not explore the technical aspects of DevOps implementation or cybersecurity frameworks such as NIST or ISO 27001, as the focus was on organisational and leadership influences rather than compliance-driven security frameworks.

The study also excluded a technical evaluation of software tools, security frameworks, and technical implementations. While the role of security tools in agile environments was discussed, the research did not assess specific security tools, CI/CD pipelines, SAST or DAST analysis tools, or vulnerability management systems. Instead, it focuses on how the use of tools influence agility, the decision-making processes, implementation barriers, and enablers that impact security tool integration within agile teams.

1.7 Definition of Terms

Agile Methodologies: A set of iterative software development approaches (e.g., Scrum, Kanban) prioritising collaboration, customer feedback, and adapting quickly to change throughout the development process.

CI/CD (Continuous Integration/Continuous Delivery): Automated processes that enable code integration, testing, and deployment, ensuring more frequent and reliable software releases.

Cyberattacks: Malicious attempts to access, disrupt, or damage digital systems, networks, or data. These include hacking, phishing, ransomware, malware, and DDoS attacks.

DAST (Dynamic Application Security Testing): Process of analysing the performance of web applications through simulated attacks. Often regarded as “black box” testing, as the tool has no access to the source code of the application being tested.

DevOps: A combination of cultural philosophies, practices, and tools that integrate software development (Dev) and IT operations (Ops) to streamline the delivery of high-quality software.

DevSecOps: An extension of DevOps that integrates security practices within the CI/CD pipelines.

FSP: Financial Services Provider. An organisation such as a bank or insurance company.

OWASP (Open Web Application Security Project): A non-profit foundation that focuses on improving the security of web applications. OWASP provides resources, tools, and guidelines to help developers and organisations identify and mitigate security vulnerabilities.

SAST (Static Application Security Testing): Analysing the source code of an application to detect software vulnerabilities and security flaws. Regarded as “white box” testing.

SAMM (Software Assurance Maturity Model): A model for organisations to analyse and improve their software security posture. This was developed by the OWASP organisation

SDPs (Secure Development Practices): Best practices for integrating security into the software applications that are built. These include threat modelling, secure coding practices and code analysis.

SDLC (Software Development Lifecycle): A structured process that defines the phases (planning, analysis, design, implementation, testing, deployment, maintenance) involved in creating and maintaining software applications.

Security Posture: The overall security status of an organisation’s applications, networks, and procedures, as well as its ability to prevent, detect, and respond to cyberattacks.

1.8 Assumptions

The following assumptions are made for the study:

- i. Participants are currently working in agile teams using Scrum or Kanban and have a knowledge of DevOps.
- ii. Participants have a sufficient understanding of cybersecurity threats and what SDPs entail.
- iii. Participants are open and honest in expressing their lived experiences.

1.9 Chapter Outline

Chapter 1 provides an overview of the research purpose, background, research problem and the rationale for the study. The purpose of the qualitative study is to examine the factors affecting the integration of SDPs in organisations that use agile methodologies or DevOps for software development. The background discusses the impact of the fourth industrial revolution on digitalisation and the increased cybersecurity threats due to technological advancements. The research problem emphasises the escalating cybersecurity threats and the need for robust security measures throughout the SDLC.

Chapter 2 provides a comprehensive review of the existing literature and theoretical frameworks related to the integration of SDPs in agile software development. By exploring the theoretical foundations and core concepts, this chapter establishes a basis for a deeper understanding of the factors influencing the successful integration of SDPs within agile development processes.

Chapter 3 describes the research methodology used and explains the qualitative research approach. It describes the data collection methods, which include interviews, and the process of engaging participants. The data analysis process, ethical considerations, and steps taken to ensure ethical research conduct are also discussed.

Chapter 4 describes the findings from qualitative research conducted with participants in the South African financial services industry. The results from the thematic analysis are presented, highlighting key themes such as organisational factors, leadership support, and technological tools that influence the integration of SDPs.

Chapter 5 discusses the meaning and implications of the findings from Chapter 4, referring to the OWASP SAMM model and comparing them to previous studies to answer each research question.

Chapter 6 provides the conclusions for each research question. It also provides pragmatic recommendations that organisations can apply in their own settings, the study's contribution, and suggestions for future research.

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL FRAMEWORK

2.1 Introduction

This chapter reviews the existing literature and research related to integrating SDPs in agile teams. The literature review provides a comprehensive understanding of agile methodologies, DevOps and SDPs. The organisational structure and cultural influences on security, leadership influence, and the use of security tools and technologies in CI/CD pipelines are provided. By synthesising and analysing previous studies, this literature review contributes to the identification of gaps in current knowledge and provide a foundation for the subsequent research in this qualitative study. A conceptual framework that guided the research and interpretation of the results is formulated.

2.2 Background Discussion

2.2.1 Agile Methodologies and DevOps

Over two decades ago, the codification of the Agile Manifesto (Cunningham, 2001) drastically changed the SDLC. Since its conception, there have been numerous studies and extensions of its fundamental principles into several frameworks (Dingsøyr et al., 2012).

A publication by Misra et al. (2012) discusses the core principles of agile methodologies. These principles aim to increase software development teams' efficiency, collaboration, flexibility, and agility. The emphasis on customer collaboration ensures that the product delivered aligns closely with the user's expectations. Agile prioritises responding quickly to change over following a fixed plan, allowing agile teams to adapt to changing requirements and customer needs effectively. This

adaptability is facilitated by iterative development cycles, dividing work into small, manageable increments, enabling continuous feedback and improvement.

Another core principle is focusing on individuals and interactions over processes and tools. This fosters a collaborative environment where motivated team members can leverage their collective expertise to solve problems more effectively. Agile also promotes the frequent delivery of working software, with a preference for shorter timescales, which helps maintain a steady pace of progress and early detection of issues.

The DevOps philosophy, which has adapted many of its principles from the Agile Manifesto, is being adopted in many organisations. DevOps is a set of practices, tools, and a cultural movement to foster collaboration. This minimises barriers between development and operations teams. Like agile, it emphasises communication and collaboration to build, test, deploy and monitor software applications continuously (Atlassian, 2024). Critical practices include CI/CD pipelines, which are key to maintaining a continuous software product lifecycle.

Many tools designed to automate and streamline processes support this cultural shift, making it easier for teams to collaborate and deliver software more efficiently. DevOps represents a significant shift in how software development and operations are approached, aiming to deliver business value more rapidly and reliably. Many organisations use DevOps and agile practices to implement an end-to-end software delivery solution.

2.2.2 Secure Development Practices (SDPs)

Software security is described as the process of engineering software so that it will function correctly even under a malicious attack (McGraw, 2004). The application of SDPs, which include threat modelling, secure coding, static and dynamic code analysis, and regular security testing, is a crucial aspect of ensuring the resilience of the software, even in the event of a malicious cyberattack.

SDPs are a set of activities that can be integrated throughout the SDLC to ensure the confidentiality, integrity and availability of software systems and data. A comprehensive survey by Otieno et al. (2023) suggests a holistic and proactive approach to ensuring application security. These practices encompass security by

design, threat modelling and incorporating security requirements during the initial stages of development. Secure coding practices, as indicated in OWASP guidelines (OWASP, 2010), are performed during development by software engineers to prevent common vulnerabilities. Static and dynamic code analysis, penetration testing, and vulnerability scanning can also be included in the development and testing phase to identify potential weaknesses. Additionally, code reviews and security audits ensure adherence to SDPs (SAFECode, 2018).

Research in this area has proven that fostering a security-conscious culture through training and awareness programs for developers and stakeholders is essential to build trust and enhance the security posture of the software (Fauzi et al., 2023). A comprehensive approach to secure software development is encouraged while addressing challenges in agile methodologies and promoting the integration of SDPs into development workflows.

Many organisations use a combination of agile methodologies and DevOps workflows to automate and rapidly deliver working software to their customers. Security considerations are often implemented after the development phase, which is too late to catch vulnerabilities. SDPs should be implemented into development workflows early and continuously.

2.3 Organisational Structure and Culture

Organisational structure and culture are pivotal in influencing SDPs by fostering an environment that prioritises security from the design phase. The organisational structure refers to how security and development teams are structured, the roles and responsibilities, and how communication and collaboration occur. A strong security culture, where employees uphold values for protecting information and comply with security policies, can significantly reduce incidents and breaches (Nurgalieva et al., 2023).

Research like those conducted by Schubert et al. (2023), Nurgalieva et al. (2023), and a longitudinal case study by Lende et al. (2023) indicate the role of organisational structures and defined roles influence the effective integration of SDPs. They identify the security champion as a solution to bridge the communication and knowledge gap between security and development teams. These individuals, with their specialised

knowledge and collaborative skills, foster security awareness and facilitate the integration of SDPs. There was a positive shift in the software engineer's security awareness when a security champion was a part of the development team. However, a case study by Gutfleisch et al. (2023) cautions that the success of this role is contingent upon organisational support, clear mandates, and a well-defined security strategy.

Several studies, including the quantitative research of Ryan et al. (2023) and the mixed-methods approach of Tolah et al. (2021), highlighted the importance of a security-conscious culture. Well-communicated security policies, adequate resource allocation, and leadership support fostered an environment where security is valued and prioritised. A positive security culture encouraged the successful integration of SDPs and empowered individuals to take ownership of security in their daily tasks. A case study performed by Alshaikh & Adamson (2021), showed that awareness is not enough and influencing employee behaviour has a greater impact on adhering to the organisation's security policies.

A qualitative study by Tøndel et al. (2022) further underscored the importance of developer motivation and the detrimental impact of organisational silos on the effective integration of SDPs. The collaboration between development, operations, and security teams, as emphasised in DevOps practices, is vital for integrating security into the SDLC. This collaboration involves technical integration and necessitates behavioural changes to create a security culture.

However, the results of the study from Ryan et al. (2023) show that effective integration of SDPs and a security-conscious culture does not always correlate. Introducing policies and processes is not enough, as this can lead to box-ticking for the sake of compliance. The environment in which the organisation operates influences the security and data privacy policies it must comply with.

Gutfleisch et al. (2021) and Ardo et al. (2022) highlight the practical challenges of SDP implementation. They reveal the complexities of navigating industry regulations and compliance frameworks, all of which can hinder the integration of usable security requirements and necessitate a flexible, adaptable approach.

The diverse body of literature on SDPs, encompassing qualitative, quantitative, mixed-methods, and case studies, paints a complex and nuanced picture. While technical

solutions for SDPs are essential, a holistic approach that considers the cultural, organisational, and individual dimensions is equally vital. Organisations can create an environment where secure software development thrives by fostering a security-conscious culture, empowering security champions, and addressing contextual challenges.

2.3.1 Proposition 1

Organisational structure and culture are vital to creating a security-conscious mindset within agile teams. Within the organisational structure, the security champion role can effectively bridge the gap between security and software development teams. Cultural factors such as awareness, training, and cross-team collaboration can create a security culture, leading to the effective integration of SDPs.

2.4 Leadership Support and Commitment

A systematic literature review by Uchendu et al. (2021) and qualitative research by Arizon-Peretz et al. (2022) , found that a commonly mentioned factor in cultivating a security-conscious culture was top management and leadership support. A positive organisational security culture was found to be strongly linked to proactive security behaviour among software engineers, therefore, more effective integration of SDPs into their processes. By prioritising security and demonstrating its importance through their actions and decisions, leaders influence the attitudes and behaviours of employees towards security. Therefore, it becomes imperative that leaders have an adequate understanding of security and its importance (Schneider et al., 2020).

A qualitative study by Arizon-Peretz et al. (2021) determined that when the priority of SDPs is not well communicated and made a priority by management, software engineers receive mixed messages and do not perceive that there is an issue with the security posture of the applications that they build.

Leadership should show a commitment to security by providing adequate resources such as budget, time, training, and the appointment of security experts within the team to emphasise its importance (Ardo et al., 2022; Gutfleisch et al., 2021). Similarly, Herath et al. (2020) highlighted the pivotal role of leadership support in allocating resources and prioritising security initiatives and hence influencing the adoption of information security solutions. Valdés-Rodríguez et al. (2024) concurred, emphasising

the need for top management support to implement appropriate security controls. With leadership's support in security initiatives, employees are more likely to perform their daily tasks with security in mind.

Leadership should effectively communicate the importance of security and their expectations regarding SDPs to their employees. This helped align the entire organisation towards a common security goal and ensures that all team members understand their roles in maintaining security. Management attention and commitment are two success factors for increasing the security posture of applications from a strategy perspective (Schubert et al., 2023). Management is required to prioritise security as part of their strategic initiatives. When leadership showed a commitment to supporting the integration of SDPs, it led to more sustainable initiatives (Tolah et al., 2021).

Effective leadership can mediate communications between various interest groups within the organisation to resolve conflicts related to security practices. This helps ensure that security measures are not compromised due to internal disagreements. The prioritisation of security activities and requirements can be driven by strong leadership (Tøndel et al., 2022). Well-communicated and accessible security policies showed a positive influence on the employee successfully adapting these practices into their daily activities.

Leaders can encourage innovation in SDPs by supporting research and development initiatives aimed at finding new and effective ways to integrate security into the development process. This proactive approach can lead to the discovery of more efficient and effective security solutions. Appiah et al. (Appiah et al., 2022) suggests that leadership strategies should include taking a flexible and innovative approach to proactively increasing the resilience and security posture of the organisation. Processes and practices should be evaluated and updated regularly in response to feedback from the procedures in place.

Leaders should initiate and support continuous training and awareness programs to educate employees about security threats and best practices. This ongoing education is essential for keeping the workforce informed and vigilant about evolving security issues (Rindell et al., 2021). By establishing clear accountability and responsibility for security within the organisation, leaders can ensure that all team members are aware

of their specific roles in maintaining security. Respondents who had a higher job satisfaction, were more motivated and showed an increased tendency to integrate SDPs into their daily activities (Tolah et al., 2021).

2.4.1 Proposition 2

Leadership commitment and support are driving factors in fostering a security-conscious environment, where agile teams are empowered and possess the necessary resources, skills, and knowledge to integrate security into their processes and, therefore increase the software's security posture.

2.5 Security Tools and Technology

The literature highlighted the significant role of tools and technology in the successful integration of SDPs. A qualitative study by Schubert et al. (2023) performed in Germany, found that integrating security tools into CI/CD pipelines increased team efficiency and the security posture of the cloud application developed. Umeugo (2023) employed a quantitative approach and identified relative advantage, compatibility, and trialability as key factors influencing the adoption of SDPs, particularly in small and medium-sized enterprises (SMEs). Their findings suggested that technical teams should prioritise tools that offer clear benefits, integrate seamlessly with existing processes, and are easy to evaluate their effectiveness in increasing the security posture.

Through a narrative review, Nurgalieva et al. (2023) cautioned that conflicts can arise between security requirements and existing processes or technologies, potentially impacting application performance and usability. Other challenges experienced with integrating security tools included educating development teams on making security decisions, efficiency within the CI/CD pipeline to avoid delays and continuously improving security hardening (Moyón et al., 2024).

Otieno et al. (2023) advocated for the early integration of automated static and dynamic analysis tools to proactively identify and mitigate risks, enabling continuous monitoring and improvement. Jenkins (2022) further explained the importance of integrating technology into the SDLC for proactive vulnerability management and detection, highlighting the complementary roles of software developer knowledge and analysis tools in enhancing the security posture. These results demonstrated the

effects of software developer awareness, knowledge and skills on the effective use of tools and technology. Gasiba et al. (2021) performed a large-scale software developer survey and the results revealed that a lack of awareness regarding SDPs and secure coding tools, which underscored the need for hands-on training and awareness programs.

Using grounded theory, Weir et al. (2021) observed that vulnerability scans and penetration testing are effective assurance techniques to ensure security. Rindell et al. (2021) identified automated testing tools and security testing methods as the most effective security practices, particularly when applied before deployment. This aligns with the findings of Valdés-Rodríguez et al. (2024), who emphasised the fundamental role of automated security testing tools like SAST and DAST in identifying vulnerabilities in advance. Additionally, an earlier study by Valdés-Rodríguez et al. (2023) highlighted the importance of integrating SDPs into agile methodologies without compromising agility, emphasising the need for research focused on inexperienced teams, using DevOps metrics for decision-making. Shostack (2022), through qualitative analysis, noted the importance of tool effectiveness, ease of integration, and lightweight methodologies for practical use.

In the context of cloud applications, Bhuiyan et al. (2019) demonstrated the importance of considering cloud security concerns, organisational readiness, and regulatory support when adopting cloud-based security tools, demonstrating the need for compatibility with existing infrastructure. Through a case study, Angermeir et al. (2021) noted a lack of automated security practices within CI/CD pipelines despite the benefits.

This diverse body of studies using different research methods such as qualitative studies, quantitative surveys, narrative reviews, and case studies, demonstrates that tools and technology are essential enablers of SDPs. However, their successful integration requires careful consideration of organisational context, user needs, and potential conflicts with existing processes.

2.5.1 Proposition 3

Tools and technology can help agile teams successfully integrate SDPs into their existing processes by implementing security practices within their CI/CD pipelines.

Factors such as organisational readiness, compatibility, budget, and technical skills play a role in the selection and adoption of tools.

2.6 Analytic Framework

2.6.1 Theoretical Framework

In qualitative studies, a conceptual or theoretical framework provides a lens to shape the investigation, and the questions asked to participants (Creswell & Creswell, 2018). There are several theories used in the field of software engineering that can be used to explain acceptance or adoption behaviour of technology. The purpose of using acceptance behavioural theories in software engineering includes guiding the design of applications and explains the acceptance and adoption level of technologies. Taherdoost (2018) and Börstler et al. (2024) argue that the Technology Acceptance Model (TAM) and Roger's Diffusion of Innovation (DOI) theory are the most used theoretical frameworks in the field of Information Technology (IT). The Technology-Organisation-Environment (TOE) framework has also been employed as a theoretical framework to analyse cyber security decisions, privacy issues and the adoption of 4IR (Herath et al., 2020; Salleh & Janczewski, 2016; Wallace et al., 2020).

Roger's DOI aims to explain the factors that impact the adoption of technology innovation, focusing on system characteristics and organisational and environmental influences (Rogers, 1983). The five constructs of DOI are relative advantage, compatibility, complexity, trialability, and observability.

TAM was first developed by Davis (1985) and posits that an individual's intentions to adopt technology are determined by two factors, i.e. perceived usefulness and perceived ease of use. It further posits the effect of external variables on these two constructs. There have been two more extensions of TAM with the formulation of TAM2 and TAM3 (Venkatesh & Bala, 2008). These have been successfully applied to explain the adoption of technology. However, a pitfall of this theory is that it has not been applied on an organisational level and focuses on future behaviour, not actual behaviour.

The Technology-Organisation-Environment (TOE) framework is an organisation-level theory first formulated by Tornatzky and Fleisher (1990). The framework posits three factors, namely technological context, organisational context, and environmental

context, that influence the adoption of technology innovation within organisations. These factors influence each other and thereby influence the decision of the organisation to adopt a technological innovation, as depicted in Figure 1.

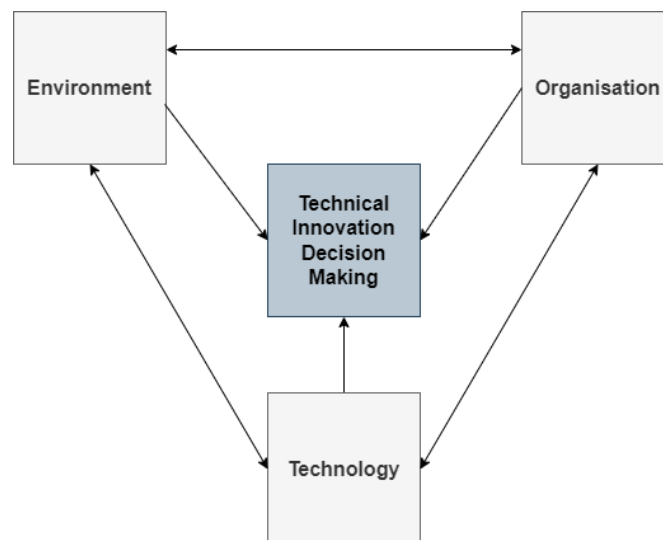


Figure 1. Technology-Organisation-Environment Framework Diagram. Adapted from Tornatzky & Fleischer, 1990

TAM, TOE, and DOI were considered to answer this study's research questions. There were some overlapping factors between TAM and DOI such as perceived ease of use, complexity, perceived usefulness and advantage. For the purposes of this research study the TOE framework was selected.

The TOE framework provided a detailed understanding of the organisational factors, including leadership, that influence the integration of SDPs and the use of security technologies, ensuring that all critical elements were considered in the analysis and the interplay between the factors of technology, organisation and the environment.

Within the TOE framework, the technological context covers the constructs such as technological readiness, complexity or ease of use, compatibility, and relative advantage. Each of these highlighted a different aspect of how well a new practice or tool fits within existing development pipelines. Regarding the integration of SDPs, these constructs highlighted how capable the team is of integrating security tools, how simple or difficult those tools are to adopt, and whether they seamlessly align with their current processes. For example, by investigating compatibility and ease of use for

automated security scanning tools, it can determine how easy they are to integrate. This technology perspective directly addresses RQ3 by explaining why some teams adopt certain security tools quickly, while others struggle due to complexity or poor compatibility.

In the organisational context, constructs include leadership support, organisational culture, structure, and resource availability. These concepts help explain how internal dynamics affect the integration of SDPs. For example, leadership support can be displayed through resource allocation and support for security initiatives. Organisational structure relates to security roles and how teams within the organisation are structured and how they communicate. This directly influences communication and cross-team collaboration, with the presence of security champions in agile teams. Culture pertains to the shared norms and values that guide how teams prioritise security relative to other objectives, such as speed or new features. Resources indicate how roles and budgets are arranged to support security activities. These constructs align with RQ1 and RQ2, highlighting the conditions under which organisational culture, leadership commitment, and formalised structures either enable or hinder the effective integration of SDPs.

The environmental context considers regulatory pressures, competitive environment, and external support. These factors come from outside the organisation yet influence the integration of SDPs. Regulations, such as data protection laws, may require teams to embed security controls in every step of the SDLC, while competitive pressure can push organisations to develop software at a rapid pace. Although the environmental context is not explicitly included in the three research questions as technology or organisational aspects, it underpins them by determining the urgency and strategic importance of integrating SDPs. For instance, organisations in heavily regulated sectors may have stronger cultural and leadership incentives to champion security, influencing how they prioritise SDPs and which tools they adopt.

Through these three interrelated contexts, the TOE framework provides a structured way to explore various influences on security integration in agile teams. It clarifies the technological enablers and barriers (RQ3), highlights how leadership, structure, and culture shape organisational readiness (RQ1 and RQ2), and acknowledges how external pressures can influence the organisation.

2.6.2 Software Security Maturity Models

There are several security maturity models and frameworks that organisations can use to measure their security posture and maturity. Some frameworks offer descriptive and prescriptive measures and can be applied to information security. Although some industry-specific models and framework exists, most are generic in nature, and numerous studies have investigated synthesising the recommendations into custom frameworks for application (Alenezi et al., 2022; Khan et al., 2021; Moyon et al., 2020).

As a tool to measure the effectiveness of the factors being investigated in this study, one of these models will be selected. A comparison of some of the common models used in software engineering, include OWASP SAMM, BSIMM, and CMMI. These frameworks are considered for use as a measurement model for the achieved software security posture.

Building Security in Maturity Model (BSIMM) is an assessment model that can be used to measure the current security posture of an organisation. This model is adaptable to agile and DevOps teams; however, it doesn't offer prescriptive security practices. An advantage of BSIMM is the availability of benchmarking data that organisations can use to compare the security posture against others in the same industry (Synopsys, 2024).

The Capability Maturity Model Integration (CMMI) is a tool that allows organisations to understand their current level of capability within the context of their own business objectives. This model is quite generic and can be adapted to assess any business objective to drive adoption of different processes to make improvements (ISACA, 2024).

The OWASP Software Assurance Maturity Model (SAMM) is a model that can be used by organisations to measure and improve their security posture. SAMM can be applied to the full SDLC and is process and technology-agnostic, meaning it can be applied to agile and DevOps teams. An advantage of SAMM is that the model is both prescriptive and descriptive. The OWASP SAMM v2 framework consists of five business functions, each containing three security practices. The security practices have two streams with a set of activities and three maturity levels. The model is depicted in Figure 2.

The business functions include *Governance*, which ensures organisational oversight of the overall software related activities through strategy initiatives, creation and measurement of metrics, and ensuring employees possess the required knowledge and skills to design, build and maintain secure software. *Design*, which integrates security requirements into architecture and design through threat assessments and ensuring secure architecture from the start of the SDLC. *Implementation*, which focuses on secure coding, deployment and its related defects during the development phase of the SDLC. *Verification*, which ensures software security through tests and quality assurance. Lastly, *Operations*, which supports post-deployment security to ensure that operational remains resilient in the to the evolving threat landscape.

The three maturity levels are basic implementation (Level 1), intermediate processes with consistency (Level 2), and automated integration with continuous improvement (Level 3). These maturity levels provide a roadmap for continuously improving secure software development practices. This structure ensures that organisations can assess, plan, and improve their security posture in alignment with their specific goals and constraints (OWASP, 2024).

For these reasons, OWASP SAMM v2 was chosen as the model to assess the security posture under technological, organisational, and environmental influences. The use of SAMM was also employed by Gutfleisch et al. (2023) in their study on assessing the security practices in agile teams and the effectiveness of the security champion within these teams.

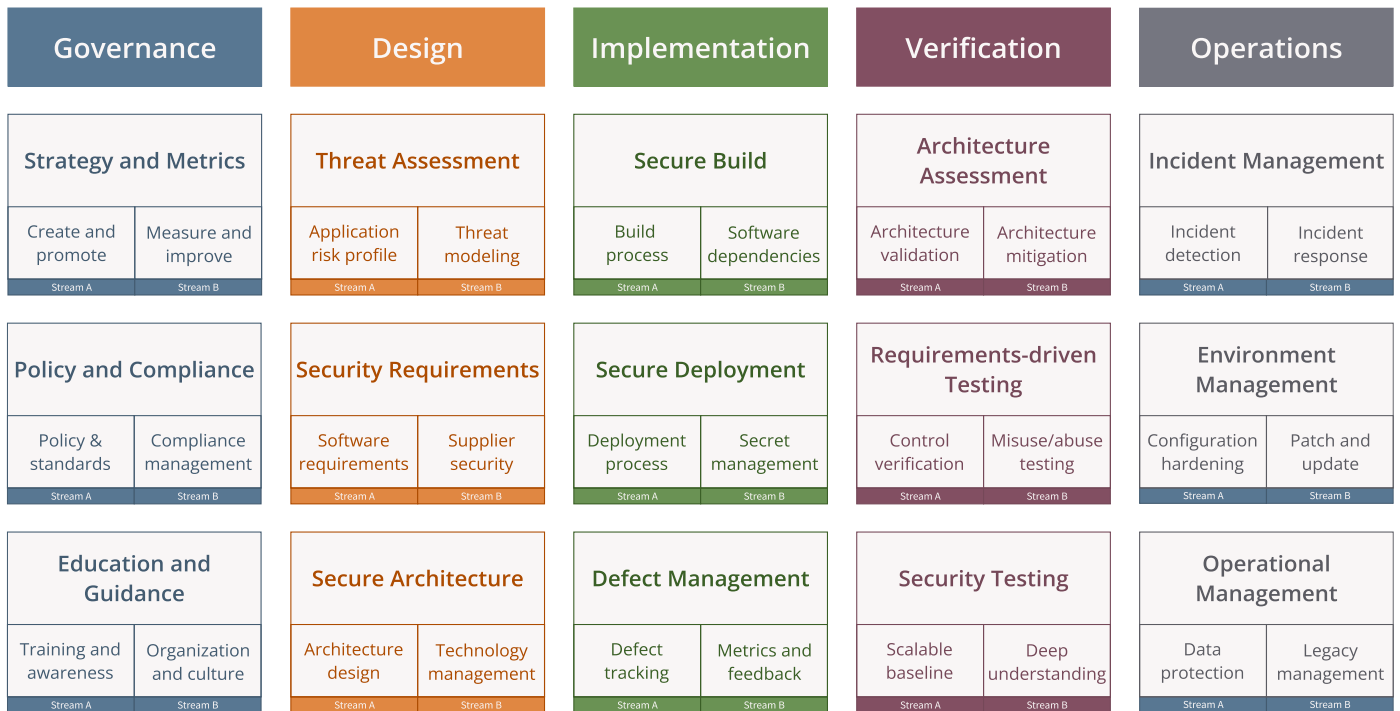


Figure 2. Software Assurance Maturity Model (SAMM) [Source: (OWASP, 2024)]

2.6.3 Conceptual Framework

To answer the research questions, a conceptual framework based on the TOE framework and OWASP SAMM v2 for validation was formulated as shown in Figure 3. The constructs of the traditional TOE framework were expanded to include factors that directly influence the integration of SDPs (Wallace et al., 2020). The conceptual framework posits that the technological, organisational and environmental facets influence the effective integration of SDPs in agile teams.

The Organisation factor underscores the role of team structures, dedicated security roles, collaboration, security awareness, training and policies in facilitating or hindering the integration of SDPs. The Organisation factor is separated into two blocks to answer RQ1 and RQ2 on organisation culture and leadership separately.

The Technology factor emphasises the importance of awareness, knowledge, complexity, effectiveness, compatibility, and perceived usefulness in shaping the adoption of security tools and technologies pertaining to RQ3.

The Environment factor recognises the impact of an evolving threat landscape and industry regulations on the integration of SDPs and how they influence both the Organisation and Technology factors.

SAMM was then employed to assess and validate the organisation's maturity in integrating SDPs across various stages of the SDLC, i.e. *Governance, Design, Implementation, Verification, and Operations*.

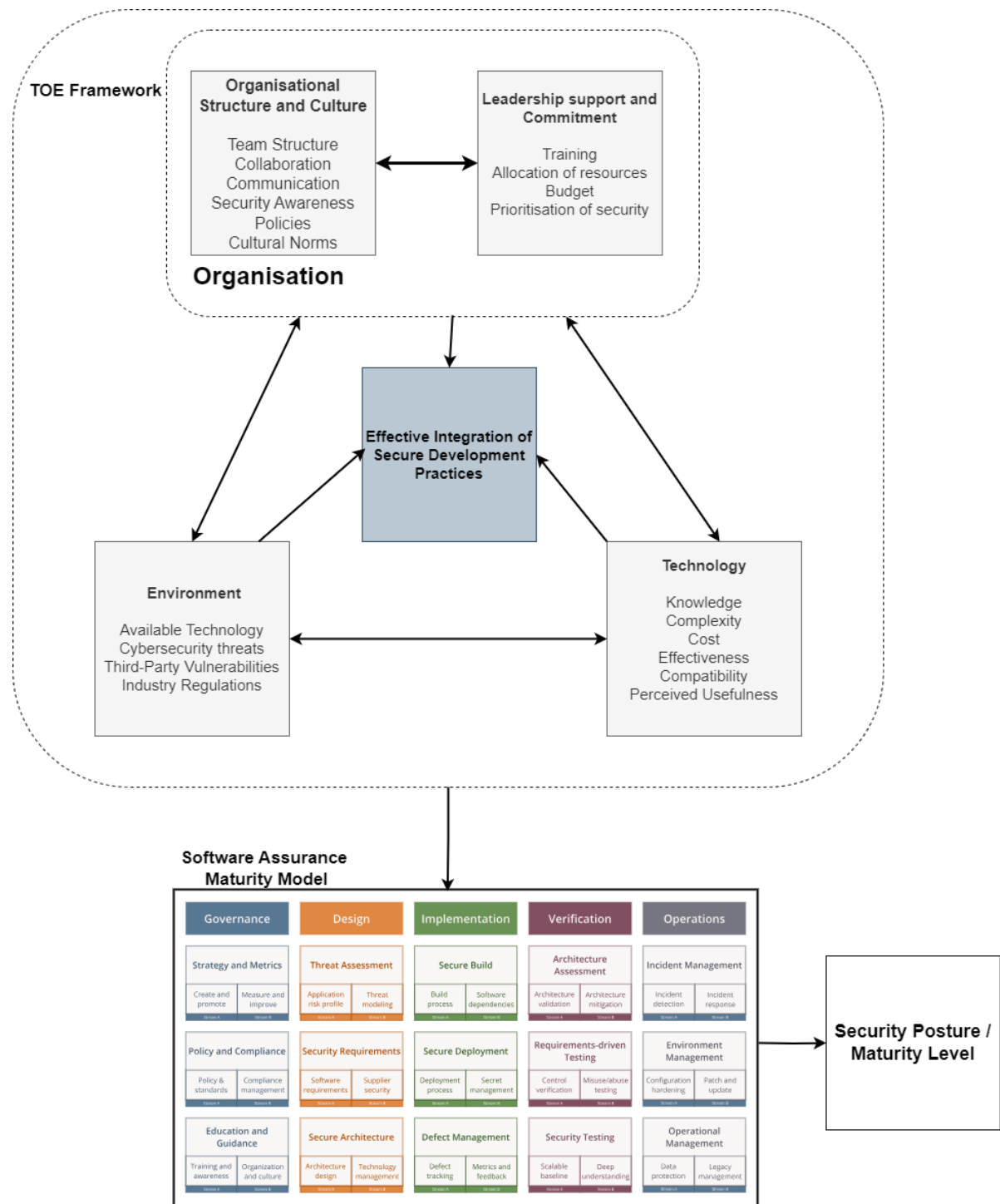


Figure 3. Conceptual Framework

2.7 Conclusion of Literature Review

The literature review highlights how organisational structure and culture, leadership and the use of security tools influence the effective integration of SDPs. The external environment influences the policies and processes adopted within the organisation

regarding security measures. A security-conscious culture within the organisation is a factor that encourages the integration of security measures within daily tasks. The role of the security champion can bridge the gap between security and agile teams. Leadership is vital in fostering a security-conscious culture and providing adequate resources to achieve this. The conceptual framework is formulated based on TOE and SAMM to answer the three research questions and validate the findings for practical recommendations.

Proposition 1: Organisational structure and culture are vital to creating a security-conscious mindset within agile teams. Within the organisational structure, the security champion role can effectively bridge the gap between security and software development teams. Cultural factors such as awareness, training, and cross-team collaboration can create a security culture, leading to the effective integration of SDPs.

Proposition 2: Leadership commitment and support are driving factors in fostering a security-conscious environment, where agile teams are empowered and possess the necessary resources, skills, and knowledge to integrate security into their processes and, therefore, increase the software's security posture.

Proposition 3: Tools and technology can help agile teams successfully integrate SDPs into their existing processes by implementing security practices within their CI/CD pipelines. Factors such as organisational readiness, compatibility, budget, and technical skills play a role in the selection and adoption of tools.

CHAPTER 3. RESEARCH METHODOLOGY

3.1 Research Paradigm

The research took an interpretivism and social constructivism research paradigm. This applied to the study since the research aim was to understand individual experiences within agile teams. A qualitative approach helped to explore how different team members perceive, adopt, and adapt SDPs within their specific context and role in the team (Saunders et al., 2023).

Ontologically, interpretivism recognises that individuals or groups socially construct multiple realities. In the context of this research, interpretivism means that the integration of SDPs is not a uniform process. Instead, each agile team's approach is influenced by its unique context, culture, and values. This paradigm allows a deep dive into the organisation's unique culture and team dynamics. It also recognises that the agile team members' individual roles, experiences, beliefs, and values shape their understanding and integration of SDPs.

Epistemologically, interpretivism emphasises that knowledge is co-created through interactions between the researcher and participants, where understanding is derived through interpretation. Hence, the researcher's role is to actively engage with the participants by interpreting their lived experiences to understand the factors influencing the integration of SDPs.

Axiologically, interpretivism acknowledges that research is inherently value-laden, with the researcher's values and perspectives influencing the research process and outcomes.

3.2 Research Approach

The research method used in this study is qualitative, with an abductive research approach. Qualitative studies foster a collaborative approach to knowledge creation, with the researcher and participants engaging in active interaction and dialogue. This approach enables a deeper understanding of the individual's perspective. Open-ended questions empower participants to articulate their experiences and perspectives, enriching the research process (Tong et al., 2007).

According to Clarke & Braun (2017), abductive reasoning involves starting with specific observations or data and then moving towards broader generalisations or theories. In this study, the abductive approach allows for the emergence of themes and patterns from the data as well as from pre-existing theories or frameworks. This flexibility enabled exploring the nuanced factors that influence the integration of SDPs in agile teams, as these factors may vary across different contexts and teams.

3.3 Research Design

The research employed a generic qualitative design. In generic qualitative studies, people's beliefs, attitudes and lived experiences are studied (Percy et al., 2015). This

was selected as it provided a flexible approach to qualitative research, drawing on the facets from other designs such as ethnography, phenomenology, narrative, case studies and grounded theory. A generic design does not adhere to the rigid guidelines of specific qualitative design methodologies, thus allowing themes and patterns to emerge directly from the data collected. It has been used in an increasing number of research submissions (Ellis & Hart, 2023). An exploration of the phenomenon is supported without being constrained by pre-existing theoretical frameworks. Adaptability is practical when studying a dynamic and evolving field like security in software engineering in agile contexts, as it enables a response to emerging insights and findings throughout the research process (Ellis & Hart, 2023).

3.4 Data Collection Methods

Data was collected through semi-structured interviews. Semi-structured interviews excel at capturing detailed, contextual information on the effective integration of SDPs. This method captured team interactions and individual interpretations, offering insights into the underlying processes as well as organisational structures, cultural, leadership, and technology-related factors that hinder or enable the integration of SDPs.

3.5 Population and Sample

3.5.1 Population

The population of participants consisted of IT professionals and experts working in agile software development teams. This included IT Managers, Scrum Masters, Product Owners (PO), Software Architects, Software Engineers, DevOps Engineers, and QA Engineers at the intermediate to senior level.

Participants in these roles were selected because they formed part of agile teams and contributed to prioritising, implementing and delivering working software. The diversity of roles allowed for the collection of in-depth data and varied perspectives on the factors influencing the integration of SDPs, ranging from technical challenges to organisational barriers and cultural norms. IT Managers were selected to provide insights from a leadership perspective and their involvement in the operational aspects of the team. The participant information is shown in Table 1, indicating the number of participants for each selected role.

Table 1. Sample Population

Role	Number of participants
IT Manager	2
Scrum Master / Project Manager	2
Product Owner (PO)	2
Software Engineer/Developer	2
QA Engineer	2
DevOps Engineer	2
Software Architect	2

3.5.2 Sampling and Sampling Method

Purposive sampling was imposed, as participants were selected based on their specific knowledge and experience with agile software development (Maxwell, 2013).

Participants were identified based on their roles and experience in the financial services industries. A total of 14 participants were invited to be interviewed from 3 different organisations in the South African financial services industry. This allowed for a broad range of perspectives to be investigated, and a saturation point to be reached in terms of the data collected.

Although there is no set number of participants required for qualitative studies, the guideline is between 6 and 12 to reach theoretical saturation (Guest et al., 2006). While Creswell & Creswell (2018) recommend between 5 and 25 for phenomenological studies and between 20 and 30 for grounded theory. Since this study used a generic qualitative approach, a total of 14 participants using purposive sampling were considered sufficient to reach theoretical saturation.

3.6 The Research Instrument

Since data was collected through semi-structured interviews, the research instrument was an interview guide where open-ended questions were asked. The interview questions differed slightly for the separate roles of the participants to gain more perspectives. The conceptual framework using TOE and SAMM guided the formulation of the questions. The research instrument is found in APPENDIX D: Research Instrument.

3.7 Procedure for Data Collection

One-on-one interviews were conducted virtually and facilitated through Microsoft Teams. The interviews lasted between 30 and 45 minutes and were audio-recorded with the participant's permission. A transcript was captured with Microsoft Teams.

3.8 Data Analysis Strategies and Interpretation

This qualitative study employed a thematic analysis approach for data analysis. This method involved a systematic process of identifying, analysing, and reporting patterns within the data (Creswell, 2013). The data was first organised, and the researcher became familiar with the data, coding and organising themes, representing the data, and forming an interpretation of the results. Creswell (2013) describes this data analysis process as “The Data Analysis Spiral”, in which all the steps are interconnected.

The interpretivism paradigm guided the interpretation of the data, acknowledging that meaning is constructed through the interaction between the researcher and participants. The aim is to understand the participants' subjective experiences and interpretations regarding the integration of SDPs in agile teams. Qualitative analysis software, NVivo (Version 14), was used to assist with the coding.

The interview data was validated using OWASP SAMM v2 to assess each explored construct's effectiveness and influence and how they affect one another. This provided empirical and practical findings for the study so that feasible recommendations could be made. The overview of how this was done is depicted in Figure 4.

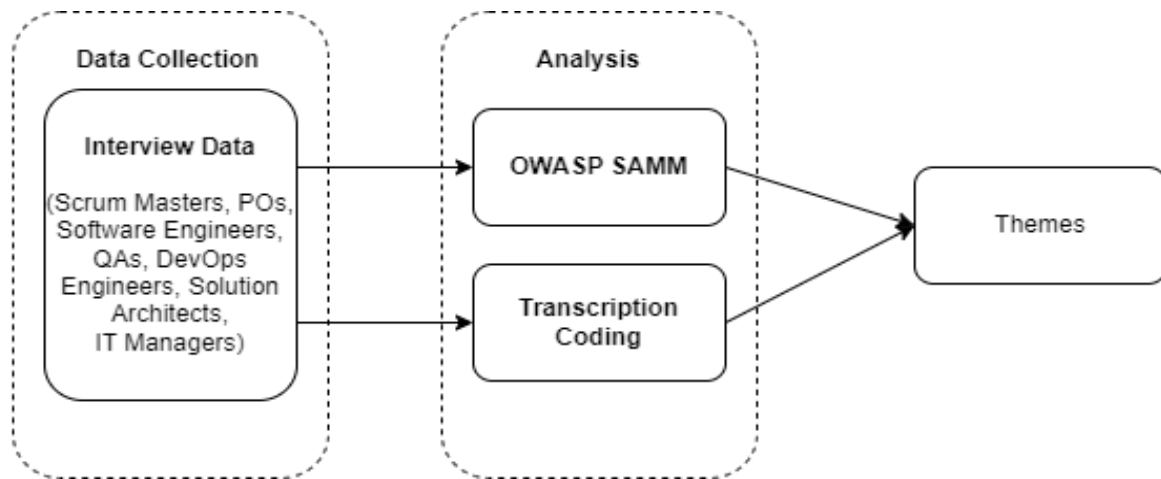


Figure 4. Data collection and analysis overview

The data was analysed using an abductive thematic analysis approach, integrating both inductive and deductive coding methods (Naeem et al., 2023). This allowed for the identification of codes and themes based on the study's conceptual framework while also enabling the discovery of new insights emerging from the collected data. The steps taken during the analysis phase are depicted in Figure 5 and explained below.

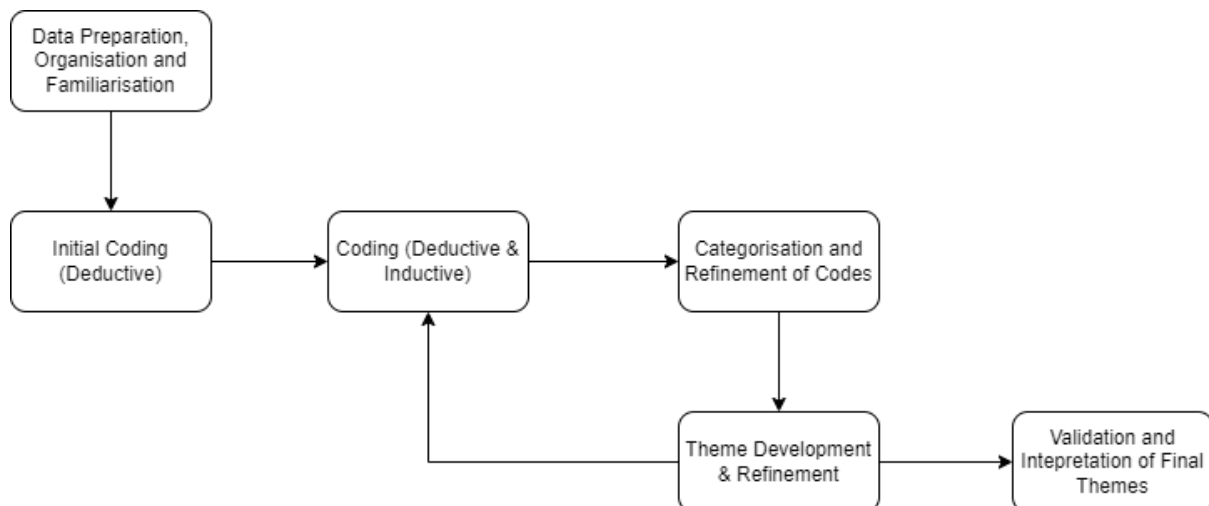


Figure 5. Thematic Analysis Procedure

Following the completion of interviews, audio recordings were transcribed verbatim to ensure an accurate representation of participant responses. These transcripts were

then imported into NVivo for analysis and coding. An initial familiarisation phase was conducted to immerse in the data, where transcripts were read and re-read multiple times to identify patterns, recurring concepts, and potential new areas of interest (Bingham, 2023). During this phase, notes and annotations were made to capture initial thoughts and reflections on possible themes.

NVivo allows for the hierarchical organisation of codes, categories, and themes. A structured coding framework was first established, ensuring that each research question was represented by a parent code. The coding followed an abductive thematic analysis technique, combining deductive and inductive coding.

In the initial coding phase, deductive coding was employed and based on the conceptual framework. A priori codebook was developed, aligning with the study's research questions and theoretical constructs. These predetermined codes provided structure and allowed for a focused analysis of expected themes. Each text segment was assigned a code; where necessary, multiple codes were applied to a single passage if it aligned with more than one code. As the transcripts were coded, new codes emerged that were not part of the initial framework. These inductive codes were iteratively added to capture new patterns and insights that participants highlighted.

Once the coding phase was completed, codes were categorised into broader groups to form higher-level categories. The categorisation process followed an iterative and reflexive approach, ensuring that codes were merged, split, or reorganised based on conceptual similarities and relevance (Williams & Moser, 2019).

The codes were systematically reviewed to identify themes and categories that reflected meaningful patterns in the data. Themes were assessed based on the relevance to research questions, ensuring alignment with research questions. Codes were categorised by examining whether the grouped codes logically fit under a theme. This coding and refinement process was conducted iteratively over three cycles by reviewing the coded transcripts and categories to ensure that emerging themes accurately represented participants' perspectives.

Chapter 4 presents the final themes, categories, and codes analysed to provide practical insights for SDPs within agile teams. By applying this structured thematic analysis process, the study ensured depth in qualitative data interpretation, leading to

a comprehensive understanding of the factors influencing SDP integration in agile teams.

3.9 Possible Limitations and Challenges

A possible limitation of the study was that participants were not willing to provide full insights on the security of the software applications due to negative perceptions about the security posture of the organisation, particularly in an interview setting.

3.10 Quality Assurance

To demonstrate rigour and trustworthiness in the research, multiple methods to ensure the validity of the qualitative research were employed as recommended by Creswell and Creswell (2018).

3.10.1 Transferability

Purposive sampling was used to carefully select participants that represented a diverse range of experiences and contexts, enhancing the potential transferability of findings to other similar settings (Maxwell, 2013). The participants were selected from the financial services industry in South Africa and had many years of experience as an IT professional.

3.10.2 Credibility

To ensure credibility, member checking and peer debriefing were used. The data collected from the interviews was shared with the participants to confirm the accuracy of the interpretation. Peer debriefing was performed with other IT professionals to review the collected data and interpretations.

3.10.3 Confirmability

Reflexivity was practised to provide an objective analysis of the collected data, ensuring that personal bias and assumptions did not influence the collection, analysis and interpretation of the results.

3.10.4 Dependability

A detailed record of the research procedure was taken and maintained in an audit trail. The audit trail contained the data collection and analysis procedures, and any

decisions or challenges encountered during the data collection phase was documented.

3.11 Ethical Considerations

The research process followed all ethical standards prescribed by The University of the Witwatersrand. Data collection only began once ethical clearance had been obtained in writing. The signed Ethics Clearance Certificate can be found in APPENDIX E: Ethics Clearance Certificate.

All participants were provided with the Participant Information Form and the Participant Consent Form, which they were required to sign before the interview was conducted. These forms can be found in

APPENDIX A: Participant Information Form and APPENDIX B: Participant Consent Form respectively. This was to make the participant aware of what their involvement entailed and how the data was collected, stored, analysed and reported.

Participating in the research posed minimal risk to the participants since no biographical information was collected, and no sensitive or confidential topics were discussed. No information was asked that could reveal the participant or the organisation in which they are employed. No confidential information about the organisation was discussed, and the questions did not infringe on the intellectual property of the participant or their organisation. The participant group consisted of IT experts and professionals working in the South African financial services industry. The interview guide was based on questions about their everyday work lives and activities.

CHAPTER 4. PRESENTATION OF FINDINGS

4.1 Introduction

Chapter 4 highlights the findings and key themes derived from the semi-structured interviews conducted in the qualitative study. The aim of the study was to investigate the how organisational structures and culture, leadership and technological factors influence the effective integration of SDPs within agile teams.

The data was collected by interviewing eleven IT professionals working in the financial services industry, including banks and insurance companies. Some of the participants had held multiple roles within their organisations and provided insights from their experiences in these roles. For example, Solution Architects and Software Engineers have performed DevOps tasks in their organisations.

Three of the participants who were invited declined to participate in the study. One invited candidate noted that they were not comfortable discussing cybersecurity in an interview. Although three of the participants did not participate, saturation was reached at ninth interview, after which no new themes were emerging. With majority of the participants having more than ten years of experience as an IT professional in the industry, they were able to provide a great depth of insights of the topic.

Table 2. Participant Information

Participant	Current Role	Years of experience
P1	Solutions Architect	15
P2	Development Manager	20
P3	QA / IT Manager	30
P4	Software Engineer	10
P5	Software Engineer	5
P6	Product Owner	10
P7	Scrum Master/Project Manager	20
P8	Software Engineer	5
P9	Systems Analyst	20
P10	DevOps Engineer	3
P11	Security Manager	20

4.2 Thematic Analysis

Thematic analysis was performed with an abductive approach, and the following codes, categories and themes were generated through the process described in Chapter 3. The codes, categories and themes for all the research questions are presented below in Table 3.

Table 3. Thematic Analysis Results

Research Question	Codes	Category	Theme
RQ 1: Organisational Structure and Culture	Mandatory cybersecurity training Voluntary specialised security training High level of security awareness Resistance to cultural changes Security-conscious mindset and culture	Training and Awareness	Organisational Culture
	Feature delivery pressure Agile mindset focused on speed Security as a non-functional requirement Security as an afterthought	Prioritisation of Feature Delivery Over Security	
	Hierarchical communication Inefficient information flow Communication with stakeholders on security issues Policies and documentation not accessible Siloed security teams Security team as an accessible resource	Organisational Communication	Organisational Structure
	No dedicated security champion Separate Security Team	Team structures	

Research Question	Codes	Category	Theme
	Proactive collaboration between security and development teams		
	Organisational policies and frameworks for compliance Security in software development policies Rigorous security testing policies	Organisational Policies and Security Frameworks	
	Industry-specific standards and compliance requirements Data protection laws Business-driven deadlines	Industry Environment	Environmental Factors
	Security is never-ending New threats always coming up	Security Threat Landscape	
RQ 2: Leadership Commitment and Support	Reactive leadership approach Short-term remediation efforts Leadership has a good understanding of security requirements Emphasis on delivery over security Lack of awareness of impact	Reactive vs. Proactive Leadership Approaches	Leadership Commitment

Research Question	Codes	Category	Theme
	Inadequate investment in training Time allocation for security Understaffed security and development teams Budget limitations	Resource Allocation and Investment	
	Leadership empowerment for security initiatives Leadership considers improvements and suggestions Architects as an accessible resource Security tickets given priority High level of management support and security awareness Leadership has a good understanding of why security can cause delivery delays	Leadership Support for Security Initiatives	Leadership Support
	Personal development and self-learning Continuous learning promoted Encourage collaboration between different disciplines	Encouragement of Personal Development and Self-Learning	

Research Question	Codes	Category	Theme
RQ 3: Tools and Technology	Seamless integration of security tools Automation of security scans Integrated security feedback Continuous monitoring Delays due to lack of knowledge Tools and Technology increase agility	Integration and Automation of Security Tools	Integration and Automation
	Security risks in legacy applications Modernising legacy systems to enhance security Upgrading legacy security infrastructure comes with risks Legacy system compatibility Ensuring tools and technology are up to date	Challenges with Legacy Systems	
	False positives and delays Multiple tools and technologies required Tools are cost-effective and reliable Impact on performance and usability Ease of use Security can be taken for granted with too much reliance on tools	Tool Effectiveness, Reliance and Usability	Tools and Technology

Research Question	Codes	Category	Theme
	Processes are manageable, repeatable, and deployable Tools are selected from an organisational level in a systematic manner Penetration testing done outside the agile team Static code analysis tools Dynamic application testing tools	Tool Selection and Deployment	

4.3 Findings of Proposition 1

To answer RQ 1, this study sought to explore how organisational structure and culture influence agile teams in effectively integrating SDPs into their workflows and processes.

Proposition 1: Organisational structure and culture are vital to creating a security-conscious mindset within agile teams. Within the organisational structure, the security champion role can effectively bridge the gap between security and software development teams. Cultural factors such as awareness, training, and cross-team collaboration can create a security culture, leading to the effective integration of SDPs.

A word cloud was generated from NVivo, with the responses relating to the themes identified for RQ 1 and depicted below in Figure 6. This word cloud visualises the recurring terms related to RQ 1 on how organisational structure and culture influence the integration of SDPs within agile teams. Dominant terms such as “security”, “organisation”, “development”, “team”, “practices”, and “training” highlight the emphasis the organisations have placed on security, creating a security-conscious culture and the importance of communication and training within agile teams. The prominence of these terms reflects participants' discussions about organisational policies, processes and team dynamics.

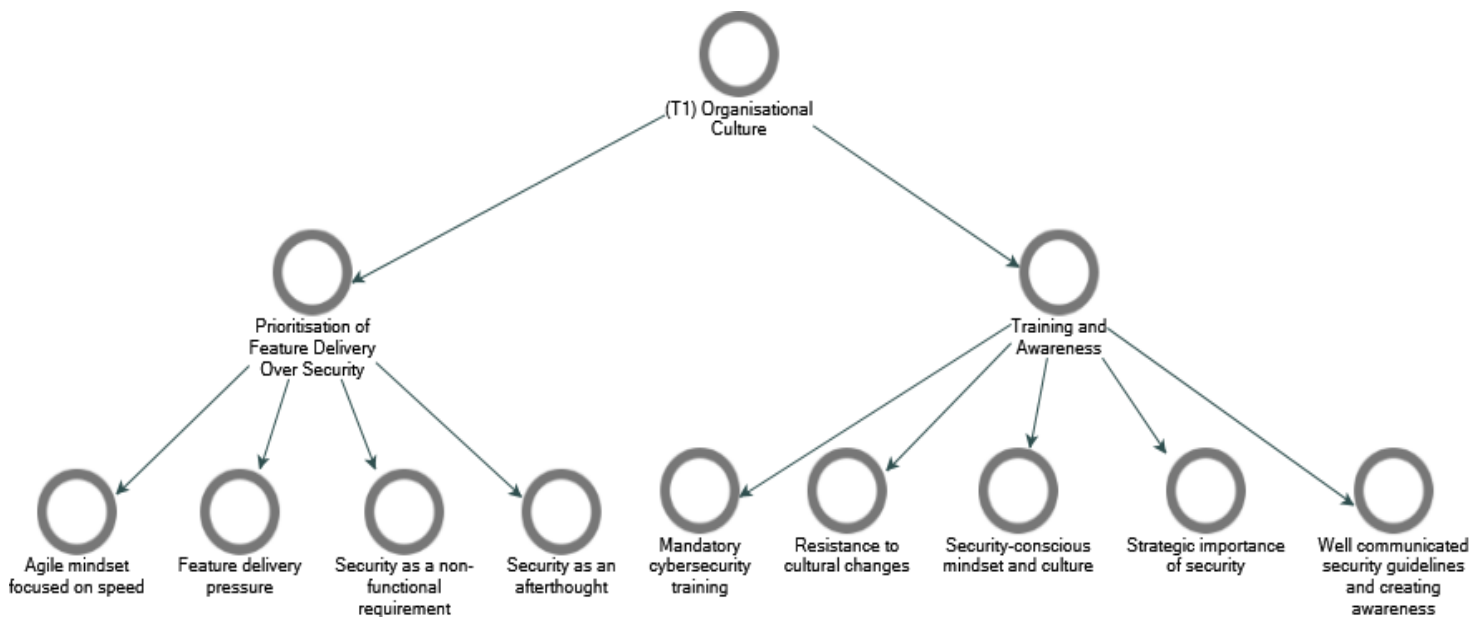


Figure 7. Theme, category and code hierarchy - Organisational Culture

4.3.1.1 Training and Awareness

When asked to describe the security culture with the organisation, all the participants indicated that security is given a high importance and in general given a lot of attention.

“From a cultural perspective, the organisation has placed a high priority on security since COVID-19, and security has become relevant. On a broad cybersecurity level, the organisation creates a high level of awareness, for example, frequent phishing emails as a test.” – P2

“Security is quite prominent from a team level all the way up the organisational level. It is constantly communicated.” – P9

All the participants mentioned that there is cybersecurity training that is offered to all staff. However, half of the participants indicated that it was mandatory to complete. With regards to more specialised security training for technical staff, there seemed to be a lack of comprehensive training available, with very basic software engineering training offered, but not compulsory to perform their tasks.

“In terms of cyber security, the organisation provides training videos on phishing attacks and malware attached to emails. These are mandatory, and it's audited to ensure everyone has watched the training videos” – P3

“Mostly there are security courses that are mandatory to complete. Depends on the role that you perform.” – P4

Participants were asked about some of the cultural norms regarding security. A recurring theme among the participants indicated that the organisation has a security-conscious mindset and culture.

“There is a security-first approach that takes into account security during development and whatever technology we are working on. Security comes first, and there is no compromise on that.” – P4

“I think from a culture point of everyone in the team understands how important testing and making sure everything is secure, before we push to prod.” – P9

Participant 1, Participant 3 and Participant 7 echoed the sentiment that although the organisation has prioritised security, there are small pockets within the organisation that don't take security as seriously and a mind shift change was required.

“But when you look closely at individual teams, there are some that take it seriously and some that don't consider security as important.” – P1

“This required a mind shift change, because every little piece of code would need to be security tested.” – P3

“With regards to security itself, it is about instilling the security conscious mindset within the smaller organisation. There has been a lot of resistance to change and a change management process, with campaigns, events and team building.” – P7

A significant number of the participants spoke about how their organisations create awareness and frequently communicate about cybersecurity. Emails on topics such as phishing, new security threats, how to avoid being scammed, and protecting the organisation and yourself as individuals were sent out regularly.

“In terms of creating cyber security awareness, there is a lot of awareness around rolling out and placing restrictions” – P6

“In terms of communication, we receive email communications quite regularly. This is about current breaches that have occurred at other companies, or new ways of phishing attempts and threats.” – P9

IT professionals in a management role were asked how they create awareness and advocate for security within the organisation. Participant 2 and Participant 11, both cited communication as a way to create awareness and advocate for security.

“Ensuring code reviews are performed and frequently communicating the importance of security to all members of the development team.” – P2

“When it comes to communication, it's more around education, so we are really communicating in terms of security and what you can and can't do. It's not only cyber security, but also physical security. So, we do emails and provide short courses, and some are compulsory. We would even do a group training session, in the case of an incident occurring.” – P11

Participant 3 noted ensuring that their teams adhere to the minimum mandatory practices and encouraging collaboration between different disciplines in the team.

“Ensuring that the minimum mandatory practices also apply to the QA teams. Encouraging collaboration with the developer and discussing features that they are building can also lead to questions about whether they have considered aspects that could impact security.” – P3

4.3.1.2 Prioritisation of Feature Delivery over Security

Participants were asked to describe the cultural norms regarding security, and a significant number mentioned that high workloads and market pressure can often lead to the cultural norm of prioritising feature delivery over security requirements. Participant 1 mentioned that the agile mindset focuses on rapidly delivering features to customers.

“Within agile teams especially, the focus is on rapid delivery of working software. Security not a big deal for some of the teams because the key mindset is that we need to deliver a feature because the business needs it” – P1

Participant 2 and Participant 8 highlighted that the pressure to deliver new product features may sometimes lead to security being overlooked or considered the last step of the SDLC.

“In some instances, security can be overlooked as there is an urgency to get product features delivered to the customer.” – P2

“When there is an urgency on delivery, then it can influence security as something could be missed during development or peer review. If there are any security issues picked up during the development, then they are completed later as hotfixes. So, this would be at the final stage of the development process, after it's gone to production”

– P8

Participants were asked how they balance security requirements with other product features and how security issues are prioritised within sprints, and it was noted that high-ranking security threats are always prioritised, as per the organisational policies. All other participants confirmed that this was a common organisational policy and that all teams adhere to it.

“Depending on the severity level of the security threat reported, critical and high issues are picked up immediately. Lower levels are discussed in the sprint planning and planned accordingly.” – P4

“The high security, blocking issues are given priority so that you know it goes into the Sprint. When it comes to issues affecting a member's usability, we will prioritise items that could cause reputational damage.” – P9

4.3.2 Organisational Structure

A graphical representation of the theme, category and code hierarchy was generated from NVivo and presented below in Figure 8. The broad theme of Organisational Structure had three categories, namely, Organisational Policies and Security Frameworks, Organisation Communication and Team Structures, for which the findings are presented.

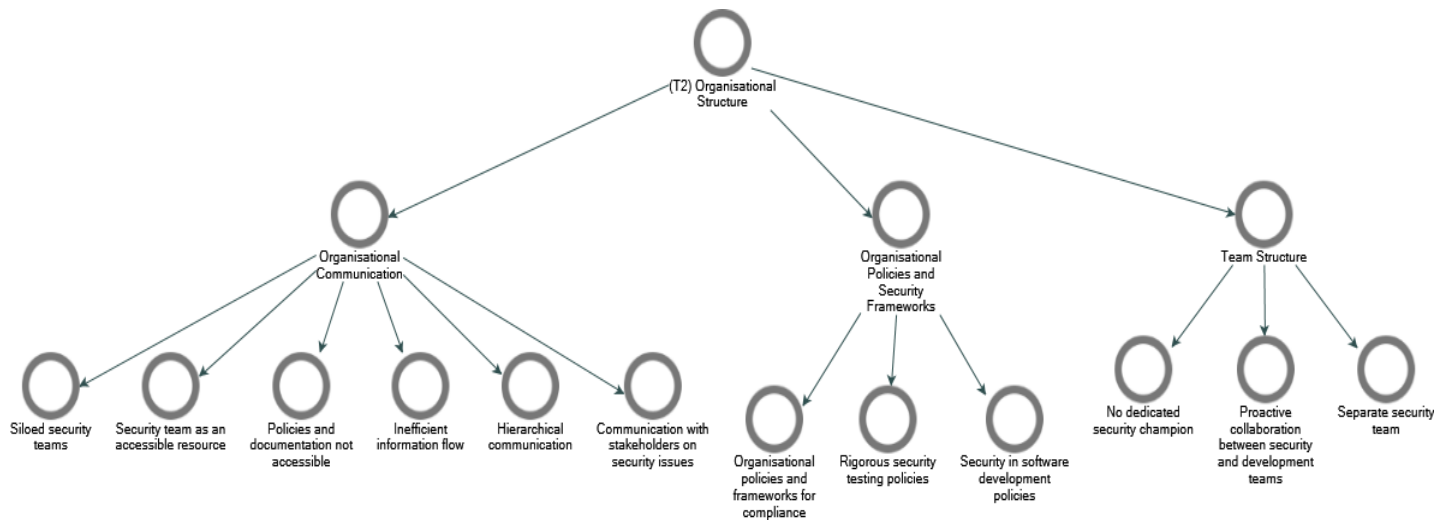


Figure 8. Theme, category and code hierarchy - Organisational Structure

4.3.2.1 Organisational Policies and Security Frameworks

Participants were asked to describe the policies and standards set by their organisations. The organisations from which the participants are employed have established comprehensive security policies and frameworks to ensure secure development practices. These policies include mandatory practices for both software development and testing, spanning areas such as cloud security controls, testing protocols, and risk assessments. The policies are designed to provide a consistent, organisation-wide approach to security, although the level of rigor and adherence varies across teams and projects.

“Divisions in the organisation such as Infrastructure and Operations also have security controls in place on the cloud environments.” – P1

“At the organisation, there is a minimum set of mandatory practices that are prescribed, and all software development teams are required to adhere to them. This also includes software testing.” – P3

Risk assessments were another essential part of the organisational security framework. Participant 7 noted that the organisation held workshops with cross-functional teams, including legal, compliance, and IT security, to review potential security risks before project initiation. During these assessments, teams evaluated IT risks associated with each new product and established protocols to mitigate them. This collaborative approach ensured that security requirements were clearly defined

early in the project lifecycle, aligning all stakeholders on the importance of security compliance.

“We have risk assessments, where IT security is also included. Within this phase we will answer all the IT risk questions relating to the product that we are going to build. With workshops, we take the other teams (legal, compliance, etc) through the project, including security. This ensures that we are meeting all the security standards.” – P7

There were policies in place for both software development and testing. These were set out by the organisation and filtered down into the teams by management and team leads. All teams were expected to adhere to these practices. Most of the participants mentioned the rigorous testing policies that were in place. These policies required that all applications undergo comprehensive security testing, including penetration testing, code scans, and both positive and negative testing to validate security effectiveness.

“A norm within the team is that testing is never bypassed. All new applications are always sent through to the cybersecurity experts to scan and review and they will come back with a report. There’s always test evidence, not just for the sake of ticking a box, but ensuring negative testing, positive testing and regression testing is done.” – P9

“There are best practices that each developer needs to follow. There are basic guidelines provided and peer review within the team.” – P4

4.3.2.2 Organisational Communication

The organisations from which the participants were interviewed all had separate security teams that managed cyber security and set standards and policies from an organisational level. FSPs are generally large corporations with multiple business units, and each business unit had its own cybersecurity team that manages this from a business unit level. This structural layout has a very hierarchical communication flow, with no direct communication between the agile team and the security team.

“We don’t have direct communication with them, and it often goes through a lot of people before it reaches the development team. The security team would often communicate with the management or leadership team.” – P5

This has led to an inefficient flow of information and often back and forth over email, or meetings between a representative of the development and security team as indicated by Participant 4.

“There are monthly/biweekly with the security team and a representative from our team will need to be present. The representative will then communicate with the project team on any findings and if anything needs to be resolved and we get feedback in this manner.” – P4

Although most participants felt that this way of communication was not very effective.

“There is a communication breakdown when this occurs, as security requirements might only get filtered down successfully, if the issue is really critical and can result to a reputational damage. A lot of people within the development team might not even know that a security team exists and who they are.” – P1

“It’s a lot of downstream information that takes time to reach the relevant people, but I’m not sure how we can streamline that further.” – P10

Participants in management roles indicated that this method of communication was proactive between the teams and this partly because these team members had direct contact with the security team.

“So, I’ll say communication’s good. It starts off normally via e-mail and it can go to actual meetings depending on the need for it.” – P9

“And we normally work in collaboration with the teams to ensure that is up to the standard and that it also meets our guidelines.” – P11

Majority of the participants thought that having a separate security team with no direct contact siloed the teams with these sentiments highlighted by Participant 1 and Participant 7.

“The security team are in their own bubble and only communicate when a problem is detected.” – P1

“It feels like we work in silos. Since there is a separate security team that takes care of the policies.” – P7

4.3.2.3 Team Structures

Participants were asked how the team's structures either helped or hinder the integration of security, and if a security champion exists.

Participant 8 indicated that it was preferable to have a more accessible security team.

“I would prefer if the security team were more accessible to the development team, so that any questions and concerns regarding security can be discussed.” – P8

In all the organisations the participants were interviewed from, there was no dedicated security champion, and the role didn't exist within the agile development team. When the role of the security champion was described to the participants, all agreed that the development team would benefit from having this skillset within the team. A few participants indicated that a person in a technical role such as a senior software engineer or architect would have security initiatives as part of their responsibilities.

“This person is also a senior within the development team who can provide support if further assistance is required” – P4

Regarding the communication of policies and documentation at the organisational level, there is a substantial amount of material available. However, some participants expressed concerns that these resources are not easily accessible, and communication was poor as highlighted by Participant 5 and Participant 6.

“In terms of communication, I feel there is a gap somewhere but can't pinpoint it. Although the documentation could be done better, updated regularly and have us be reminded of where we can find them.” – P5

“I do believe that they do things from a cyber security perspective, however, is communicated well.” – P6

Product Owners and IT managers were asked to describe stakeholder communication between the teams when security issues are picked up. This is usually done when the stakeholder can be affected by a delay.

“In the form of a meeting and showcase why it is actually a problem. This is more important than the development of a feature because it impacts the business, so it's always around business impact.” – P6

“When communicating to a manager or the team, we'll communicate any critical vulnerabilities that would impact them. And when dealing with a development team, we will give them all the technical details and also provide detail on how to solve for it.” – P11

4.3.3 Environmental Factors

A graphical representation of the theme, category and code hierarchy was generated from NVivo and presented below in Figure 9. The broad theme of Environmental Factors has two categories, namely, Industry Environment and Security Threat Landscape, for which the findings are presented.

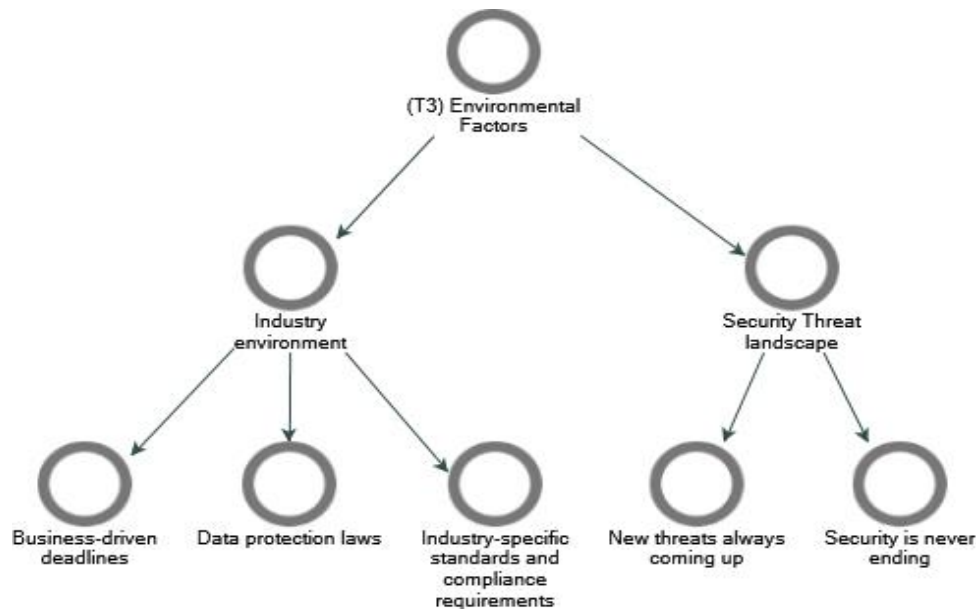


Figure 9. Theme, category and code hierarchy - Environmental Factors

4.3.3.1 Industry Environment

A recurring theme from the participants where the external factors that impact the adoption of security requirements. Participants understood the impact of working in the financial services industry and security, as put by Participant 6 and echoed by Participant 9.

“Within the industry, they are very stringent with security requirements and not lenient even in the event of an issue that requires support. This impacts the business workflow and hinders operations.” – P6

“Because we work in the financial industry, specifically medial aid and administration, we work with member data and even our member’s money. It’s a very similar industry to banking, and also because we run 24/7 and the systems are up all the time.” - P9

Changing industry regulations and data privacy policies require FSPs to adhere to them. Participant 8 recounted an event where a possible breach of POPIA could have occurred and received adequate support from leadership.

“Policies like POPI can make this a major issue. Since we were able to quickly prioritise the hotfix, leadership supported this.” – P8

The current competitive landscape and business demands put agile teams under pressure to ensure that software is delivered, and this sentiment was echoed by Participant 7, Participant 10 and Participant 11.

“The focus right now is to deliver, deliver, deliver. Delivery dates are just dictated, before anything can be planned.” – P7

“There is a strong focus on feature delivery over nonfunctional requirements, and this is because its more measurable to business.” – P10

“From a negative point, we are always busy, there is always a lot of work, and many ongoing projects.” – P11

4.3.3.2 Security Threat Landscape

During the interviews, a recurring theme from the participants was an acknowledgement that security is an initiative that must be continuously monitored, and as Participant 4 mentioned it is a *“never-ending story”*.

“Security is never complete because there is always something new to secure or improve.” – P11

It was observed that the organisation endeavoured to educate its employees about new threats by communicating via email and updating training activities. Regular meetings with senior leadership and technical architects ensure that new vulnerabilities, security policies, and updates are discussed.

“The organisation keeps us updated with mail shots about new vulnerabilities, phishing techniques, or ransomware. These updates ensure we are aware of the latest threats; even as new ones arise almost every day” – P11

“The organisation has monthly security forums, in which these topics are discussed.” – P1

Software security scans that are run periodically or every time a feature is developed had become an integral part of the process to ensure that any new vulnerabilities that arose can be detected and resolved timeously.

“We are constantly updating our applications and infrastructure to counter new threats. For instance, when a new security vulnerability was found in a library we use, we had to patch the issue and upgrade the library immediately.” – P9

“New threats constantly arise, especially in the cloud. We’ve implemented 2FA for all programmatic access and continuously monitor for evolving security challenges.” – P10

4.4 Findings of Proposition 2

To answer RQ2, the study explored how leadership commitment and support influenced the ability of the agile development teams to integrate security into their processes.

Proposition 2: Leadership commitment and support are driving factors in fostering a security-conscious environment, where agile teams are empowered and possess the necessary resources, skills, and knowledge to integrate security into their processes and, therefore increase the software's security posture.

A word cloud was generated from NVivo, with the responses relating to the themes identified for RQ 2 and depicted below in Figure 10. This word cloud visualises the recurring terms related to RQ 2 on how leadership commitment and support influence the integration of SDPs within agile teams. Dominant terms such as “security”, “leadership”, “management”, “support”, “risks”, and “deliver” highlight the role of leadership’s influence on fostering an environment where team members feel empowered to raise security concerns and have the necessary resources available.



Figure 10. Word Cloud related to findings for RQ 2

4.4.1 Leadership Commitment

A graphical representation of the theme, category and codes hierarchy was generated from NVivo and presented below in Figure 11. The broad theme of Leadership Commitment has two categories, namely, Reactive vs. Proactive Leadership Approaches and Resource Allocation and Investment for which the findings are presented.

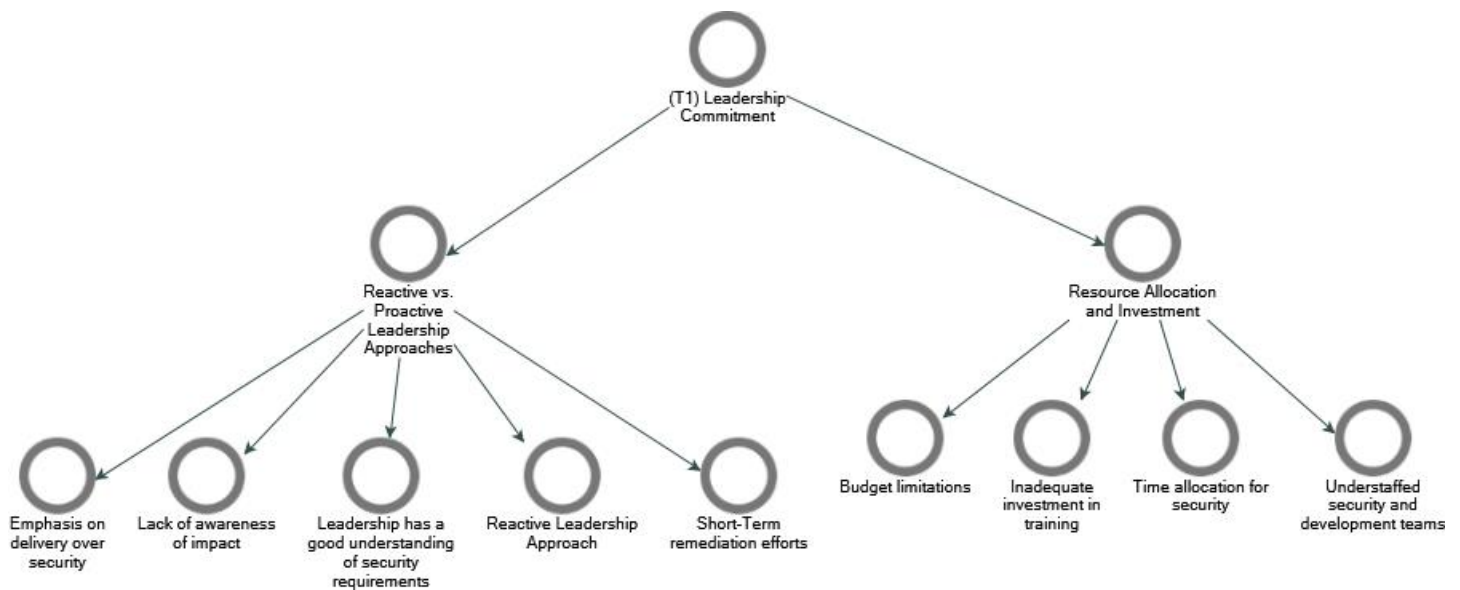


Figure 11. Theme, category and code hierarchy - Leadership Commitment

4.4.1.1 Reactive vs. Proactive Leadership Approaches

Participants were asked to describe the level of support from leadership for the integration of SDPs. The findings that arose from this question indicated that there in some organisations there was a high level of support and understanding of security impact from leaders. Half of the participants indicated a reactive approach from leadership where they only acknowledged an issue when there was a direct impact on their operations.

“Leadership doesn’t show a high level of support on security and only when upper management raises an issue, is time allocated to focus on security concerns.” – P1

“And I think they would only acknowledge security as important when they see like a breach or something. So, unless they have like a major impact in that way, will they try and prioritise it, but it can be too late at this point.” – P10

“Leadership seems to support security only at a low level unless it directly impacts them. The responsibility is left to the employee.” – P6

Three participants mentioned a lack of understanding and awareness from leaders on security issues as the reason for no support. A recurring theme linked to leadership support was the emphasis on feature delivery. Short-term remediation was also

implemented for security issues, but no long-term strategy was established to address these issues upfront.

“I have raised risks around this, and I ask leadership to accept the risk before putting it into production. So, I don’t get the support and commitment from leadership around security.” – P7

However, the other half of participants indicated high level of support from leadership as Participant 3 noted that leadership is help accountable for security.

“Leadership provides a high level of support, as they are also held accountable for the application’s security. So, their jobs are on the line, and they are under a lot of pressure.” – P3

Participants 3 and 8 mentioned security hot fixes are often implemented after a feature has gone to production and security vulnerabilities are detected.

4.4.1.2 Resource Allocation and Investment

Participants were asked what kind of support such as resources, training, incentives did they receive from leadership to integrate secure development practices.

All participants indicated that they would require allocated time to work on security requirements or upskilling themselves. They emphasised the difficulty in prioritising security tasks amid other pressing work demands.

“With the amount of other work to be delivered, it becomes difficult to prioritise the low and medium vulnerabilities.” – P2

“The QA team did also not have the capacity to test anything else other than the features, regression testing and bug fixes.” – P3

“I would need to focus specifically on DevSecOps. There are a lot of moving parts in the process, so focusing on security would mean less time to focus on other parts.” – P10

Leaders were asked what challenges they face in getting resources, budget and a shortage of cybersecurity skills were indicated. Often there is no software engineering or testing capacity available and security scans or testing needs to be outsourced and it can become expensive.

“Another challenge we find is that getting experienced cybersecurity staff is costly. If we have to hire junior staff, it takes a long time until they are up to speed and perform at the level that we require.” – P11

4.4.2 Leadership Support

A graphical representation of the theme, category and codes hierarchy was generated from NVivo and presented below in Figure 12. The broad theme of Leadership Commitment has two categories, namely, Leadership Support for Security Initiatives and Encouragement of Personal Development and Self-Learning for which the findings are presented.

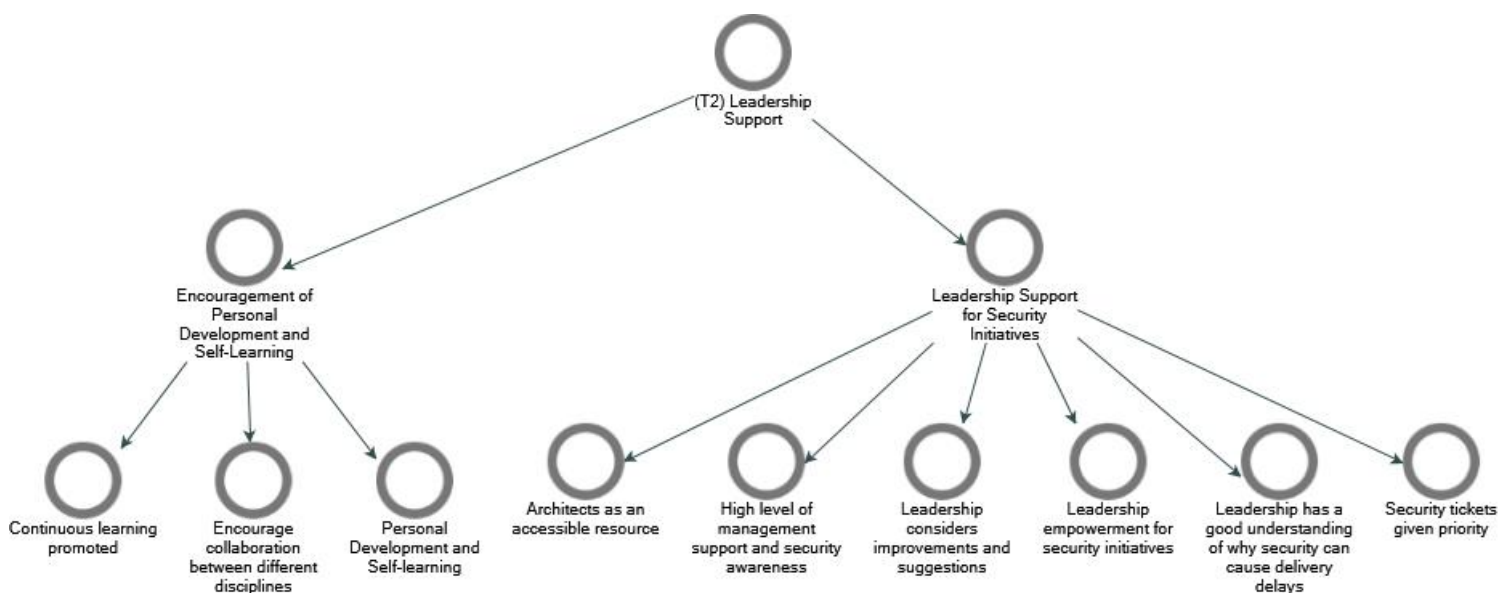


Figure 12. Theme, category and code hierarchy - Leadership Support

4.4.2.1 Leadership Support for Security Initiatives

Participants were asked if they felt empowered to raise security concerns. Despite challenges, participants generally felt empowered to raise security concerns, indicating a level of openness and support from leadership for security initiatives. Participant 9 emphasised the inclusive environment.

“People, be it a developer, senior developer, team leader, analyst or product owner or whoever, everyone is free to speak. And to put forward their ideas on how to make sure that what we put out there is best in market once again sometimes the out of the

discussion comes things usually take a bit longer to implement that if you just did it the normal way.” – P9

Participants were asked to share an instance where leadership's actions or decisions directly influenced the team's ability to implement security practices. In general, the participants' responses indicated that leadership becomes more involved when the risk and impact are higher. These examples illustrate that leadership is responsive and supportive when security concerns are raised, enabling teams to address critical issues promptly.

“There was a high vulnerability issue that was picked up. Leadership motivated as to why urgent development was required and it was prioritised as high and other items were placed on hold.” – P1

“We’ve has instances where vulnerabilities were detected, and we had to drop everything and resolve it.” -P5

Participants were asked whether leadership understands security's impact on product development timelines and the consensus amongst half of the participants was that this recognition allowed teams to integrate necessary security measures without undue pressure to compromise on quality.

“Yes and based on the emphasis that is placed on security. So, they would understand if something is delayed by a few days due to a security fix.” – P5

Participant 2 also noted that leadership ensured adequate technical support for security requirements by ensuring that architects were part of the design phase, and that security was considered right from the start. This was echoed by Participant 4 and Participant 8.

“Most of the security threats are handled within the team. But if its beyond our technical knowledge or control, top management and leadership are always willing to assist or get assistance from an external security expert.” – P4

4.4.2.2 Encouragement of Personal Development and Self-Learning

Participants were asked if they received any comprehensive security training and how they kept up to date with the latest cyber threats.

“We currently have three, and one is specifically aimed at developers and secure coding. It is very similar to the way you would learn on Udemy, except it is focused on the different types of security within the SDLC.” – P11

From a leadership perspective, a cultural norm that was recurring is continuous learning is promoted as indicated by Participant 11.

“A cultural norm that is encouraged is continuous learning. We offer all our staff members Udemy Business which gives them access to all the top-rated courses, and this is paid for by the company. They can use that to learn everything and anything, but there are also specific tracks that cover your area of expertise.” – P11

The participants in technical roles also noted that comprehensive training is offered but not mandatory. Specialised training was also offered to the development team leads. All of the participants indicated that they did additional online learning on their own time and the cybersecurity training offered by the organisation. Self-learning, accountability, and individual efficacy were recurring themes.

“Keeping up to date, requires continuously learning, so taking online courses. There are also architecture forums within the organisation that takes place where there is knowledge sharing.” – P4

“In addition to that, the self learning, as an individual I use to increase my knowledge on the security perspective.” - P5

Half of the participants mentioned that cross discipline collaboration was useful in acquiring knowledge related to security. Architects, software engineers, DevOps and QA engineers indicated knowledge sharing and collaborations on design and testing can lead to better design decisions.

“Collaboration with the developer and discussing features that they are building can also lead to questions about whether they have considered aspects that could impact security.” – P3

4.5 Findings of Proposition 3

To answer RQ 3, the participants were asked about the type of tools and technology being utilised at the organisation, the selection process and how effective they are in enhancing agility and security practices.

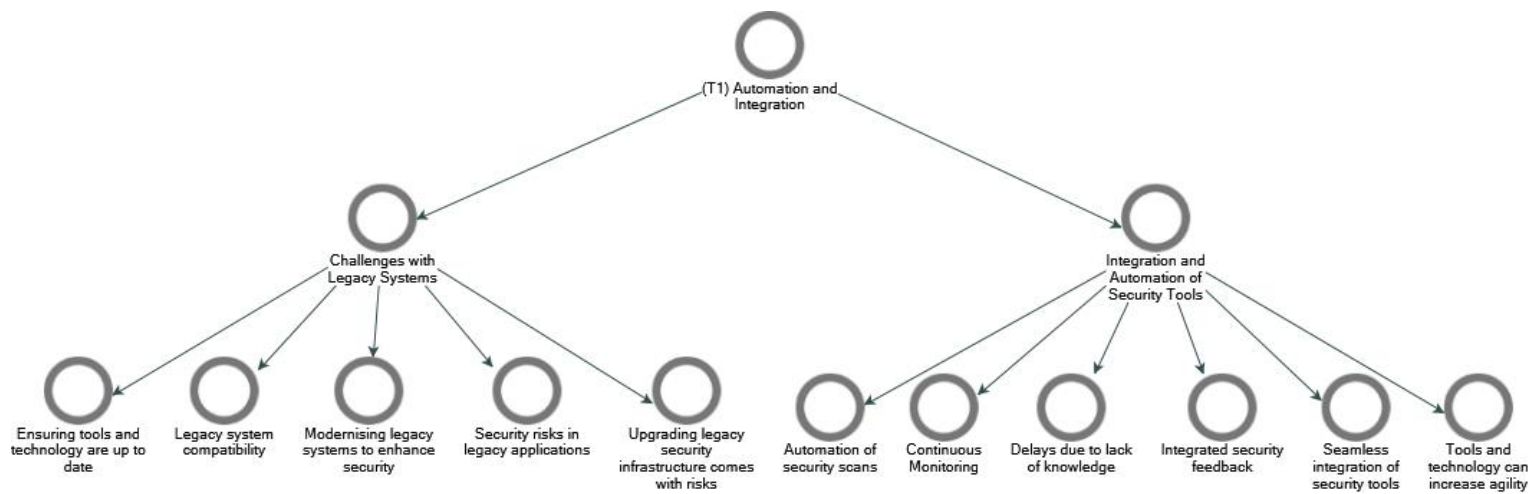


Figure 14. Theme, category and code hierarchy - Automation and Integration

4.5.1.1 Integration and Automation of Security Tools

Participants were asked how the use of, or lack of technology, affected the agility of the team in terms of delivery of working software and security.

More than half of the participants indicated that the use of tools integrated into their build pipelines had increased the efficiency and agility of their teams. Participant 8 indicated that using security tools within the CI/CD pipeline had increased their agility. This was also echoed by Participant 11.

“I think the tools integrated into our process make it much more efficient and effective because there are certain things that the tool can do very quickly. Sometimes, you need more than one tool to assist you, which is why some use online tools to help.” – P11

In contrast, a few participants indicated that tools can sometimes hinder the delivery of working software due to a lack of developer knowledge and turnaround time to resolve issues. Participant 1, Participant 3 and Participant 6 acknowledged the long turnaround time with the use of SAST tools hindering the delivery.

“The scanning tools that are used can take a really long time to run. And can cause delays of up to a day if any security issues are detected.” – P3

“In terms of operations, the security technology and policies have hindered operations and negatively impacted SLAs.” – P6

Majority of the participants mentioned the automation of security scans within their CI/CD pipelines. This also allowed for continuous monitoring of security vulnerabilities.

“There are tools that can improve our confidence in terms of security. Autonomous deployments for example takes the risk out what goes live. When human error is removed, we can improve the security.” – P9

Participants also mentioned that the tools were integrated seamlessly into their existing processes, with integrated and detailed feedback provided to the developers on how to resolve issues.

“Integration into the pipeline has been done effectively. There is no manual work required. There were no challenges encountered, as the tools were very simple to integrate and use.” – P1

“The tools we have are quite effective, as they are able to detect vulnerabilities and provide recommendations.” – P5

“The report gives a detailed explanation of what is wrong with the code and how you can improve it, including security.” – P8

4.5.1.2 Challenges with Legacy Systems

The participants were asked what challenges were encountered with the tools and technologies used for secure development practices.

A recurring theme among more than half of the participants was the challenges experienced with security and legacy applications. Participant 4 and Participant 10 indicated that new features can take a long time to develop on legacy applications. Legacy applications are often more susceptible to attack due to outdated packages and libraries. However, upgrading can be challenging as it could further delay delivery.

“Many older libraries are being exploited for security. So, it is becoming more important to try and upgrade things to the latest version, if only to gain those security patches. It's very difficult to upgrade without causing issues with the developers' working environment.” – P10

“On the legacy applications, the security team keeps sending concerns from the scans. So, we have to keep managing those threats that keep coming up in the legacy apps. If we rewrite it with the latest technologies, we can get rid of most of those vulnerabilities.” – P4

Participant 9 and Participant 7 agreed with this view and indicated that the use of modern technologies, such as cloud technology, had the advantage of security.

“We have recently adopted cloud technology, and there are many tools that provide security built in the cloud. As well as the interaction between our API’s and external entities. I think that investment in cloud infrastructure and the tools that we use to interact in that environment also adds a lot to the security infrastructure.” – P9

4.5.2 Tools and Technology

A graphical representation of the theme, category and code hierarchy was generated from NVivo and presented below in Figure 15. The broad theme of Automation and Integration has two categories, namely, Integration and Automation of Security Tools and Challenges with Legacy Systems, for which the findings are presented.

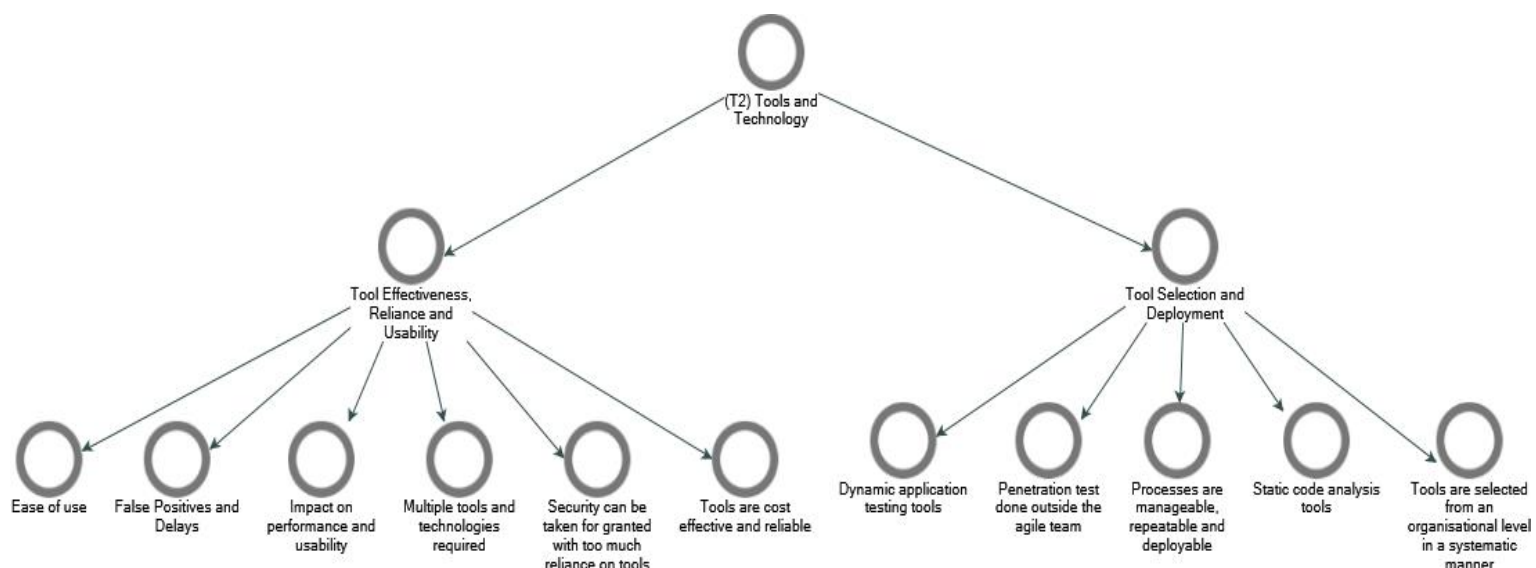


Figure 15. Theme, category and code hierarchy - Tools and Technology

4.5.2.1 Tool Effectiveness, Reliance and Usability

Participants were asked how they measure the effectiveness of security tools. Participant 11 noting that a drop in detected vulnerabilities was used to measure the tools effectiveness.

“If we take SonarQube, we look at the current vulnerabilities versus the ones that appeared after we implemented the tool and resolved all the detected vulnerabilities. Over a certain time period, there was a drop in the critical, high, and medium vulnerabilities, so they actually started appearing less.” – P11

Participants in a technical role who used the security tools in their everyday roles indicated that the tools are easy to use.

“For the developers, the tools that are in place are quite simple to use and understand.” – P2

False positives affected the reliability of the tools in place and the delivery timeline, as indicated by Participants 2 and 3. A few participants mentioned that security tools and technologies added to applications can cause usability issues.

“The security initiatives have negatively impacted external vendors as well. Also, from an integration point of view with external vendors, we cannot confirm that they have received the information we want to send them. Access control has become a problem, although from a security point of view, all parties need to protect their data.” – P6

Multiple tools and technologies are required to cover the different types of applications. No one tool can cover Java applications, front-end, and mobile apps.

“There is no one tool that we can implement on different types of applications, such as backend, front end, and mobile apps, and that is the challenge.” – P4

All participants agreed that security requires continuous monitoring and is a task never stops. Participant 3 and Participant 8 felt that with an overreliance on tools security can be taken for granted.

“If we want to take these things seriously, you need dedicated experts that can do it for you and automation processes that can make it be done quickly. But even with an automated process, you have to be careful not to have a warm feeling.” – P3

“The organisation feels it relatively safe, as they’ve complied with the framework that is in place. The frameworks and policies in place give the impression of a secure application, which can be taken for granted.” – P8

4.5.2.2 Tool Selection and Deployment

Participants were asked what tools and technologies their team currently use to support secure development practices. (SAST, DAST, threat modelling, etc.) and how were they chosen and implemented.

SAST and DAST tools were the most widely used and integrated directed into the pipeline. These tools were able to provide reports with security vulnerabilities and recommendations on how to fix them. SonarQube was frequently mentioned by the participants to perform code analysis.

“For static code analysis SonarQube was implemented recently. Integrated within the pipeline and all code is scanned before being deployed to production. This also gives a good indication of the quality of the code.” – P2

All participants indicated that their organisations followed a procedure for the selection of tools and technologies. In some cases, the individual teams performed a proof of concept on the tools they researched, which could increase the security posture of the applications.

“We follow a process in terms of how we choose a product. We look at what’s out there in the market and we look at what is in the top quadrant of products. We then choose the top 3 based on our requirement and then we contact the vendor of the product.” – P11

Participant 1 explained organisational oversight, where this centralised approach ensured consistency and compliance with organisational standards.

“Within the organisation, there is a group that makes decisions on which tools and technology can be used. The group meets monthly to discuss these topics. Individual teams are only allowed to use technology that has been approved by this committee.” – P1

Penetration tests were often performed outside the agile teams by either a third party or the security team within the organisation. Four participants noted that when these

tests are performed by a third party or outside of the team, they must be planned in advance to avoid any delivery delays

“Pen Tests are also done by a third-party company. This can become very expensive and can also take time before they are able to complete it, so it needs to be planned in advance.” – P2

4.6 Summary of Findings

The integration of SDPs within agile teams in the South African financial services industry is shaped by organisational culture and structure, leadership support and commitment, and the utilisation of tools and technology. A strong security culture characterised by widespread attention to security issues and mandatory training that fostered a security-conscious mindset among team members was observed from the findings.

Environmental factors such as the competitive nature of the financial services industry and the quickly evolving threat landscape impact the cultural norms and policies within the organisation. Challenges arose from inconsistencies in training, communication gaps, and the prioritisation of feature delivery over security. Additionally, the lack of security champions and the presence of siloed team structures hindered effective integration.

Leadership played a pivotal role in this process where proactive leaders promoted better integration of SDPs by fostering accountability, allocating resources effectively, and empowering team members. In contrast, reactive leadership undermined the effectiveness of security measures.

The integration and automation of security tools enhanced efficiency and enabled continuous monitoring of software applications. However, challenges such as legacy systems, concerns regarding tool effectiveness, usability issues, and an overreliance on technology hindered the successful integration of SDPs.

CHAPTER 5. DISCUSSION OF FINDINGS

5.1 Introduction

Chapter 5 presents a detailed discussion of the study's qualitative findings, examining how organisational structure and culture, leadership support and commitment, and the deployment of security tools and technologies influenced the effective integration of SDPs within agile teams. This chapter contextualises these findings by comparing them to existing literature, highlighting key similarities and differences from prior research in the field of secure software development.

An aspect of this discussion is the validation of findings against the OWASP SAMM framework. By mapping the study's insights to the three maturity levels of SAMM. This approach ensures that the study's conclusions align with industry best practices and contribute to a more structured, security-conscious SDLC.

5.2 Discussion of Proposition 1

The effective integration of SDPs within agile teams is influenced by organisational structure and culture. The findings from this qualitative study highlight three themes: Organisational Culture, Organisational Structure, and Environmental Factors. These themes align with the findings from recent academic research.

Proposition 1: Organisational structure and culture are vital to creating a security-conscious mindset within agile teams. Within the organisational structure, the security champion role can effectively bridge the gap between security and software development teams. Cultural factors such as awareness, training, and cross-team collaboration can create a security culture, leading to the effective integration of SDPs.

5.2.1 Organisational Culture

5.2.1.1 Training and Awareness

The participants in the study indicated that the security culture within their organisation was highly prioritised, with attention given to creating and maintaining a security-conscious mindset at all levels of the organisation. The onset of remote work, expanding digital services, and evolving cyber threats reiterated the importance of security at a strategic level in South African FSPs.

Mandatory cybersecurity training was offered to all participants. However, specialised and comprehensive software security training was offered to some participants but not made compulsory, creating a critical gap for technical teams who require more advanced competencies.

These findings are consistent with those of Ergen et al. (2021) who argue that a robust security culture is fostered by both awareness initiatives and specialised training tailored to employees' specific roles and responsibilities. Similarly, Janisar et al. (2024) emphasised the importance of security awareness as a contributing factor in the successful implementation of security requirements engineering.

Hence, while a broad awareness of security principles enhances day-to-day secure behaviours, the lack of in-depth, role-specific training poses challenges to effectively integrating SDPs. This gap aligns with findings from Tejay and Mohammed (2023), who reported that heightened awareness significantly improves security culture, yet may not suffice if technical teams lack the specialised knowledge to address nuanced security concerns. This can lead to long-term risks as technical debt increases due to inefficient remediation efforts. Although the participants in the study have a good understanding of the importance of security practices, it is observed that a lack of specialised knowledge is hindering the effectiveness.

According to SAMM, training is critical to enhancing a team's maturity in security practices. Under the *Governance* pillar, the *Education and Guidance* practice of SAMM recommends offering ongoing, relevant training to ensure all team members understand and can effectively remediate security implications. Since security threats evolve rapidly, ongoing education is required.

5.2.1.2 Prioritisation of Feature Delivery over Security

A recurring cultural norm was the prioritisation of feature delivery over security requirements. This was attributed to market pressures and the agile mindset focused on rapid delivery. This pattern showed the contrast between rapid release cycles and the thorough, time-consuming process of implementing robust security controls. According to Weir et al. (2023) feature priority and a lack of security resources contributed to security being overlooked and deprioritised in agile teams.

Participants indicated that high workloads and business demands often led to security being overlooked or addressed only after product release. Although, it was noted that

high-ranking threats always took the highest priority, as per the organisational policy. However, it appeared that medium to low-ranking security tickets often went to the requirements backlog to be addressed later. While this works in the short term, this practice can accumulate technical debt and increase long-term risk, as vulnerabilities might remain unaddressed until they escalate in severity.

As Participant 1 also noted that on an organisational level, there is a great emphasis on security. However, it doesn't always filter down in the individual teams. This is in line with a study by Ryan et al. (2023) in which they concluded that a security-conscious culture doesn't always translate into secure coding practices. Although organisational culture values security, teams under tight deadlines may deprioritise it in day-to-day tasks to meet deadlines. This misalignment could undermine the organisational objectives for stringent cybersecurity values.

Studies have been conducted to recommend frameworks for agile teams to integrate security into their agile process. However, few studies have empirical data, and with the business demands, it is not always practical for development teams to adapt to a new way of working or different frameworks (Ardo et al., 2022; Moyon et al., 2020). This finding correlates with the challenges identified in agile environments, where speed is often prioritised over thorough security practices. Studies focussed on security requirements highlighted that in agile methodologies, the emphasis on quick iterations can lead to insufficient attention to security, as teams may perceive security tasks as impediments to rapid delivery (Mihelič et al., 2023; Tøndel et al., 2022).

SAMM addresses this challenge through its *Design* and *Implementation* pillars, which advocate for integrating security requirements and controls early in the SDLC. By embedding security considerations from the outset, organisations can mitigate the tension between rapid feature delivery and security compliance.

5.2.2 Organisational Structure

5.2.2.1 Organisational Policies and Security Frameworks

Participants reported that organisations had established security policies and mandatory practices for software development and testing. Rigorous testing policies, code reviews and vulnerability scanning were standard procedures. This indicated a top-down structural effort to create a security-conscious working environment. Some

of these policies and procedures were ingrained in the participants' daily activities, indicating a positive influence on the integration of SDPs.

These findings align with best practices outlined in recent literature. Ryan et al. (2023) and Tolah et al. (2021) emphasised the importance of formalised security policies and standards in guiding teams toward consistent security practices. Their research indicated that clear policies provided a foundation for implementing SDPs effectively.

However, challenges arose regarding the accessibility and communication of these policies. Some participants felt that documentation was not “*well-advertised*” or easily accessible, hindering their ability to comply fully with security standards.

Effective communication of policies is critical, as highlighted by Nurgalieva et al. (2023), who found that the success of security initiatives depended not only on the existence of policies but also on how well they are communicated and understood by team members. They advocate for regular training and easy access to documentation to ensure compliance. When documentation is not easily accessible, teams may inadvertently deviate from best practices.

SAMM’s *Governance* pillar underscores the necessity of clear policy distribution and continuous improvement processes. Ensuring that all team members know and understand security policies is essential for effective implementation of these standards.

5.2.2.2 Organisational Communication

In the South African context, organisational hierarchies are influenced by traditional structures and are still prevalent in FSPs. All participants indicated that there was a separate security team within the organisation. This separation resulted in siloed teams and hierarchical communication structures. Since access to the security team and communication often went through multiple layers, this resulted in inefficiencies. Although some of the participants did not feel that the way of communication was counterproductive, others noted the inefficiencies in how reported issues were remediated.

This siloed structure negatively influenced the integration of SDP’s. While the organisational level policies may exist, their efficacy depends on effective communication. Software engineers struggled to interpret policies, clarify

misunderstandings, or proactively address emerging threats on their own. As a result, security inadvertently became a reactive process, with the security team only communicating when issues arise rather than working as a continuous collaborator in the software development lifecycle.

Tøndel et al. (2022) and Appiah et al. (2022) underscored the importance of collaboration between different teams to enhance the effective integration of SDPs, which aligns with the sentiments of the participants. Teams must collaborate to scan for threats, coordinate responses, and refine procedures post-incident to continuously improve. Janisar et al. (2024) argued that ongoing communication between development teams and security experts was essential to enhance understanding of security requirements. It was also noted that effective communication and well-defined policies positively influenced the security mindset of software developers (Arizon-Peretz et al., 2021).

With regards to SAMM, under the *Governance* pillar, *Education and Guidance* suggests that all team members are aware of the organisational policies in place regarding security.

5.2.2.3 Team Structures

There was a noticeable lack of a designated security champion within the agile team. Although participants acknowledged the potential advantages of such a role, none were formally established. Given that traditional roles and structures in these large organisations are already entrenched, creating new positions involves a bureaucratic process. Participant 11 noted that a shortage of cybersecurity expertise was a contributing factor.

Accountability for security was currently shared across the entire agile team, and this made it easy for team members to overlook security during the SDLC, due to other pressures. When there is a dedicated person who is accountable for security, this can be an effective way to ingrain SDPs into the agile processes and not regard security as an afterthought.

In contrast to the findings, numerous studies indicated that this role is crucial to ensuring that security is thought of at every stage of the SDLC. Integrating security champions within agile teams improves communication and facilitates the effective integration of SDP's. Security champions liaise between development and security

teams, promoting a security-conscious mindset and providing immediate support for security-related queries (Lende et al., 2023; Nurgalieva et al., 2023; Schubert et al., 2023). The absence of designated security champions indicates a missed opportunity to enhance collaboration and implement security controls more effectively (Jaatun & Cruzes, 2021).

SAMM highlights the value of involving security personnel within development teams and encourages the establishment of roles like security champions to bridge communication gaps. This is a part of the *Governance* pillar within *Education and Guidance* in which assigning a Security Champion to each development team results in an increase in the maturity level.

5.2.3 Environmental Factors

5.2.3.1 Industry Requirements

External factors significantly influenced the adoption of security requirements among financial service providers (FSPs). Participants highlighted that evolving industry regulations and data privacy policies necessitated strict adherence, thereby shaping organisational structure and culture. Operating within a framework of stringent regulations, industry standards, and market demands, FSPs faced unique challenges.

In the South African financial services industry, competitive pressures, and the imperative to innovate rapidly to meet customer expectations deepen these challenges. Banks and insurance companies are compelled to digitise services and enhance customer experiences, often leading to accelerated development cycles. This rapid pace sometimes results in security considerations being overlooked by the urgency of feature delivery (Bisschoff & Els, 2023; FSCA, 2022).

The regulatory landscape in South Africa added another layer of complexity. FSPs are required to comply with stringent regulations such as POPIA and FICA, which mandated robust data protection measures. These regulations necessitated the implementation of comprehensive security policies to safeguard sensitive customer information (Onubedo, 2023).

Comparatively, global financial institutions also faced challenges with rapid innovation and stringent regulatory compliance pressures. A study by McKinsey & Company found that 70% of financial institutions reported that increased compliance with

regulations prompted the maturation of their cybersecurity capabilities. This underscores the global impact of regulatory frameworks on security practices within the financial sector (Atkins et al., 2024).

5.2.3.2 Security Threat Landscape

Security is seen as an ongoing task rather than something that is only considered once. Agile teams recognised that achieving 100% security was near impossible due to the evolving nature of the threat landscape. The response of "*security is never-ending*" reflected the necessity of integrating SDPs as a continuous process which is supported by regular updates, reviews, and proactive measures.

The rapid evolution of technology introduces new security challenges and threats, as evidenced by the consistent need to address vulnerabilities in new and existing systems. This was particularly evident in cloud environments and legacy systems, where outdated packages were vulnerable to exploitation. Continuous updates and training are critical to keep pace with these emerging threats.

Proactive organisations ensured that training, regular updates to policies and procedures were made ensuring that security processes evolve alongside technological changes and security threats. Factors such as limited budgets, siloed communication, or resistance to change, made it difficult to address new threats effectively. The impact of the evolving security landscape impacted the selection and use of tools and technologies as well the organisation's response to updated training and awareness programs.

Appiah et al. (2022) found similar results in their study, arguing that security should be part of a resilience framework where organisations continuously scan for threats, eliminate risks, and upgrade their software systems. These were not isolated responses due to an active threat but part of an ongoing cycle of improvement to address the evolving nature of cyber threats. Security in software applications requires a lifelong effort (Janisar et al., 2024). To ensure that leadership and decision-makers are aware of and prepared for cyber threats, Parkin et al. (2023) suggested the use of recurring scenario-based training and workshops.

Under the *Design* pillar in SAMM, it is suggested that processes such as *Threat Assessment* and *Secure Architecture* be standardised, continuously reviewed, and

improved for adoption and appropriateness, thus resulting in increased security maturity.

5.3 Discussion of Proposition 2

Leadership support played a pivotal role in the successful integration of SDPs within agile teams as they influenced employee behaviour and support the security-conscious culture. The findings for this proposition identified two key themes: Leadership Commitment and Leadership Support.

Proposition 2: Leadership commitment and support are driving factors in fostering a security-conscious environment, where agile teams are empowered and possess the necessary resources, skills, and knowledge to integrate security into their processes and, therefore, increase the software's security posture.

5.3.1 Leadership Commitment

5.3.1.1 Reactive vs Proactive Leadership Approach

The findings revealed two different leadership approaches to security. Some leaders displayed a reactive stance, addressing security concerns only after incidents occur or by the presence of external pressures. Participant 1 noted that leadership allocated time for security issues only when upper management raised a problem. Some participants noted that factors influencing the leadership approach included a lack of awareness and understanding of security or business pressures, resulting in a reactive approach.

Conversely, other leaders demonstrated a proactive approach, prioritising security from the outset and holding themselves accountable for the application's security. Participant 3 highlighted that leadership provided a high level of support due to accountability and the pressure they faced regarding security outcomes.

These findings suggest that when security is not included in leadership's responsibilities, it may be neglected and viewed as a lower priority. This likely stems from the need to focus more on feature delivery than on prioritising security requirements due to the competitive industry environment.

The findings clearly show that leadership's stance, whether reactive or proactive, influenced the degree to which SDPs were integrated into the agile process. Reactive

leadership approaches resulted in event-driven security efforts with only short-term remediation. On the other hand, proactive leadership fostered a culture of security-conscious mindset, where the integration SDPs became ingrained in the daily activities of the agile team and their processes.

This study's findings are consistent with the work of Arizon-Peretz et al. (2022) which determined that when the priority of SDPs is not clearly communicated and prioritised by management, agile teams are less likely to prioritise these requirements and will only address them in response to an incident.

Recent studies on building resilience and proactively mitigating cyber threats indicate that a security strategy is required to effectively address any potential threats (Appiah et al., 2022; Ofoegbu et al., 2024). This would require a commitment from leadership to deliver on this strategy.

Under the *Operations* pillar in the SAMM model, it is suggested to proactively manage and continuously monitor the incidents, environment and operational management aspects. This would require a leadership to make these areas a priority and actively provide support and resources required to do this effectively.

5.3.1.2 Resource Allocation and Investment

All participants acknowledged the necessity of adequate time, resources, and training to implement SDPs. However, the degree to which leaders allocated these resources varied widely. Participants spoke of the difficulty of balancing security-related tasks, such as vulnerability remediation, against other urgent deliverables. Without explicit leadership support in the form of dedicated time, skilled personnel, or training budgets, security tasks often fell low on the priority list. Participants also highlighted the scarcity of cybersecurity talent and the associated costs. Participant 11 indicated that hiring experienced cybersecurity staff was expensive, and junior hires required time to upskill.

Leadership should show a commitment to security by providing adequate resources such as budget, time, training, and the appointment of security experts within the team to emphasise its importance (Ardo et al., 2022; Gutfleisch et al., 2021). Herath et al's. (2020) study similarly highlights the pivotal role of leadership support in allocating resources and prioritising security initiatives.

Without dedicated training opportunities, time, and appropriate staffing levels, agile teams struggled to consistently integrate SDPs into their processes. Organisational resources and capabilities are central to the successful adoption of new practices. Sufficient resource allocation, such as training, hiring skilled personnel and providing an adequate budget, is an essential organisational enabler. If leaders fail to provide these resources, agile teams cannot move beyond basic compliance and towards more advanced, integrated security practices (Umeugo, 2023).

Under its *Governance* pillar, SAMM underscores the importance of capacity building from both a skills and infrastructure perspective. Effective governance and strategy must be supported by a commitment to invest in talent, training, and tooling. Resource allocation is thus not an optional extra but a core component of moving up the maturity levels. Without it, even well-defined policies and leadership statements don't result in improved security posture.

5.3.2 Leadership Support

5.3.2.1 Leadership Support for Security Initiatives

Participants widely acknowledged that they felt empowered to raise security concerns with leadership. This openness suggested a foundational level of trust and communication from leaders down to team members. Instances where urgent vulnerabilities arose, saw leadership quickly intervening, prioritising security fixes over other tasks. These findings suggested that leaders recognise the importance of security, especially when facing potentially severe repercussions.

Moreover, about half of the participants confirmed that leadership understood the effect of security requirements and product timelines. When leadership understood how security tasks might temporarily slow down feature delivery, teams felt less pressure to compromise on quality. Additionally, participants noted that leadership ensured technical expertise was accessible when required.

The finding of this study closely aligns with Sas et al. (2021) who argue that leadership's understanding and communication of security risks can shape employees' perceptions and attitudes towards security. When management visibly supported security initiatives, it reinforced the importance of security to employees, encouraging them to adopt secure behaviours in their daily activities (Ryan et al., 2023; Schubert et al., 2023).

Under the *Governance* pillar, SAMM indicates that mature security integration requires governance and support at the highest levels. Leaders who champion openness, respond promptly to vulnerabilities, and value expert involvement progress toward higher security maturity levels. By facilitating access to architectural and security experts, leadership complies with SAMM's emphasis on ensuring that security is integrated at every stage of the SDLC.

5.3.2.2 Encouragement of Personal Development

A facet of leadership support is how leaders promoted continuous learning and skill development. These opportunities allowed software engineers, QAs, and architects to broaden their security knowledge, stay updated with evolving threats, and refine their secure coding skills. By investing in learning platforms, leadership sends a message that security proficiency is a valued asset, not just an individual responsibility.

Participants also noted that while training opportunities existed, they were often not mandatory. Instead, self-learning and personal accountability remained key. This implied a leadership style that trusts and empowers technical staff to shape their own professional growth. While mandatory security training could strengthen baseline competencies, the current approach fostered personal motivation, encouraging team members to take ownership of their skill development. Although, to effectively integrate SDPs, technical personnel required specialised training, which leadership should support.

Cross-functional collaboration emerged as an effective means of gaining knowledge. Participant 3 and others mentioned that discussions between architects, software engineers, DevOps, and QA engineers on design decisions often surfaced potential vulnerabilities. Leadership's role in enabling this collaboration through structured forums, architecture reviews, or inter-team workshops reinforced the importance of shared security ownership.

Arizon-Peretz et al (2022) investigated the personal factors influencing security by design and found that a security culture influenced security self-efficacy. This aligns with the results of the study in which the participants exhibited a motivation to upskill themselves and integrate security into their activities where resource permitting. Zakaria et al. (2007) very early indicated that leaders can act as role models, encouraging employees to engage in self-learning activities. Establishing a strong

security culture involved setting security as a priority at the leadership level, which can motivate employees to learn and apply security practices independently (Sas et al., 2021).

SAMM highlights that as organisations mature, their focus shifts from basic awareness to specialised skill-building under *Education and Guidance* in the *Governance* pillar. Leadership's promotion of personal development aligns with SAMM's guidance that advanced training, role-based education, and knowledge-sharing forums are indicative of higher security maturity level. Leadership's encouragement of continuous learning sets the foundation for scaling beyond compliance-oriented security tasks and toward secure-by-design development practices.

5.4 Discussion of Proposition 3

Tools and technology support the successful integration of SDPs as they have the potential to increase the agility and security of software applications. The findings for this proposition identified two key themes: Automation and Integration and Tools and Technology.

Proposition 3: Tools and technology can help agile teams successfully integrate SDPs into their existing processes by implementing security practices within their CI/CD pipelines. Factors such as organisational readiness, compatibility, budget, and technical skills play a role in the selection and adoption of tools.

5.4.1 Automation and Integration

5.4.1.1 Integration and Automation of Security Tools

Most of the participants noted that integrating security tools directly into the CI/CD pipeline increased their agility. While some noted that the time taken for the scans to complete can hinder the time to delivery, multiple participants emphasised that well-integrated tools streamlined workflows by providing immediate and actionable recommendations.

However, some participants encountered challenges with security tools. Some scanning tools introduced delays when detecting vulnerabilities that took a significant time to fix, and in some cases impacting Service Level Agreements (SLAs). While automation can expedite many tasks, inefficient or resource-intensive tools can

momentarily hinder agility when teams lack the expertise to interpret or quickly resolve reported issues.

Continuous monitoring and automated checks helped maintain a stable security baseline even as rapid iteration continues. Security was embedded in the development cycle rather than applied after. However, insufficient expertise on tools and technology can slowed productivity. Thus, the influence of technology on SDP integration is reliant on both the quality of the tools and the team's readiness to use them effectively.

These findings align with studies highlighting that automation and integration of tools in CI/CD pipelines lead to increased agility and efficiency. The studies also indicated that dynamic security testing tools in CI/CD pipelines highlighted how automation ensured fast, scalable, and effective security tests (Marandi et al., 2023; Rangnau et al., 2020; Umeugo, 2023).

The successful integration of automated security tools demonstrated how technological capability interacted with organisational processes and the external environment. When teams have the organisational support, training, and resources to leverage these tools effectively, technology adoption positively influences the integration of SDPs (Palombo et al., 2020).

SAMM emphasises continuous improvement and embedding security at every stage of the SDLC. Automated security scanning and integrated tooling align with SAMM's guidance on *Verification* and *Implementation* maturity. As organisations mature, they rely on automated controls to maintain the security posture, ensuring that vulnerabilities are consistently detected and addressed. The participants' accounts reflect a journey towards higher maturity levels, as reliance on integrated, automated tools becomes normalised.

5.4.1.2 Challenges with Legacy Systems

Legacy systems emerged again as a persistent challenge. Participants described how outdated frameworks, libraries, and technologies complicated security efforts. These struggles highlighted a gap between modern security tools which are designed for modern architectures, microservices, and cloud environments instead of legacy applications which are usually monolithic architectures.

While modern tools detected vulnerabilities, applying their recommended fixes wasn't always straightforward in older systems. Balancing upgrades against delivery deadlines created friction, placing agile teams in a dilemma. They would need to invest in modernisation now to prevent escalating technical debt or accept short-term inefficiencies. Tool adoption alone did not ensure the successful integration of SDPs when the underlying technological environment lagged. Thus, legacy systems were not just a technical burden, but they impeded the full realisation of technology-driven security practices.

Marandi et al.'s (2023) findings align with the participants responses in that the incompatibility with older systems hindered the integration of security tools. The difficulty in updating legacy systems and the inherent risks they posed were highlighted in the study. This was corroborated by findings in other studies, which noted that vulnerabilities in outdated systems often led to delays in implementation and increased risks (Angermeir et al., 2021).

The TOE framework highlighted those environmental factors such as industry standards and competitiveness pressured organisations to adopt newer technologies. Legacy systems represented organisational inertia that must be overcome. Without organisational willingness, resource allocation, and a strategic vision to modernise, the successful embedding of SDPs through automated tools remains incomplete.

Within the *Operations* pillar in SAMM, achieving higher maturity often implies modernising not just the processes but also the technology stack to ensure that security controls are effectively implemented. By acknowledging the challenges legacy systems impose, organisations can prioritise phased modernisation, aligning their technological foundations with the security maturity path.

5.4.2 Tools and Technology

5.4.2.1 Tool Effectiveness, Reliance and Usability

Participants indicated that assessing the effectiveness of security tools often involved tracking changes with vulnerability metrics over time. Participant 11 mentioned a drop in critical, high, and medium vulnerabilities detected by tools like SonarQube as a clear indicator of tool effectiveness. This quantitative approach made it possible to gauge whether a tool detects issues and whether those issues were being successfully remediated and prevented from recurring.

Another critical aspect emerged as ease of use. Participants in technical roles generally found tools straightforward, with practical recommendations, improving the developers' ability to remediate vulnerabilities quickly and confidently. Such user-friendliness helped integrate SDPs more effectively into developers' workflows and work estimations.

Some participants mentioned false positives and erroneous vulnerability detection as weakening the tool reliability and potentially slowing delivery. Usability issues extended beyond the development team as the integration of security tools within the software applications caused usability and performance issues for the users. This finding aligns with the study of Gutfleisch et al. (2021) in which they emphasise the need for creating awareness in users regarding security features.

The diversity of technology stacks also complicated tool selection. Participant 4 emphasised the lack of a single tool covering backend, frontend, and mobile applications, necessitating multiple specialised solutions. This diversity increased overhead in learning, maintenance, and integration efforts, making adopting a uniform approach to SDPs more difficult.

A subtle theme was the notion of "*security tool complacency*". Participants 3 and 8 cautioned that reliance on tools could create a false sense of security. This view was also supported by the study of Shostack (2022) and Ryan et al. (2023). When organisations uncritically trust tools and compliance frameworks, they risk ignoring deeper architectural vulnerabilities. Tools are enablers and overreliance can lead to complacency and reduced vigilance. Pan et al. (2024) similarly argue that security threats also can exist in CI/CD pipelines and must be properly understood to ensure that your technology is not vulnerable.

From the TOE perspective, technology adoption of security tools interacted with organisational factors such as policies for tool selection, training and environmental pressures such as regulatory requirements. Tools that aligned well with developers' skill levels and project requirements enhanced the successful integration of SDPs.

SAMM recommends continuous improvement and contextual adaptation of security activities. The presence of false positives, usability constraints, and tool fragmentation suggests that organisations may need to mature their *Verification* and *Governance* pillars. Adjusting processes to handle false positives, selecting tools for specific

technology stacks, and balancing automation with human expertise align with SAMM's guidance for evolving security maturity.

5.4.2.2 Tool Selection and Deployment

When participants were asked about the tools and technologies their team currently uses, SAST and DAST tools emerged as the most commonly integrated solutions. Tools like SonarQube were frequently mentioned, reflecting their popularity for code quality and security scanning. Implemented directly into CI/CD pipelines, these tools generated detailed reports and recommendations, contributing to a proactive and systematic security culture. No participant mentioned the use of threat modelling in their processes, although this is a recommended step in the *Design* pillar in SAMM.

Studies by Otieno et al. (2023) and Weir et al. (2021) similarly found that static and dynamic code analysis and vulnerability scanning are effective ways of ensuring security in software applications, especially when integrated into CI/CD pipelines.

According to all participants, the selection of tools followed a procedural approach. Organisations regularly evaluated market-leading solutions and conducted proof-of-concepts (POCs) to ensure that the chosen tool fits their technical and business needs. By involving technical teams in the tool vetting process, organisations leveraged practical insights to select solutions that were not only effective but also easy to integrate.

Centralised oversight and technical committees, ensured consistency and compliance across the whole organisation, aligning with the findings of Rangnau et al. (2020). While streamlined decision-making and enforced organisational standards, it limited teams' autonomy to experiment with emerging tools. The reliance on third-party penetration testers, noted by Participant 2, further shows that while some testing activities were internalised, others remain external, requiring advanced planning to avoid delivery delays.

The structured approach to tool selection and deployment demonstrated that organisations recognised the complexity of choosing and integrating security technologies. This careful selection process ensured that adopted tools enhance the security rather than adding unnecessary overhead.

As Umeugo's (2023) quantitative study suggested that technical teams should prioritise tools that offer clear benefits, integrate seamlessly with existing processes, and are easy to evaluate their effectiveness in increasing the security posture. Another considerable factor mentioned by Shostack (2022) was cost, which aligns with findings of this study that limited budgets can influence the technology being selected.

The TOE framework identifies internal evaluation processes and environmental pressures such as industry standards, compliance and the evolving threat landscape as influencing tool decisions. By methodically selecting proven solutions, organisations reduced uncertainty and encouraged smoother technology adoption by all teams.

SAMM prescribes establishing guidelines, roles, and metrics around security activities under the *Verification* pillar. Carefully evaluating tools and integrating them into the CI/CD pipelines aligns with SAMM's emphasis on incorporating verification activities throughout the SDLC. This measured approach to tool selection can help organisations increase the maturity level, ensuring verification becomes a standard, ongoing practice rather than an ad hoc task.

5.5 Conclusion

Chapter 5 provided a discussion of the findings, emphasising how organisational factors, leadership, and technology collectively influence the integration of SDPs within agile teams. The study revealed that while security-conscious cultures are present at an organisational level, inconsistencies at the team level, driven by prioritisation of speed over security and hierarchical communication structures, hinder the effective integration of SDPs. An implication of these themes meant that security can be overlooked, deprioritised and lead to overall technical debt.

Leadership approach emerged as an influencing factor, with proactive leaders fostering accountability, resource allocation, and strategic focus, whereas reactive approaches often led to short-term fixes and overlooked long-term security goals. Security tools, particularly those integrated into CI/CD pipelines, significantly enhanced agility and vulnerability management, but their effectiveness was limited by legacy systems, over-reliance, and false positives. The discussion highlighted the

significant interplay between tool effectiveness, usability, and organisational practices within the context of security maturity as outlined by SAMM.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

This study set out to determine how organisational, leadership and technological factors influenced the effective integration of SDPs in agile teams. The findings highlighted that the effective integration of SDPs was impacted by the interplay between these factors. Organisational structure and culture, leadership commitment and support, and technological readiness influenced the success of these efforts, revealing that secure software development requires a comprehensive approach.

While tools and frameworks like OWASP SAMM provided valuable guidance, their effectiveness depended on how well they aligned with the unique needs and constraints of the organisation and its operating environment, such as addressing legacy systems, cross-team collaboration, and industry competitiveness. Additionally, this study highlighted the importance of growing an adaptive mindset where security is viewed as an evolving task rather than a static compliance requirement. By embedding security considerations into daily tasks, upskilling and investing in cross-functional expertise, organisations can transition from reactive to proactive security postures. This transformation strengthens resilience against evolving threats and positions teams to deliver innovative, secure solutions within competitive markets.

This chapter addresses the conclusions regarding each research question. It also discusses recommendations for stakeholders regarding the findings, the study's contributions and provides suggestions for future research.

6.2 Conclusion regarding Research Question 1

The aim of RQ1 was to investigate how organisational structure and cultural factors impacted the effective integration of SDPs within agile teams. This study revealed that

both organisational structure and culture played a fundamental role in how agile teams adopted and sustained SDPs over time.

Organisational structures with separate security teams led to hierarchical communication channels and created barriers to effective collaboration. These rigid structures contributed to slower feedback loops and inefficient remediation processes when security vulnerabilities were identified. The lack of a dedicated security champion within the agile team was also noted as a barrier to the efficient remediation of security issues. One key solution identified was the integration of security champions within agile teams. Security champions who act as liaisons between development and security teams will help to bridge the communication gap and ensure security is not an afterthought.

The effectiveness of well-defined security policies, minimum mandatory security practices, and structured risk assessments depended on how well they were implemented within agile workflows. The enforcement of security testing policies and the inclusion of security requirements in the SDLC ensured that security was not merely a compliance checkbox but an aspect of day-to-day activities.

From a cultural perspective, organisations strongly emphasised security, demonstrated by regular security awareness campaigns, mandatory training programs, and security initiatives. However, this security-conscious mindset was not consistently adopted across all teams. Some agile teams viewed security as a fundamental part of the SDLC, whereas others prioritised rapid feature delivery due to market pressures and business demands. This misalignment resulted in inconsistent security adoption and pockets of resistance to change.

The study also found that environmental factors such as regulatory pressures, stringent data protection laws, and high consumer trust requirements added complexity to SDP integration. Industry competitiveness within the South African financial services industry and the ever-evolving threat landscape influenced the competing priorities of rapid software delivery and strict security compliance, making it even more challenging to achieve a balance between security and speed.

Compared to existing literature, which primarily focused on generic security challenges in agile methodologies, this study provided deeper insights into the cultural and structural factors that influenced the integration of SDPs within South African FSPs.

The findings underscored the importance of formalised risk assessments, proactive security engagement at all levels, and the need for cross-functional collaboration between development, operations, and security teams. By addressing these structural and cultural challenges, organisations could strengthen their security posture while maintaining the agility necessary for competitive software delivery.

6.3 Conclusion regarding Research Question 2

The aim of RQ2 was to explore how leadership commitment and support influenced the effective integration of SDPs. The leadership approach emerged as a decisive factor in determining how effectively SDPs were integrated within agile teams. Proactive leaders demonstrated accountability for security outcomes by allocating resources, fostering open communication, and balancing the trade-offs between agility and security. This cultivated a culture where security was treated as an integral component rather than an afterthought. These leaders who actively supported security initiatives ensured security was embedded within the SDLC, and empowered teams to take ownership of security practices. Conversely, reactive leadership led to short-term remediation efforts, post-breach interventions, and a perception that security was a secondary concern rather than a core business priority.

The contrast between proactive and reactive leadership was evident in resource allocation, investment in security training and tooling, and the level of recognition given to security-related delays. Teams that operated under proactive leadership were more likely to integrate security seamlessly into their workflows, whereas those under reactive leadership often addressed security issues only when external pressures, such as compliance audits or security incidents, demanded immediate action.

This study also revealed that most participants felt empowered to raise security concerns within their teams. When leadership responded promptly to high-risk vulnerabilities and provided the necessary support, it reinforced the notion that security was non-negotiable. Leadership's willingness to prioritise security fixes and provide technical expertise played a crucial role in fostering a security-conscious culture. However, some teams still faced challenges where leadership support was inconsistent or where security initiatives lacked long-term commitment.

Leadership encouraged and supported continuous learning through personal development efforts, and employees were responsible for their own upskilling and professional development. This was enabled by the allocation of training resources, although most participants noted that dedicated time was also required. Additionally, leadership's ability to advocate for security at the executive level influenced budget approvals, resourcing decisions, and the organisation's overall security maturity.

This study contributed to a deeper understanding of how leadership styles, accountability, regulatory compliance, and resource constraints influenced SDP integration within agile teams. Compared to existing research, which primarily highlighted leadership's role in security awareness and policy enforcement, this study provided more granular insights into how leadership commitment, or lack thereof, directly influenced team-level security practices, decision-making, and long-term security adoption.

6.4 Conclusion regarding Research Question 3

The aim of RQ3 was to explore how tools and technology influenced the effective integration of SDPs. This study found that tools and technology played a critical role in enhancing the integration of SDPs by automating vulnerability detection, integrating security checks into CI/CD pipelines, and providing real-time feedback and recommendations to development teams.

Participants reported that effective tool integration improved agility, reduced human error, and fostered a proactive approach to security by enabling early detection and remediation of vulnerabilities. However, the study also identified significant challenges, particularly in legacy systems, which limited the benefits of modern security tools and created barriers to efficient integration.

While modern cloud solutions and security tools such as SAST and DAST streamlined the integration of SDPs, outdated architectures delayed vulnerability fixes and increased reliance on manual workarounds. This gap highlighted the friction between security automation and the technical debt associated with legacy systems. Teams faced difficulties integrating modern security tools into older software environments, often requiring extensive modifications or workaround solutions to maintain compatibility.

Another key insight was the risk of overreliance on security tools. Some participants cautioned that organisations could become complacent, assuming that automated scans were equivalent to achieving a mature security posture. While security tools provide critical insights, they do not fully eliminate the need for developer awareness, secure coding practices, and manual security reviews and testing. False positives, usability issues, and limited tool coverage across diverse application stacks were additional concerns that hindered seamless integration.

In the South African financial services industry, where regulatory compliance and data protection laws are stringent, security tools are essential in ensuring compliance with frameworks such as POPIA and FICA. The ability to integrate automated security testing within development workflows reduced compliance burdens by offering consistent security validation. However, the study underscored the importance of strategic tool selection driven by developer evaluations and centralised decision-making committees. This structured approach ensured that tools aligned with both security and business requirements while mitigating integration challenges.

The findings revealed the interplay between tooling effectiveness and broader organisational factors. While the use of security tools improved agility, their success depended on organisational readiness, leadership commitment, and modernisation efforts. Organisations that invested in upskilling teams, modernising infrastructure, and aligning security practices with agile workflows saw greater benefits from security tool usage. Conversely, those with inadequate training, and outdated architectures struggled to fully leverage security tools' capabilities. Tool adoption alone did not guarantee security or agility. Instead, the study demonstrated that a combination of well-integrated tools, leadership support, and ongoing modernisation strategies was necessary to achieve sustainable SDP integration in agile teams.

6.5 Recommendations

The findings reveal that South African FSPs prioritise security, although improvements can be made in some cultural, leadership, and technological areas. The SAMM framework recommends that organisations increase their security posture across the five pillars of *Governance, Design, Implementation, Verification, and Operations*.

Agile teams have fostered a broad recognition of security importance through internal communications, phishing simulations, and regular awareness campaigns. Continuing to emphasise security in everyday activities ensures that team members remain vigilant and understand that security is everyone's responsibility and not only the job of technical personnel or the security team within the organisation. Organisations should employ security champions or dedicated security leads within agile teams and ensure that leadership's performance indicators include security objectives.

Participants noted that the organisational culture of openness allowed software engineers, QA engineers, POs, and other stakeholders to voice security concerns. Agile teams should continue with this inclusive culture, ensuring that anyone can raise potential risks, thus keeping security front and centre in discussions and decisions.

It was found that some agile teams prioritised feature delivery over security or only introduced security fixes at the end of the development cycle. Organisations should introduce metrics that evaluate both security and delivery performance. For example, targets can be set to reduce critical vulnerabilities detected per release and track time-to-remediate security issues alongside feature delivery velocity. By incorporating these metrics into leadership measures, accountability and visibility can be ensured at the highest levels.

On a leadership level, proactive leadership has been shown to positively influence the agile teams' ability to integrate SDPs into their processes effectively through open communication and showing a commitment to security initiatives. To address a reactive approach, organisations can formalise roles and responsibilities in a model that holds leaders accountable for security outcomes.

Many teams have successfully integrated security scanning tools into their development pipelines, leading to early vulnerability detection, actionable feedback, and more secure code releases. Maintaining and updating these tools will help teams keep security aligned with speed and agility.

When vulnerabilities are detected, agile teams have shown they can quickly reprioritise efforts to fix critical issues. Continuing to respond swiftly to critical security findings and treating these issues with the same urgency as major feature defects ensures that short-term risks are effectively mitigated, and long-term technical debt is not accrued.

The study highlighted that architects, software engineers, and QA engineers often collaborate informally to discuss design decisions and identify security considerations early on. Continuing to hold regular architecture forums and reviews facilitates knowledge transfer and keeps security embedded in the development process.

Teams have been following set policies, standards, and risk assessments provided by their organisations. Continuously improving and adhering to these mandatory security practices ensures consistency, compliance with regulatory requirements, and a baseline to which new team members and stakeholders can easily align.

Some teams have been tracking the reduction of critical and high-severity vulnerabilities to measure tool effectiveness and security improvements. This data-driven approach supports continuous improvement, helps justify investments in certain tools or training, and demonstrates the tangible value of SDPs to leadership.

6.6 Contribution of the Study

This study contributes to the existing literature on secure software development by investigating the relationship between organisational culture, structural hierarchies, and environmental factors within the South African financial services industry. The research enhances theoretical knowledge by emphasising how cultural norms, such as prioritising feature delivery over security, communication practices and policies, influence the effective integration of SDPs. In doing so, it contributes to understanding how top-down mandates and industry pressures overlap with day-to-day team dynamics, highlighting the role of leadership support and employee awareness in mitigating cyber risks.

Methodologically, the qualitative study captured the lived experiences and perceptions of diverse stakeholders within the agile team, including software engineers, DevOps engineers, QA engineers, architects, product owners, project managers and management. This approach offered a contextually richer perspective than quantitative measures alone, uncovering how organisational structures, such as siloed teams and a lack of designated security champions, hinder or support security integration across the SDLC.

Beyond its theoretical insights, the study provides actionable recommendations based on OWASP SAMM. It highlighted targeted interventions such as role-specific training

modules to address skill deficits and the deployment of security champions to bridge communication gaps between development and security teams. These findings serve as a roadmap for senior management and leaders, who can leverage organisational policies and structures to foster a security-conscious culture, and technical personnel who integrate security considerations into their daily workflows.

Consequently, the study enriches the global discourse on agile security integration. It offers a nuanced understanding of how local industry standards and cultural factors shape development efforts. This dual emphasis on theoretical advancement and practical applicability paves the way for future empirical inquiry, especially in highly regulated sectors seeking to reconcile the tension between rapid software delivery and the imperative for robust security protocols.

6.7 Suggestions for Further Research

Based on the study's findings and identified gaps, several recommendations for future research are made. This study focused on agile teams within the South African financial services industry. The suggestions aim to delve deeper and expand the current understanding of effectively integrating SDPs within agile teams.

This study found that while security awareness has improved, some teams still prioritise speed over security in the SDLC. Examining how cultural norms evolve, especially in response to leadership changes, regulatory shifts, or technological advancements, would provide deeper insights into the sustainability of security-conscious mindsets. Hence, longitudinal studies are recommended. Researchers can conduct longitudinal case studies to track organisational cultures, security metrics, and agile team performance over multiple release cycles or across different leadership tenures.

Legacy systems emerged as a barrier to leveraging modern security tools. Limited empirical research exists on transitioning older, monolithic applications to more secure, modular architectures within agile environments. A mixed-methods approach could also yield more insights into digital transformation by targeting a larger population of IT professionals in the South African financial services industry.

Comparative studies examining different regulatory environments, comparing FSPs in South Africa against international banks and insurers to identify which compliance

measures effectively drive more effective integration of SDPs and those that create bureaucratic overhead. While financial services have high stakes, other regulated industries (e.g., healthcare and telecommunications) may encounter similar or unique security integration challenges. Comparing insights across different industries could reveal commonalities and industry-specific nuances.

This study highlighted the impact of reactive versus proactive leadership. Yet, the different leadership styles, such as transformational, transactional, and servant leadership, and their effect on the effective integration of SDPs have not been explored in detail. Qualitative research using structured leadership style assessments correlated with measured security outcomes could reveal which leadership characteristics most effectively integrate SDPs into the agile processes.

By pursuing these research avenues, future studies can build on this work's foundation, providing deeper insights into how agile teams, in various contexts, can effectively and sustainably integrate SDPs into their current workflows. Such studies would contribute to theoretical knowledge and yield pragmatic guidance for organisations, leaders, security practitioners, and development teams navigating the evolving landscape of secure software development.

References

- Alenezi, M., Basit, H. A., Beg, M. A., & Shaukat, M. S. (2022). Synthesizing secure software development activities for linear and agile lifecycle models. *Software - Practice and Experience*, 52(6), 1426–1453. <https://doi.org/10.1002/spe.3072>
- Alshaikh, M., & Adamson, B. (2021). From awareness to influence: toward a model for improving employees' security behaviour. *Personal and Ubiquitous Computing*, 25(5), 829–841. <https://doi.org/10.1007/s00779-021-01551-2>
- Angermeir, F., Voggenreiter, M., Moyon, F., & Mendez, D. (2021). Enterprise-Driven Open Source Software: A Case Study on Security Automation. *Proceedings - International Conference on Software Engineering*, 278–287. <https://doi.org/10.1109/ICSE-SEIP52600.2021.00037>
- Appiah, G., Amankwah-Amoah, J., & Liu, Y. L. (2022). Organizational Architecture, Resilience, and Cyberattacks. *IEEE Transactions on Engineering Management*, 69(5), 2218–2233. <https://doi.org/10.1109/TEM.2020.3004610>
- Ardo, A. A., Bass, J. M., & Gaber, T. (2022). Towards Secure Agile Software Development Process: A Practice-Based Model. *Proceedings - 48th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2022*, 149–156. <https://doi.org/10.1109/SEAA56994.2022.00031>
- Arizon-Peretz, R., Hadar, I., & Luria, G. (2022). The Importance of Security Is in the Eye of the Beholder: Cultural, Organizational, and Personal Factors Affecting the Implementation of Security by Design. *IEEE Transactions on Software Engineering*, 48(11), 4433–4446. <https://doi.org/10.1109/TSE.2021.3119721>
- Arizon-Peretz, R., Hadar, I., Luria, G., & Sherman, S. (2021). Understanding developers' privacy and security mindsets via climate theory. *Empirical Software Engineering*, 26(6). <https://doi.org/10.1007/s10664-021-09995-z>
- Atkins, L., Banerjee, S., Craig, L., Hao, G., Greis, J., Boer, M., & Idler, M. (2024). *The cyber clock is ticking: Derisking emerging technologies in financial services*. <https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/the-cyber-clock-is-ticking-derisking-emerging-technologies-in-financial-services#/>
- Atlassian. (2024). *What is DevOps?* <https://www.atlassian.com/devops>

- Bhuiyan, M. Y., Othman, S. H., & Raja Mohd. Radzi, R. Z. (2019). An Enhancement of TOE Model by Investigating the Influential Factors of Cloud Adoption Security Objectives. *International Journal of Innovative Computing*, 9(1). <https://doi.org/10.11113/ijic.v9n1.192>
- Bingham, A. J. (2023). From Data Management to Actionable Findings: A Five-Phase Process of Qualitative Data Analysis. *International Journal of Qualitative Methods*, 22. <https://doi.org/10.1177/16094069231183620>
- Bisschoff, C. A., & Els, D. (2023). Brand loyalty as a competitive advantage for South African banks. *Banks and Bank Systems*, 18(1), 103–115. [https://doi.org/10.21511/bbs.18\(1\).2023.09](https://doi.org/10.21511/bbs.18(1).2023.09)
- Börstler, J., Bin Ali, N., Petersen, K., & Engström, E. (2024). Acceptance behavior theories and models in software engineering-A mapping study. *Information and Software Technology*, 172, 107469. <https://doi.org/10.5281/zenodo.8>
- Clarke, V., & Braun, V. (2017). Thematic analysis. In *Journal of Positive Psychology* (Vol. 12, Issue 3, pp. 297–298). Routledge. <https://doi.org/10.1080/17439760.2016.1262613>
- Creswell, J. W. (2013). *Qualitative Inquiry & Research Design* (3rd ed.). SAGE Publications.
- Creswell, J. W., & Creswell, J. D. (2018). *Research design: qualitative, quantitative, and mixed methods approaches* (5th ed.). Sage Publications.
- Cunningham, M. (2001). *Manifesto for Agile Software Development*. <https://agilemanifesto.org/>
- Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. Massachusetts Institute of Technology.
- Dawood, Z., Shibambu, A., Mkhwanazi, T., Mahlasela, O., Baloyi, E., & Siphambili, N. (2024). *Cybersecurity Resilience of South Africa's Public Sector - Report 1: Cybersecurity Awareness and Preparedness*. www.csir.co.za
- Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. In *Journal of*

- Systems and Software* (Vol. 85, Issue 6, pp. 1213–1221). Elsevier Inc. <https://doi.org/10.1016/j.jss.2012.02.033>
- Ellis, J., & Hart, D. (2023). Strengthening the Choice for a Generic Qualitative Research Design. *The Qualitative Report*. <https://doi.org/10.46743/2160-3715/2023.5474>
- ENISA. (2023). *ENISA Threat Landscape 2023*. <https://doi.org/10.2824/782573>
- Ergen, A., Ünal, A. N., & Saygili, M. S. (2021). Is it possible to change the cyber security behaviours of employees? Barriers and promoters. *Academic Journal of Interdisciplinary Studies*, 10(4). <https://doi.org/10.36941/AJIS-2021-0111>
- Fauzi, M. F., Mohan, V. R., Qi, Y., Chandrasegar, C., & Saira Muzafar . (2023). Secure Software Development Best Practices. *International Journal of Emerging Multidisciplinaries: Computer Science & Artificial Intelligence*, 2(1). <https://doi.org/10.54938/ijemdc sai.2023.02.1.256>
- FSCA. (2022). *Financial Sector Outlook Study*. <https://www.fsc a.co.za/Documents/FSCA%20Financial%20Sector%20Outlook%20Study%202022.pdf>
- Gasiba, T. E., Lechner, U., Pinto-Albuquerque, M., & Mendez, D. (2021). Is Secure Coding Education in the Industry Needed? An Investigation through a Large Scale Survey. *Proceedings - International Conference on Software Engineering*, 241–252. <https://doi.org/10.1109/ICSE-SEET52601.2021.00034>
- Guest, G., Bunce, A., & Johnson, L. (2006). How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods*, 18(1), 59–82. <https://doi.org/10.1177/1525822X05279903>
- Gutfleisch, M., Klemmer, J. H., Busch, N., Acar, Y., Sasse, M. A., & Fahl, S. (2021). *How Does Usable Security (Not) End Up in Software Products? Results From a Qualitative Interview Study*. <https://doi.org/10.25835/0089554>
- Gutfleisch, M., Schöps, M., Horstmann, S. A., Wichmann, D., & Angela Sasse, M. (2023). Security Champions Without Support: Results from a Case Study with OWASP SAMM in a Large-Scale E-Commerce Enterprise. *ACM International*

- Conference Proceeding Series*, 260–276.
<https://doi.org/10.1145/3617072.3617115>
- Herath, T. C., Herath, H. S. B., & D'Arcy, J. (2020). Organizational Adoption of Information Security Solutions: An Integrative Lens Based on Innovation Adoption and the Technology-Organization-Environment Framework. *The DATA BASE for Advances in Information Systems*, 51(2).
<https://doi.org/10.1145/3400043.3400046>
- IBM. (2023). *Cost of a data breach 2023*. <https://www.ibm.com/reports/data-breach>
- ISACA. (2024). *CMMI Institute - CMMI*. <https://cmmiinstitute.com/cmml>
- Jaatun, M. G., & Cruzes, D. S. (2021). Care and Feeding of Your Security Champion. *Cyber Science 2021*. <https://www.sintef.no/sos-agile/>
- Janisar, A., Shafee, K., Sarlan, A., Maiwada, U., & Salameh, A. A. (2024). Securing Software Development: A Holistic Exploration of Security Awareness in Software Development Teams. *International Journal of Academic Research in Business and Social Sciences*, 14(1). <https://doi.org/10.6007/ijarbss/v14-i1/20545>
- Jenkins, J. (2022). Toward Remaking Software Development to Secure It. *The Journal of the Southern Association for Information Systems*, 9(1), 15–37.
<https://doi.org/10.17705/3JSIS.00020>
- Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Systematic Mapping Study on Security Approaches in Secure Software Engineering. *IEEE Access*, 9.
<https://doi.org/10.1109/ACCESS.2021.3052311>
- Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2022). Systematic Literature Review on Security Risks and its Practices in Secure Software Development. In *IEEE Access* (Vol. 10, pp. 5456–5481). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2022.3140181>
- Kreitz, M. (2019). Security by Design in Software Engineering. *ACM SIGSOFT Software Engineering Notes*, 44(3), 23–23.
<https://doi.org/10.1145/3356773.3356798>
- Lende, D., Monkhouse, A., Ligatti, J., & Ou, X. (2023). Co-Creation in Secure Software Development: Applied Ethnography and the Interface of Software and

- Development. *Human Organization*, 82(1), 13–24. <https://doi.org/10.17730/1938-3525-82.1.13>
- Marandi, M., Bertia, A., & Silas, S. (2023). Implementing and Automating Security Scanning to a DevSecOps CI/CD Pipeline. *2023 World Conference on Communication and Computing, WCONF 2023*. <https://doi.org/10.1109/WCONF58270.2023.10235015>
- Maxwell, J. A. (2013). *Qualitative Research Design: An Interactive Approach* (L. Bickman & D. J. Rog, Eds.; 3rd ed.). Sage Publications.
- McGraw, G. (2004). *Building Security In*. www.computer.org/security/
- Mihelič, A., Vrhovec, S., & Hovelja, T. (2023). Agile Development of Secure Software for Small and Medium-Sized Enterprises. *Sustainability (Switzerland)*, 15(1). <https://doi.org/10.3390/su15010801>
- Misra, S., Kumar, V., Kumar, U., Fantasy, K., & Akhter, M. (2012). Agile software development practices: Evolution, principles, and criticisms. In *International Journal of Quality & Reliability Management* (Vol. 29, Issue 9, pp. 972–980). <https://doi.org/10.1108/02656711211272863>
- Moyon, F., Almeida, P., Riofrio, D., Mendez, D., & Kalinowski, M. (2020). Security Compliance in Agile Software Development: A Systematic Mapping Study. *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*, 413–420. <https://doi.org/10.1109/SEAA51224.2020.00073>
- Moyón, F., Angermeir, F., & Mendez, D. (2024). *Industrial Challenges in Secure Continuous Development*. <https://doi.org/10.1145/3639477.3639736>
- Naeem, M., Ozuem, W., Howell, K., & Ranfagni, S. (2023). A Step-by-Step Process of Thematic Analysis to Develop a Conceptual Model in Qualitative Research. *International Journal of Qualitative Methods*, 22. <https://doi.org/10.1177/16094069231205789>
- Nurgalieva, L., Frik, A., & Doherty, G. (2023). A Narrative Review of Factors Affecting the Implementation of Privacy and Security Practices in Software Development. *ACM Computing Surveys*, 55(14 S). <https://doi.org/10.1145/3589951>

- Ofoegbu, K. D. O., Osundare, O. S., Ike, C. S., Fakeyede, O. G., & Ige, A. B. (2024). Proactive cyber threat mitigation: Integrating data-driven insights with user-centric security protocols. *Computer Science & IT Research Journal*, 5(8), 2083–2106. <https://doi.org/10.51594/csitrj.v5i8.1493>
- Onubedo, O. (2023, June 23). *FICA vs POPIA: Navigating South Africa's KYC compliance regulation*. <https://usesmileid.com/blog/fica-vs-popia-navigating-south-africa-s-kyc-compliance-regulation>
- Otieno, M., Odera, D., & Ounza, J. E. (2023). Theory and practice in secure software development lifecycle: A comprehensive survey. *World Journal of Advanced Research and Reviews*, 18(3), 053–078. <https://doi.org/10.30574/wjarr.2023.18.3.0944>
- OWASP. (2010). *OWASP Secure Coding Practices*. [https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010-2017%20\(en\).pdf](https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010-2017%20(en).pdf)
- OWASP. (2024). *OWASP SAMM*. <https://owasp.org/www-project-samm/>
- Palombo, H., Tabari, A. Z., Lende, D., Ligatti, J., & Ou, X. (2020, August 10). An Ethnographic Understanding of Software (In)Security and a Co-Creation Model to Improve Secure Software Development. *Proceedings of the Sixteenth Symposium on Usable Privacy and Security*. <https://www.usenix.org/conference/soups2020/presentation/palombo>
- Pan, Z., Shen, W., Wang, X., Yang, Y., Chang, R., Liu, Y., Liu, C., Liu, Y., & Ren, K. (2024). *Ambush from All Sides: Understanding Security Threats in Open-Source Software CI/CD Pipelines*. <https://doi.org/10.1109/TDSC.2023.3253572>
- Parkin, S., Kuhn, K., & Shaikh, S. A. (2023). Executive decision-makers: a scenario-based approach to assessing organizational cyber-risk perception. *Journal of Cybersecurity*, 9(1). <https://doi.org/10.1093/cybsec/tyad018>
- Percy, W. H., Kostere, K., & Kostere, S. (2015). Generic Qualitative Research in Psychology. *The Qualitative Report*, 20(2), 76–85. <http://www.nova.edu/ssss/QR/QR20/2/percy5.pdf>

- PWC. (2024). *Global Digital Trust Insights Survey 2025: South Africa and Africa Report*. <https://www.pwc.co.za/en/publications/digital-trust-insights.html>
- Rangnau, T., Buijtenen, R. V., Fransen, F., & Turkmen, F. (2020). Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines. *Proceedings - 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, EDOC 2020*, 145–154. <https://doi.org/10.1109/EDOC49727.2020.00026>
- Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., & Leppänen, V. (2021). Security in agile software development: A practitioner survey. *Information and Software Technology*, 131. <https://doi.org/10.1016/j.infsof.2020.106488>
- Rogers, E. M. (1983). *Diffusion of innovations*. Free Press.
- Ryan, I., Roedig, U., & Stol, K. J. (2023). Measuring Secure Coding Practice and Culture: A Finger Pointing at the Moon is not the Moon. *Proceedings - International Conference on Software Engineering*, 1622–1634. <https://doi.org/10.1109/ICSE48619.2023.00140>
- SAFECode. (2018). *Fundamental Practices for Secure Software Development - Third Edition*. https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf
- Salleh, K. A., & Janczewski, L. (2016). Technological, Organizational and Environmental Security and Privacy Issues of Big Data: A Literature Review. *Procedia Computer Science*, 100, 19–28. <https://doi.org/10.1016/j.procs.2016.09.119>
- Sas, M., Hardyns, W., van Nunen, K., Reniers, G., & Ponnet, K. (2021). Measuring the security culture in organizations: a systematic overview of existing tools. *Security Journal*, 34(2), 340–357. <https://doi.org/10.1057/s41284-020-00228-4>
- Saunders, M. N., Lewis, P., & Thornhill, A. (2023). *Research Methods for Business Students* (9th ed.). Pearson.
- Schneider, B., Asprion, P. M., Androvicsova, S., & Azan, W. (2020). A Practical Guideline for Developing a Managerial Information Security Awareness Program.

- AMCIS 2020 Proceedings, 12, 18.
https://doi.org/https://aisel.aisnet.org/amcis2020/info_security_privacy/info_security_privacy/18
- Schubert, M. A., Pagel, S., & von Korfflesch, H. F. O. (2023). Success Factors in Secure Software Development of Cloud Applications in Germany: A Qualitative-explorative Expert Study. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2023-January. <https://doi.org/10.24251/HICSS.2023.80>
- Shostack, A. (2022). Fast, Cheap, Good: Lightweight Methods are Undervalued. *IEEE*. <https://doi.org/10.48550/arXiv.2301.03593>
- Synopsys. (2024). *Building Security Maturity Model (BSIMM)*. <https://www.synopsys.com/software-integrity/software-security-services/bsimm-maturity-model.html>
- Taherdoost, H. (2018). A review of technology acceptance and adoption models and theories. *Procedia Manufacturing*, 22, 960–967. <https://doi.org/10.1016/j.promfg.2018.03.137>
- Tejay, G. P. S., & Mohammed, Z. A. (2023). Cultivating security culture for information security success: A mixed-methods study based on anthropological perspective. *Information and Management*, 60(3). <https://doi.org/10.1016/j.im.2022.103751>
- Tolah, A., Furnell, S. M., & Papadaki, M. (2021). An empirical analysis of the information security culture key factors framework. *Computers and Security*, 108. <https://doi.org/10.1016/j.cose.2021.102354>
- Tøndel, I. A., Cruzes, D. S., Jaatun, M. G., & Sindre, G. (2022). Influencing the security prioritisation of an agile software development project. *Computers and Security*, 118. <https://doi.org/10.1016/j.cose.2022.102744>
- Tong, A., Sainsbury, P., & Craig, J. (2007). Consolidated criteria for reporting qualitative research (COREQ): A 32-item checklist for interviews and focus groups. *International Journal for Quality in Health Care*, 19(6), 349–357. <https://doi.org/10.1093/intqhc/mzm042>

- Tornatzky, L., & Fleischer, M. (1990). *The Process of Technology Innovation*. Lexington Books.
- Uchendu, B., Nurse, J. R. C., Bada, M., & Furnell, S. (2021). Developing a cyber security culture: Current practices and future needs. *Computers and Security*, 109. <https://doi.org/10.1016/j.cose.2021.102387>
- Umeugo, W. (2023). Factors Affecting the Adoption of Secure Software Practices in Small and Medium Enterprises that Build Software In-House. *International Journal of Advanced Research in Computer Science*, 14(02), 1–7. <https://doi.org/10.26483/ijarcs.v14i2.6955>
- Valdés-Rodríguez, Y., Hochstetter-Diez, J., Díaz-Arancibia, J., & Cadena-Martínez, R. (2023). Towards the Integration of Security Practices in Agile Software Development: A Systematic Mapping Review. *Applied Sciences (Switzerland)*, 13(7). <https://doi.org/10.3390/app13074578>
- Valdés-Rodríguez, Y., Hochstetter-Diez, J., Diéguez-Rebolledo, M., Bustamante-Mora, A., & Cadena-Martínez, R. (2024). Analysis of Strategies for the Integration of Security Practices in Agile Software Development: A Sustainable SME Approach. *IEEE Access*, 12. <https://doi.org/10.1109/ACCESS.2024.3372385>
- Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2), 273–315. <https://doi.org/10.1111/j.1540-5915.2008.00192.x>
- Veracode. (2024). *The State of Software Security 2024: Addressing the Threat of Security Debt*.
- Wallace, S., Green, K. Y., Johnson, C., Cooper, J., & Gilstrap, C. (2020). An Extended TOE Framework for Cybersecurity-adoption Decisions. *Communications of the Association for Information Systems*, 47, 227–247. <https://doi.org/10.17705/1CAIS.04716>
- Weir, C., Becker, I., & Blair, L. (2023). Incorporating software security: using developer workshops to engage product managers. *Empirical Software Engineering*, 28(2). <https://doi.org/10.1007/s10664-022-10252-0>

- Weir, C., Rashid, A., & Noble, J. (2021). Challenging software developers: Dialectic as a foundation for security assurance techniques. *Journal of Cybersecurity*, 6(1). <https://doi.org/10.1093/CYBSEC/TYAA007>
- Williams, M., & Moser, T. (2019). The Art of Coding and Thematic Exploration in Qualitative Research. *International Management Review*, 15(1).
- Zakaria, O., Gani, A., Nor, M. M., & Anuar, N. B. (2007, June 17). Reengineering Information Security Culture Formulation Through Management Perspective. *Proceedings of the International Conference on Electrical Engineering and Informatics*.

APPENDIX A: Participant Information Form

Good day,

My name is Kaylieh Manikam. I am a master's student in the Wits Business School, at the University of the Witwatersrand, Johannesburg. I am conducting a research study about the factors influencing the effective integration of secure development practices under the supervision of Dr Pius Oba. The aim of the study is to explore the organisational, leadership and technology factors influencing effective integration of secure development practices in agile teams.

I am inviting you to take part in a 30–45-minute interview. The interview will take place online at a time that is convenient for you. With your permission, I would like to audio record the interview. This data will be stored in Microsoft OneDrive and deleted after 2 years. Only the researcher will have access to the data.

The interview will be confidential and anonymous. When I share the results of the research study, I will not include your name, organisation or anything else that could identify you.

If you decide to take part in the research study, it should be because you want to volunteer. You do not have to take part. You can stop being in the study at any time. You do not have to answer any questions if you do not want to. You will not get any direct benefits if you choose to join the research study. You will not lose any services, benefits or rights you would normally have if you decided not to join. Taking part in the research study will not cost you anything. You will not be paid for being in this research study. The risks for this research study are minimal and no more than what happens in everyday life.

This research study will be written up as a research report. The report will be available on the university library website. If you would like to receive a summary of this report, I will be happy to send it to you.

If you have any questions during or afterwards about this research study, feel free to contact me or my supervisor on the details listed below. If you have any concerns or complaints about the ethical procedures of this research study, you are welcome to contact the University Human Research Ethics Committee (Non-Medical), telephone +27(0) 11 717 1408, email hrecnon-medical@wits.ac.za.

Yours sincerely,

Researcher:

Kaylieh Manikam, 1542635@students.wits.ac.za, +27(0) 84 633 0155, +31(0) 62 3059 183

Supervisor:

Dr Pius Oba, pius.oba@wits.ac.za, +27(0) 117173976

APPENDIX B: Participant Consent Form

The factors influencing the effective integration of secure development practices in agile teams

Kaylieh Manikam

I, ..., agree to participate in this research project.

I agree to the following:

(Please circle the relevant options below)

The research study was explained to me. I understand what this study is about	YES	NO
---	-----	----

I understand that I can volunteer to take part in the study	YES	NO
---	-----	----

I agree that the interview may be audio recorded	YES	NO
--	-----	----

I agree that direct quotations from my interview may be used by the researcher in their research report	YES	NO
---	-----	----

I agree that my participation will remain anonymous (my name or other identifying data will not be used by the researcher in their research report)	YES	NO
---	-----	----

I agree that other researchers may use the information I provide in my interview (depending on their own ethics clearance being obtained) but my name and any personal information will not be used or passed on	YES	NO
--	-----	----

..... (signature)

..... (name of participant)

..... (date)

..... (signature)

..... (name of researcher/person seeking consent)

..... (date)

APPENDIX C: Consistency Table

Table 4. Consistency Table - RQs, Propositions, Data Collection and Analysis

RQ#	Research Question	Proposition #	Proposition	Data Collection Detail	Data Analysis Method
1	How do organisational structure and culture influence the integration of SDPs in agile teams?	1	Organisational structure and culture are vital to creating a security-conscious mindset within agile teams. Within the organisational structure, the security champion role can effectively bridge the gap between security and software development teams. Cultural factors such as awareness, training, and cross-team collaboration can create a security culture, leading to the effective integration of SDPs.	Semi-structured Interview Interview Questions (1.1 - 1.10)	Thematic Analysis
2	How does leadership support and commitment influence the integration of SDPs in agile teams?	2	Leadership commitment and support are driving factors in fostering a security-conscious environment, where agile teams are empowered and possess the necessary	Semi-structured Interview Interview Questions (2.1 - 2.7)	Thematic Analysis

RQ#	Research Question	Proposition #	Proposition	Data Collection Detail	Data Analysis Method
			resources, skills, and knowledge to integrate security into their processes and, therefore increase the software's security posture.		
3	How does the use of tools and technology influence the successful integration of SDPs in agile teams?	3	Tools and technology can help agile teams successfully integrate SDPs into their existing processes by implementing security practices within their CI/CD pipelines. Factors such as organisational readiness, compatibility, budget, and technical skills play a role in the selection and adoption of tools.	Semi-structured Interview Interview Questions (3.1 - 3.6)	Thematic Analysis

APPENDIX D: Research Instrument

Table 5. Interview Questions

RQ 1: Organisational Structure and Culture		
Participants	No	Question
All	1.1	How would you describe the culture in your organisation regarding security with examples? (Is it a priority? How is it communicated?)
	1.2	Are there any specific cultural values or norms that impact the adoption of SDPs?
	1.3	How are communication and collaboration managed across different departments regarding security (e.g., security team, development team)?
	1.4	How does your team's structure either help or hinder integrating security practices? (Is there a security champion within the team?)
	1.5	Describe instructions, policies, standards or any other documentation that you get from the organisation regarding security
Product Owner	1.6	How do you balance security requirements with other product features?
Product Owner	1.7	How do you communicate security risks to stakeholders?
Scrum Master	1.8	How do you prioritise security concerns within sprints?
IT Manager	1.9	How do you create awareness and communicate the importance of security to your team?
IT Manager	1.10	How do you advocate for security within the organisation?
RQ 2: Leadership commitment and support		
Participants	No	Question
All	2.1	How would you describe the level of support from leadership (managers, executives) for integrating security practices? Do you feel encouraged to do this?

	2.2	Do you feel empowered to raise security concerns or suggest improvements? Why or why not?
	2.3	Can you share an instance where leadership's actions or decisions directly influenced the team's ability to implement security practices?
	2.4	What kind of support (e.g., resources, training, incentives) do you receive from leadership to integrate secure development practices in your agile team?
Product Owner	2.5	Does leadership understand the impact of security on product development timelines?
IT Manager	2.6	What challenges do you face in getting resources for security initiatives?
IT Manager	2.7	What support do you need from higher-level leadership to improve security?
RQ 3: Tools and Technology		
Participant	No	Question
All	3.1	How has the use, or lack of technology, affected the agility of the team in terms of delivery of working software?
	3.2	What tools and technologies do your team currently use to support secure development practices? (SAST, DAST, threat modelling, etc.)
	3.3	How were these tools and technologies chosen and implemented in your team?
	3.4	How effective are these tools in integrating secure development practices within your agile/ DevOps processes?
	3.5	What challenges have you encountered with the tools and technologies used for secure development practices?
	3.6	How do you measure the effectiveness of security tools?

APPENDIX E: Ethics Clearance Certificate

Graduate School of Business Administration
University of the Witwatersrand, Johannesburg



Wits Business School Ethics Committee
Constituted under the University Human Research Ethics Committee (Non-Medical)

Ethics Clearance Certificate

Ethics protocol number: WBS/DB1546235/100

This certificate is only valid with a legitimate ethics protocol number and signed by the Researcher (below).

Project title	An exploration of the factors influencing the effective integration of secure development practices in agile teams
Investigator / Researcher	Mrs Kaylieh Manikam
Nature of Project	MM (Digital Business)
Decision of the Committee	Approved, provided stakeholders and participants are guaranteed confidentiality.
Issue Date of Certificate	02/09/2024
Expiry date	Date of submission of the project / research report
Chairperson	Dr Ayanda Magida  +27 11 717 3953  ayanda.magida@wits.ac.za 

Declaration by Researcher

One copy must be signed by the Researcher and returned to the Chairperson of the Wits Business School Ethics Committee.

I fully understand the conditions under which I am authorized to carry out the abovementioned research and I guarantee to ensure compliance with these conditions. Should any departure to be contemplated from the research procedure as approved I undertake to resubmit the protocol to the Committee.

Signature

2 September 2024

Date: