A Conceptual Framework for Global DevSecOps: Delphi-AHP Study

Xiaofan (Gavin) Zhao

2025

School of Engineering, Computer and Mathematical Sciences

Auckland University of Technology (AUT)

A thesis submitted to the Auckland University of Technology in fulfilment of the requirements for the degree of Doctor of Philosophy

Primary Supervisor:

Dr. Ramesh Lal

Secondary Supervisor:

Assoc. Prof. Tony Clear

Abstract

Context: DevOps has become mainstream in the Software Engineering (SE) industry and academia, enhancing software development performance by bridging the gap between development (Dev) and operations (Ops). However, security requirements are often overlooked and devalued because they are perceived as hindrances to the high velocity required in DevOps. DevSecOps, as a security-oriented variant of DevOps, aims to integrate security into DevOps implementation by promoting collaboration among development (Dev), operations (Ops), and security (Sec) teams. Meanwhile, academia and industry's interest in another trend – Global Software Engineering (GSE), has also significantly increased. GSE is a business strategy that arranges software development teams geographically distributed across the world. The foundational idea of DevOps/DevSecOps is to reduce functional silos and foster collaboration, and it encounters magnified challenges in GSE contexts due to geographical, temporal, linguistic, and cultural distances. Researchers and practitioners have paid attention to DevOps adoption in GSE, yet a research gap exists between DevSecOps and GSE that warrants academic investigation.

Aim: This research aimed to provide an in-depth understanding of DevSecOps and its adoption in GSE by developing an empirically grounded conceptual framework.

Methods: This research was divided into two stages. First, a Multivocal Literature Review (MLR) study was conducted to explore the current state of DevSecOps. A Thematic Analysis (TA) was performed to identify, synthesise, and analyse themes within the data for reporting MLR results and to further establish a conceptual framework as a theoretical basis for the following research. Second, an empirical study was conducted to validate, refine, and upgrade the MLR findings. It employed a qualitative research methodology, incorporating a quantitative survey that combined a Delphi survey and the Analytic Hierarchy Process (AHP). The Delphi-AHP study consisted of three survey rounds with 18 international participants, who are DevSecOps experts with various roles, including academic, industrial, managerial, and technical. The data were collected via an online survey that used multiple question formats, including AHP pairwise comparisons, multiple-choice, and open-ended questions. A dissent analysis was conducted to determine whether there is consensus or dissent regarding DevSecOps.

Results: The MLR study identifies five aspects of DevSecOps research (Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement), collects related themes of each aspect, and generates a "DevSecOps CPTM (Challenge-Practice-Tool-Metric) Model (Version 1.0)" by integrating the themes of the latter four aspects. An unexplored area relating to the application of DevSecOps in GSE has been identified. Subsequently, the Delphi-AHP study evaluates and prioritises the identified challenges, practices, tools, and metrics, collects new items into each aspect, identifies slight differences between local and global DevSecOps, and upgrades the DevSecOps CPTM Model from Version 1.0 to 2.0 by incorporating additional GSE aspects. Additionally, the dissent analysis reveals that dissenting opinions exist on DevSecOps between the SE industry and academia.

Conclusion: This research provides implications for both practice and theory by providing an in-depth understanding of DevSecOps and its adoption in GSE. As the key artifact, the DevSecOps CPTM Model (Version 2.0) is presented to effectively support SE academia and industry by providing a broad landscape and a prioritised breakdown of DevSecOps, from which researchers and practitioners can select an area of focus to enhance their knowledge or practice. With DevSecOps spanning many stages of the lifecycle, the framework will enable the exploration of new emphases and future opportunities, such as AI-driven DevSecOps practices and tools.

Table of Contents

List of Figures

List of Tables

List of Equations

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Xiaofan (Gavin) Zhao

21/08/2025

Acknowledgements

First of all, I would like to express my most profound appreciation to my primary supervisor, Dr Ramesh Lal, for his insightful discussions, valuable guidance, and continuous support throughout my entire PhD journey. I am also grateful to my secondary supervisor, Associate Professor Tony Clear, for his mentorship, guidance, and advice, as well as for making all necessary resources available to support my PhD research.

In addition to the supervisor team, I would like to extend my gratitude to Professor Aurora Vizcaino and Associate Professor Sherlock Licorish for examining my PhD thesis and providing constructive feedback. I would also like to thank Professor Stephen MacDonell and Dr Mali Senapathi for their careful review and insightful feedback on my "Confirmation-of-Candidature" Research Proposal (PGR9 report).

I am indebted to the Faculty of Design Creative Technologies (FDCT) and the School of Engineering, Computer and Mathematical Sciences (SECMS) for my scholarship to pursue my studies at AUT. I also appreciate the employment opportunities provided by the SECMS. Then I would like to thank the Graduate Research School (GRS) at AUT for providing training, workshops, and conferences, which helped me build strong research skills and interpersonal relationships.

I must acknowledge the generous contributions of expertise and time that the participants made to the Delphi survey. Without their precious support and cooperation, it would not have been possible to conduct this research.

I am sincerely grateful to my family for their everlasting love, support, and encouragement. Last but not least, I would like to extend my thanks to all my friends and colleagues who have shown me kindness throughout my entire PhD journey.

Ethics Approval

This research has been approved by the Auckland University of Technology Ethics Committee (AUTEC) on 14 September 2023, AUTEC reference number is 23/225 “Representing global DevSecOps to usefully support software engineering practice”.

Confidential Materials

In respect of privacy and confidentiality, participants of this research are not identifiable in the research outputs or findings. Hence, the following confidential materials, which involve the identifiable information on the participants, are not provided:

- Consent forms signed by participants
- Participant management form
- Response sheets

1 Chapter 1: Introduction

This chapter introduces the research work investigating the topic of 'DevSecOps' reported in this doctoral thesis. In Section 1.1, the background and importance of conducting this research are provided. Next, the research motivation, aim, and objectives are presented in Section 1.2. Based on the above, the research design overview is briefly explained in Section 1.3. In Section 1.4, a summary of the key contributions of this work is discussed. The publications resulting from this research are listed in Section 1.5. Finally, the thesis structure is outlined in Section 1.6.

1.1 Research Background

Development Operations (DevOps) has gained popularity and is becoming mainstream in the Software Engineering (SE) industry and academia (Akbar, Rafi, et al., 2022; Gall & Pigni, 2022; NCCoE, 2025). It aims to facilitate collaboration between software development (Dev) and IT Operations (Ops) by automating tasks for building, deploying, and testing (Fitzgerald & Stol, 2017; Wiedemann et al., 2023). By breaking down silos, DevOps enables organisations to reduce software development time, provide continuous software delivery, and increase software stability, thereby enhancing customer satisfaction (Akbar, Rafi, et al., 2022; Fitzgerald & Stol, 2017; Hussain, Clear, & MacDonell, 2017; Wiedemann et al., 2023). However, security requirements are often overlooked and devalued because they are perceived as hindrances to the high velocity required in DevOps implementation (Myrbakken & Colomo-Palacios, 2017).

With software development, security considerations require two distinct focuses: first, ensuring the security of the software being developed by detecting and removing vulnerabilities and faulty code; second, safeguarding the overall software environment and development process against security threats (Morales et al., 2020). These two security considerations should be at the forefront, as DevOps has gained access to emerging tools and techniques with the rise of cloud-native technologies, microservice architectures, serverless frameworks, and artificial intelligence (AI) tools and capabilities (Fernandez & Brito, 2019; NCCoE, 2025).

An industry survey presented by SANS (Edmundson & Hartman, 2022) suggests that cloud-

hosted technologies for the software industry are gaining momentum. It reported that in 2022, 92% of the responding organisations were using clouds, and 25% were using multiple cloud providers; 65% of the responding organisations ran over 25% of their applications in the cloud; 8% of the responding organisations ran 100% of their applications in the cloud. Virtual machines, containers, and serverless were the top three cloud-hosted technologies for the software industry. Cloud environments needed to be secured, as the use of cloud-based technologies would not only benefit organisations but also pose security challenges due to the nature of cloud computing, which delivers IT resources on demand over the internet. On the other hand, the same survey (Edmundson & Hartman, 2022) highlights the underutilisation of security methods, such as Cloud Security Posture Management and Cloud Workload Protection Platform. The latest State of DevOps Report by Puppet (2025) also highlights security as a significant consideration of DevOps.

To build security into DevOps, the term 'DevSecOps' has emerged as a security-oriented variant of DevOps. It aims to integrate security considerations and requirements into DevOps practices without impacting implementation speed or feature quality. DevSecOps addresses risks and security issues through collaboration amongst development, operations, and security teams (Zaydi & Nassereddine, 2020). A key benefit of DevSecOps is that it shifts security implementation and testing upfront with development (called 'shift-security-left') (Myrbakken & Colomo-Palacios, 2017). It also enables continuous security implementation throughout the Software Development Lifecycle (SDLC) (Mohan & Othmane, 2016). Hence, DevSecOps practices help reduce security threats and address security issues earlier and faster (Carter, 2017). Another benefit of DevSecOps is that it automates security testing, allowing development teams to focus on improving security policies (Ahmed & Francis, 2019).

An increasing number of organisations are embracing DevSecOps practices due to those benefits. The SANS survey (Edmundson & Hartman, 2022) reports on the status of DevSecOps adoption: in 2022, 58% of responding organisations adopted DevSecOps to varying extents; 21% had not adopted DevSecOps; and 18% deemed their DevSecOps adoption to be "spurious". Gartner (Betts, 2022) forecasts that 85% of organisations will adopt DevSecOps practices by 2027, through migrating from DevOps to DevSecOps. Hence, there appears to be a lack of guidelines or reference models to support organisations in promoting DevSecOps adoption.

Evidence suggests that the academic and industry interests in Global Software Engineering (GSE), also known as Global Software Development (GSD), have significantly increased over the last two decades (Cico et al., 2021; Grande, Vizcaino, & Garcia, 2024). GSE is a business strategy that organises software development teams geographically distributed across the globe (Vizcaíno et al., 2016). It provides global access to specialised and diverse skilled human resources and enables reductions in software development costs and time-to-market by leveraging time-zone effectiveness and round-the-clock productivity (Conchuir et al., 2009; Vizcaíno et al., 2016).

Collaboration is pervasive throughout SE (Constantino et al., 2020; Whitehead et al., 2010), and the SE process is inherently collaborative, requiring many workers to share artifacts, coordinate activities, and communicate to produce large and complex software systems (Omoronyia et al., 2010; Whitehead et al., 2010). Thus, those modern software development paradigms such as DevOps, DevSecOps, GSE, and GSD are part of Collaborative Software Engineering (CoSE), which is ‘‘about creating the organisational structures, reward structures, and work breakdown structures that afford effective work towards goal’’ (Whitehead et al., 2010).

The foundational idea of DevOps/DevSecOps is to reduce functional silos and foster collaboration among teams (Rajapakse & Szabo, 2024). Adopting DevOps/DevSecOps in GSE contexts may encounter magnified challenges due to geographical, temporal, linguistic, and cultural distances (Jalali, Gencel, & Šmite, 2010; Tamburri et al., 2012). Both academia and industry are increasingly researching and adopting DevOps in GSE contexts (Cico et al., 2021; Grande, Vizcaino, & Garcia, 2024). As a security-oriented expansion of DevOps, the adoption of DevSecOps in GSE also requires an academic investigation.

1.2 Research Motivation, Aim, and Objectives

1.2.1 Research Motivation

A brief literature review was conducted to capture related work and identify the research gap (in Chapter 2). To summarise, the existing literature is relatively early or focuses on a single perspective or aspect of DevSecOps (primary issue), and little is known about its application in the GSE context (secondary issue). Hence, a substantial body of academic research on the topic

has not yet been built. Furthermore, although research is moving toward exploring the interrelation among multiple DevSecOps aspects and providing a comprehensive understanding, evidence suggests that there remains a shortage of guidelines and reference models for the adoption of DevSecOps.

Therefore, the motivation for this research is to gain a thorough understanding of DevSecOps by developing an empirically grounded conceptual framework that represents the global DevSecOps approach and helps to identify the adoption of relevant structures (teams, roles) and practices based on a collaborative software engineering method. The conceptual framework is defined as a network or "a plane" of interlinked concepts that together provide a comprehensive understanding of a phenomenon or phenomena (Jabareen, 2009).

1.2.2 Research Aim

The aim of this research is to provide an in-depth understanding of DevSecOps and its adoption in GSE by developing an empirically grounded conceptual framework.

1.2.3 Research Objectives

To achieve the overall research aim, the specific objectives of this research are listed as follows:

For the theoretical foundation:

- To explore the current state of DevSecOps in the existing white and grey literature, on which to base this research.
- To explore the adoption of DevSecOps in GSE from the existing white and grey literature.
- To establish a conceptual framework of DevSecOps based on the existing literature.

For the validation and refinement:

- To validate and refine the conceptual framework through an empirical investigation.
- To investigate differences between DevSecOps in local and global settings and further upgrade the drafted conceptual framework into a global version that could guide practitioners adopting the DevSecOps approach to support software engineering practices in a GSE setting.
- To investigate whether a consensus or dissent exists on the DevSecOps approach between

the SE industry and academia.

1.3 Research Design

To address these objectives and achieve the overall research aim, this doctoral research was divided into two stages:

- For the theoretical foundation, in Stage One, a Multivocal Literature Review (MLR) study was conducted to identify existing research and practical trends and establish a DevSecOps conceptual framework, which served as a theoretical basis for further research.
- For the validation and refinement of the DevSecOps conceptual framework, in Stage Two, a combined Delphi-AHP study was conducted for an empirical investigation to validate, refine, and upgrade the conceptual framework.

1.3.1 Research Questions

Two sets of research questions were therefore raised as follows:

Research/Review Questions of the MLR study:

- ***RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect, and their links) in the existing white and grey literature?***

  Sub-question 1.1: What aspects of DevSecOps can be found in the existing white and grey literature?

  Sub-question 1.2: What themes do these aspects contain?

  Sub-question 1.3: How do the identified aspects and themes link to each other?

- ***RQ2: How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?***

Research Questions of the Delphi-AHP Study:

- ***RQ3: How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?*** *(Challenges, practices, tools, and metrics are four aspects of DevSecOps, which have been identified by the MLR. They are also four elements of the conceptual model.)*

Sub-question 3.1: What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?

Sub-question 3.2: Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?

- ***RQ4: What are the experts' opinions on DevSecOps in GSE contexts?***

  Sub-question 4.1: How is DevSecOps different between local and global settings?

  Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?

1.3.2 Stage One – MLR Study

The MLR is a special form of Systematic Literature Review (SLR) which uses formally peer-reviewed and commercially published literature (called White Literature, e.g., journal articles and conference papers), and unpublished or industrial work (called Grey Literature, e.g., technical reports, websites, blogs, etc.) (Garousi, Felderer, & Mäntylä, 2019).

The reason for using MLR is that DevSecOps is an evolving and topical area in industrial settings, where practitioners are the first to adopt emerging approaches. They constantly produce technical reports, feedback, and reviews, which can serve as an essential supplement to the academic literature. To get the best outcome, both white and grey literature should be included.

Data were extracted from included papers and synthesised by using the Thematic Analysis (TA) method, which is a data analysis approach for identifying, analysing, and reporting the themes with data, combining both qualitative (text segments, codes, themes) and quantitative (frequency statistics) evidence (Braun & Clarke, 2006).

The MLR study identifies five aspects of DevSecOps (Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement), collects related themes for each, and establishes a DevSecOps CPTM Model (Version 1.0) by integrating the themes of the latter four aspects. An unexplored area of applying DevSecOps in GSE has been identified. These findings serve as a theoretical foundation for the subsequent empirical investigation and the entire PhD research.

1.3.3 Stage Two – Delphi-AHP Study

To validate, refine, and improve the DevSecOps CPTM Model (Version 1.0), an empirical investigation was conducted by employing a Delphi survey, which is a data collection method to solicit opinions and feedback from experts in certain domains by performing multiple rounds of surveys, aiming to generate insights and get consensus on controversial subjects for which there is limited information (Beiderbeck et al., 2021). The Delphi method can be used to collect both quantitative and qualitative data (Lilja, Laakso, & Palomäki, 2011), depending on the format used, such as binary questions, rating scales, or open questions (Bastiaansen & Wilderom, 2021).

The Delphi survey was combined with the Analytic Hierarchy Process (AHP), a multi-criteria decision-making approach that structures complex problems hierarchically, prioritises criteria, shows the relationships between criteria and alternatives, and makes the final decision (Brunelli, 2015; Saaty, 2013). Hence, the survey mainly employed closed-ended questions (i.e., pairwise comparisons in AHP format) to collect and analyse quantitative data. Several open-ended questions were added at the end of the questionnaire to collect participants' comments.

In summary, this empirical study combines both positivist and interpretive paradigms (Guba & Lincoln, 1994), employs a qualitative research methodology, and incorporates a quantitative survey, thereby adopting a hybrid approach that utilises a combination of Delphi and Analytic Hierarchy Process (AHP) methods. It evaluates the MLR findings, prioritises the identified DevSecOps challenges, practices, tools, and metrics, and collects new items into each aspect. It also identifies the slight differences between local and global DevSecOps and upgrades the DevSecOps CPTM Model from Version 1.0 to 2.0. Additionally, the dissent analysis reveals that there is disagreement within DevSecOps between the SE industry and academia.

1.4 Key Contributions

This research has made several key contributions that are categorised into three domains proposed by McGrath and Brinberg (1983): Conceptual, Methodological, and Substantive.

Contributions in the Conceptual Domain:

- This research presents a DevSecOps conceptual framework (i.e., the DevSecOps CPTM

Model Version 2.0) that identifies four elements of DevSecOps (Challenges, Practices, Tools, and Metrics), prioritises them, and depicts their interrelationships.

- The overlooked global dimension of DevSecOps in existing literature has been incorporated into the proposed DevSecOps CPTM Model (Version 2.0).
- Researchers can use this model to guide future studies on DevSecOps. It can also serve as a valuable roadmap for practitioners to improve their DevSecOps knowledge and practices.

Contributions in the Methodological Domain:

- This research validates its innovative research design and confirms its relevance and practicality for investigating research gaps in the SE discipline. It employed an MLR study to establish a conceptual framework as a theoretical basis, and subsequently conducted an empirical investigation to validate the MLR findings and refine the conceptual framework.
- The empirical investigation combines positivist and interpretive paradigms, employs a qualitative research methodology, and incorporates a quantitative survey, thereby adopting a hybrid-method approach that utilises the combination of Delphi and Analytic Hierarchy Process (AHP). Pairing Delphi with AHP remains a novel approach in empirical studies and is the first such study to investigate DevSecOps. This research has demonstrated the feasibility of the hybrid-method approach and explained its validity, appropriateness, benefits, and limitations, thereby serving as a helpful reference or illustrative guide for researchers who intend to use similar research methods.

Contributions in the Substantive Domain:

- A Multivocal Literature Review (MLR) protocol is developed and published to ensure MLR's transparency, reproducibility, and objectivity.
- A Multivocal Literature Review (MLR) study on DevSecOps is conducted and published to consolidate, update, and add value to the extant literature.
- This research identifies and prioritises a set of challenges, practices, tools, and metrics of DevSecOps, and depicts their relations by developing a novel conceptual framework, i.e., the DevSecOps CPTM Model, which is built based on the MLR study (Version 1.0), and has

been validated and upgraded through the empirical Delphi-AHP study (Version 2.0).

- A comprehensive list of DevSecOps tools is presented, enabling practitioners to select those needed throughout the DevSecOps lifecycle.
- This research addresses the research gap between DevSecOps and Global Software Engineering (GSE) by investigating differences in DevSecOps across local and global settings.
- This research investigates the consensus and dissent on DevSecOps between the SE industry and academia.

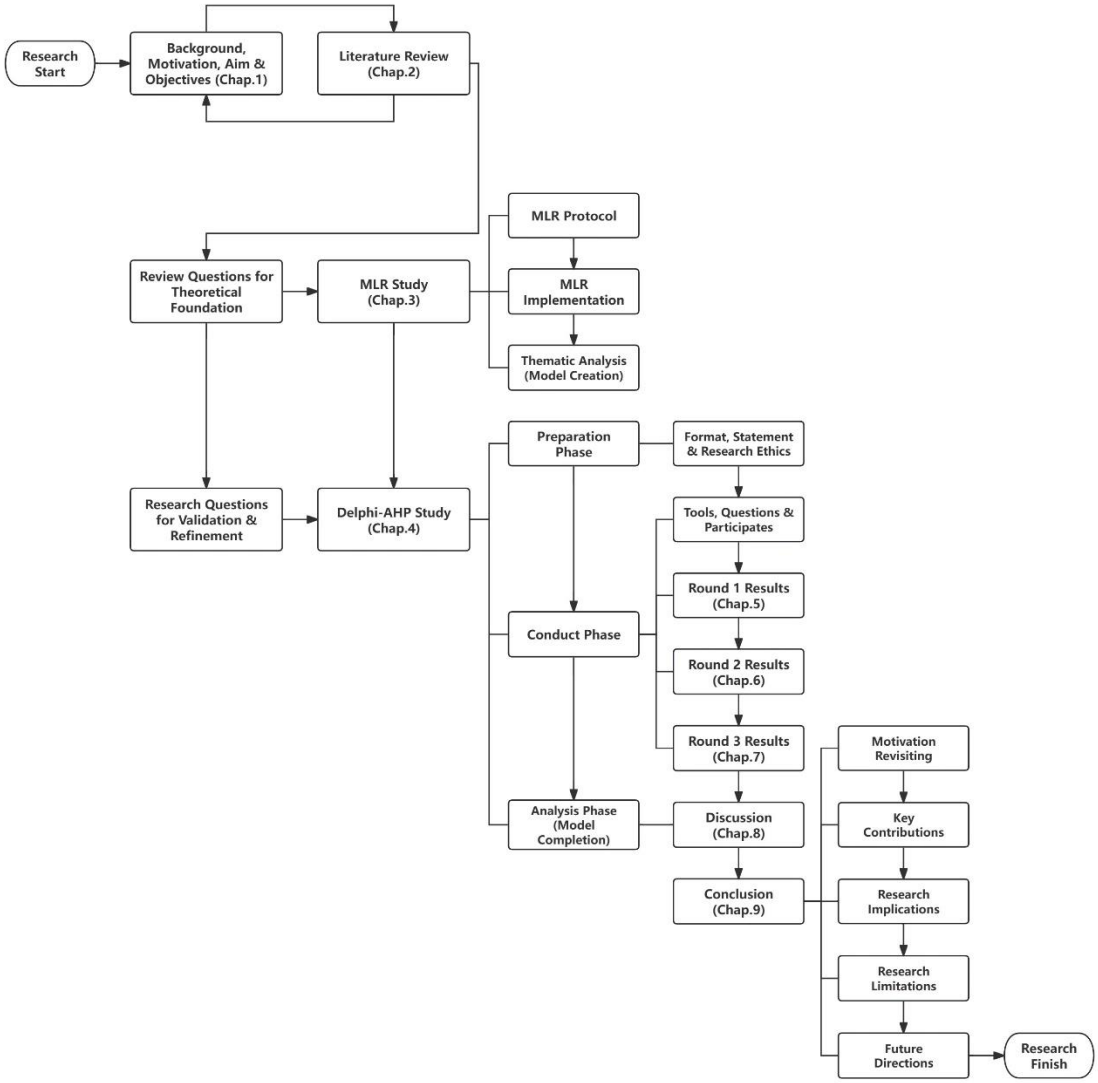
1.5 Publications Related to This Thesis

Zhao, X., Clear, T., & Lal, R., 2024. Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. Journal of Systems and Software, 214, 112063. https://doi.org/10.1016/j.jss.2024.112063.

1.6 Thesis Structure

This thesis is organised into nine chapters. **Figure 1. 1** depicts the entire research process and the thesis structure.

Figure 1. 1 - Research process and thesis structure



Chapter 1 provides background information on conducting this research, offering an understanding of the DevSecOps approach. It presents the research motivation, aim, and objectives, contributions, research design, related publications, and the entire thesis structure.

Chapter 2 introduces some key concepts used in this research and reviews the related work. It also identifies the research gap and indicates the motivation behind conducting a secondary study, i.e., an MLR study on DevSecOps and its application in GSE contexts.

Chapter 3 presents a Multivocal Literature Review (MLR) study on DevSecOps (Zhao, Clear, & Lal, 2024b), which identifies five aspects of DevSecOps research and focuses (Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement); collects related themes of each aspect; and generates a DevSecOps CPTM Model (Version 1.0) by integrating the themes of

the latter four aspects. An unexplored dimension related to the global application of DevSecOps has been identified. The MLR study is a published work (Zhao, Clear, & Lal, 2024b), so this chapter provides only a summary rather than the full article.

Chapter 4 explains the research design in terms of research paradigms, methodology, and methods, and reviews related work based on the research design. Next, the Delphi study process is presented in three phases, i.e., the preparing phase, the conducting phase, and the analysing phase. The AHP method is also discussed, including its three levels, seven steps, sets of equations, and the combination with the Delphi survey.

Chapters 5, 6, and 7 present the results of three rounds of the Delphi survey, and data analysis is conducted using an AHP approach. Through three iterations of the Delphi survey, the identified challenges, practices, and metrics of DevSecOps have been evaluated, revised, and validated.

Chapter 8 addresses the research questions, summarises the findings, and concludes by providing a comprehensive global version of the DevSecOps CPTM Model (Version 2.0).

Chapter 9 concludes this research by revisiting motivation, outlining contributions, providing implications, discussing threats to validity, and suggesting future directions.

2 Chapter 2: Key Concepts and Related Work

The aim of Chapter 2 is to identify the research gap and determine the focus based on the literature review and to state the motivation for DevSecOps investigation. This chapter introduces some key concepts used in this research in Section 2.1. Next, it reviews the related work and identifies the research gap in Section 2.2. Section 2.3 summarises this chapter and outlines the motivation for conducting a secondary study, specifically an MLR study on DevSecOps and its application in GSE contexts.

2.1 Key Concepts and Definitions

This section introduces key concepts and definitions used in this research to provide a consensual understanding of the topic.

2.1.1 Software Engineering (SE), SE Process, Models and Methods

Software Engineering (SE) is defined by Humphrey (1988) as "the disciplined application of engineering, scientific, and mathematical principles and methods to the economical production of quality software". "The software engineering process is the total set of software engineering activities needed to transform a user's requirements into software. This process may include, as appropriate, requirements specification, design, implementation, verification, installation, operational support, and documentation. It also may include either temporary or long-term repair and/or enhancement to meet continuing needs" (Humphrey, 1988).

According to the Software Engineering Body of Knowledge (SWEBOK) V4 (Washizaki, 2024), software engineering models and methods provide structure and systematisation to the software engineering process, enabling repeatability and improving success in software development projects. SE models offer frameworks for problem solving using notations and procedures for constructing and analysing representations. Classic SE models include waterfall, spiral, and iterative (Washizaki, 2024). SE methods provide systematic approaches for specifying, designing, building, testing, and verifying software products and associated deliverables. Examples are agile methods, scaling agile methods, DevOps, DevSecOps, etc (Xu, 2025). They can address a single

development phase or encompass the entire development life cycle (Washizaki, 2024).

2.1.2 DevOps

DevOps is a compound of Development (Dev) and Operations (Ops) (Sebastian et al., 2020). The origin of this concept can be traced back to a famous industrial presentation in 2009, which is titled "10+ Deploys per Day: Dev and Ops Cooperation at Flickr" (Mezak, 2018), presented by two Flickr employees – John Allspaw, vice president of technical operations, and Paul Hammond, director of engineering (Allspaw & Hammond, 2009). They role-played the contentious interplay between representatives of development and operations in their presentation, indicating that Dev focuses on speed while Ops focuses on reliability. They made the case that "the only rational way forward is for application development and operations activities to be seamless, transparent, and fully integrated" (Allspaw & Hammond, 2009). In the same year, the first "DevOps Days" conference was held in Ghent, Belgium, and the term "DevOps" was officially named (Mezak, 2018). The prominence of cloud-native technologies and automation tools empowers DevOps. Its ability to provide shorter lifecycles and more frequent releases enables companies to increase their competitiveness (Washizaki, 2024). After more than a decade of evolution, DevOps has gained popularity and is becoming mainstream in SE (Akbar, Rafi, et al., 2022; Gall & Pigni, 2022; NCCoE, 2025).

The literature on DevOps is abundant and defines DevOps multidimensionally. Thus, some researchers state that there is "a lack of a homogeneous and clear conceptualisation of DevOps" (Gall & Pigni, 2022). On the contrary, Hemon-Hildgen and Rowe (2022) argue that the existing literature presents different definitions because this phenomenon is relatively new, rich, and complex, thereby being defined from a plurality of perspectives. Even so, the core principles and basic common understandings of DevOps tend to remain the same and are embedded in recent definitions (Hemon-Hildgen & Rowe, 2022). Some common themes or components of existing DevOps definitions can be found, such as Culture of Collaboration and Communication, Process of Practices, Automation of Tools/Technologies, Enables/Capabilities, CAMS (Culture, Automation, Measurement, and Sharing), Methodology/Methods, and CI/CD (Continuous Integration/Delivery/Deployment).

Culture of Collaboration and Communication:

DevOps can be defined as a culture (Soni, 2015), aiming to bridge the gaps between developers and operations (Huttermann, 2012), emphasising the collaboration within and between teams involved in the Software Development Lifecycle (SDLC) (Dyck, Penners, & Lichter, 2015; Humble & Molesky, 2011). Rajapakse and Szabo (2024) define the foundational idea of DevOps as reducing functional silos and fostering a collaborative culture. Luz, Pinto and Bonifácio (2018) also suggest that practitioners should foremost focus on building a collaborative culture in DevOps adoption.

Process of Practices:

Bass, Weber and Zhu (2015) define DevOps as a process that is "a set of practices aimed to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality". Gall and Pigni (2022) define DevOps as "a continuous practice in which achievements are constantly challenged with the aim of constantly improving processes, activities and, ultimately, delivery and operation capabilities."

Automation of Tools/Technologies:

Loukides (2012) defines DevOps from a technological perspective and emphasises the automation of software delivery and infrastructure changes through the use of automated tools and technologies.

CAMS and Enables/Capabilities:

Humble and Molesky (2011) define Culture, Automation, Measurement, and Sharing (CAMS) as the four pillars of DevOps. Smeds, Nybom and Porres (2015) define Capabilities, Cultural Enablers, and Technological Enablers as critical elements of DevOps and identify Capabilities as the most important of the three.

Methodology/Methods:

Jabbari et al. (2016), based on the previous definitions, redefine DevOps as a development methodology that "aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance, and delivery with

automated deployment utilizing a set of development practices".

In contrast, Hemon-Hildgen and Rowe (2022) disagree that DevOps constitutes a methodology, because the Merriam-Webster Dictionary defines a methodology as "a body of methods, rules, and postulates employed by a discipline: a particular procedure or set of procedures". They state that "there is no DevOps procedure or process that is systematically applicable to all organisations". Their synthesised definition of DevOps based on literature is "set of a principles for collaborative work implemented between the IS design and development function and the IS operations function, along with potentially other constituencies (stakeholders), which is founded on the sharing of culture, goals, measures, automation tools and automated processes towards continuous delivery of software" (Hemon-Hildgen, Rowe, & Monnier-Senicourt, 2020).

CI/CD:

DevOps is designed to ensure the continuity of the SE flow (Hemon-Hildgen & Rowe, 2022) by combining development, operations, and maintenance to perform continuous integration, delivery, testing, and deployment (Washizaki, 2024). In which case, some literature uses terminologies such as "CI/CD", "Continuous Integration", "Continuous Delivery", and "Continuous Deployment" interchangeably without providing a clear definition. This could be a leading cause of DevOps remaining ill-defined (Gall & Pigni, 2022).

According to the four definitions given by SWEBOK V4 (Washizaki, 2024):

- "Continuous integration (CI) is a software engineering practice that continually merges artifacts, including source code updates from all members of a team, into a shared mainline for evolving and testing the developed system".
- "Continuous delivery (CD) is a software engineering practice that uses automated tools to provide frequent releases of new systems (including software) to staging or various test environments".
- "Continuous testing is a software testing practice that involves testing the software at every stage of the software development life cycle".
- "Continuous deployment (aka CD) is an automated process of deploying changes to production by verifying intended features and validations to reduce risk".

2.1.3 Software Security

A software system is typically specified by its functional and non-functional requirements (Ragab, Ahmed, & AlHashmi, 2015). Functional requirements refer to the observable behaviours that the software is to provide (what the system should do). Non-functional requirements constrain the technologies used in the implementation and the quality of service, including reliability, accuracy, maintainability, and security (how the system should behave) (Washizaki, 2024). Unlike functional requirements, which are intuitively emphasised in software development, non-functional requirements do not receive the same level of concern(Ragab, Ahmed, & AlHashmi, 2015). In particular, security, as a key non-functional requirement, is often overlooked and devalued due to its perceived hindrance to the high velocity required in modern software development approaches, such as agile and DevOps (Ragab, Ahmed, & AlHashmi, 2015).

With software development, security can be divided into two categories: the security of the software being developed; and the security of the software development environment (Morales et al., 2020). Testing and securing software under development is what comes to most people's minds when they think of software security, which refers to the detection and removal of vulnerabilities and faulty code. Some examples of software security testing include: Software Composition Analysis (SCA), Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Interactive Application Security Testing (IAST) (Morales et al., 2020).

The other type of security emphasises safeguarding the development environment and the overall software factory against security threats, which encompasses various components, including infrastructure, tools, datacentres, stakeholder mindset, management and governance, and business considerations (DoD, 2025; Morales et al., 2020). Some typical security threats in this category include: unauthorised data access, inappropriate privilege management, immature monitoring and alerting, server hijacking, data leakage, data tampering, etc. (Morales et al., 2020). Furthermore, Jiang (2021) defines the cybersecurity domain as including security architecture, security engineering, network security, physical security, cloud security, software/application security, data protection, security governance, policies, laws and regulations, etc.

2.1.4 DevSecOps

With the adoption of DevOps, security requirements ought to be considered at the forefront, due to cloud as a driver for business application deployment and the need for DevOps to be supported by emerging tools and techniques, such as cloud-native technologies, microservice architectures, serverless frameworks, and artificial intelligence (AI) tools and capabilities (NCCoE, 2025). Those emerging technologies also bring security risks and complications to the software development process (Fernandez & Brito, 2019).

The importance of security considerations has led to the emergence of DevSecOps, which is a security-oriented expansion of DevOps (Morales et al., 2020). Mohan and Othmane (2016) define DevSecOps as ‘‘incorporating security practices in the DevOps processes by promoting collaboration between development, operations, and security teams’’ (Mohan & Othmane, 2016).

According to Myrbakken and Colomo-Palacios (2017), “DevSecOps is seen as a necessary expansion to DevOps, where the purpose is to integrate security controls and processes into the DevOps software development cycle and that it is done by promoting the collaboration between security teams, development teams and operations teams.”

Rahman and Williams (2016) define DevSecOps as “the process of integrating secure development best practices and methodologies into development and deployment processes which DevOps makes possible”.

Similar to the multidimensional conceptualisation of DevOps, there is ongoing debate in the literature over how to define DevSecOps, and no homogeneous definition exists. However, its basic common understandings and core principles tend to remain the same. Evidences support that the conceptual difference of those definitions is more apparent than real, and there is a convergence in definitions. Different perspectives and ways to define DevOps have led to various taxonomies of DevSecOps, which have been explained previously in Section 2.1.2, and later in Section 3.3.1.2, Chapter 3 (Page 46), as well.

2.1.5 Global Software Engineering

In a global economy, an increasing number of SE companies are adopting a globally distributed SE approach (Whitehead et al., 2010). Global Software Engineering (GSE), also known as Global

Software Development (GSD), is a business strategy that involves geographically distributed software development teams across different countries (Grande, Vizcaino, & Garcia, 2024). It provides global access to specialised and diverse skilled human resources and enables reductions in software development costs and time-to-market by leveraging time-zone effectiveness and round-the-clock productivity (Conchuir et al., 2009; Vizcaíno et al., 2016).

GSE depends on the distributed teams comprising stakeholders from different geographic locations, different time zones, and even different organisational and national cultures. It therefore has to face the challenges from geographical, temporal, linguistic, and cultural distances (Jalali, Gencel, & Šmite, 2010; Tamburri et al., 2012).

Additionally, the COVID-19 pandemic marked a turning point in the way software is developed. Many software companies experiment with the "hybrid work", which is a spectrum of flexible work arrangements, where employees' work locations and/or hours are not strictly standardised. Hybrid work is also used to refer to distributed teams consisting of co-located and remote members in GSE/GSD contexts (Smite et al., 2023).

2.1.6 Collaborative Software Engineering (CoSE)

Collaboration is pervasive throughout software engineering (Constantino et al., 2020; Whitehead et al., 2010), and the software engineering process is inherently cooperative and collaborative, requiring many roles and stakeholders to work together, share artifacts, coordinate activities, and communicate to produce large and complex software systems (Omoronyia et al., 2010; Whitehead et al., 2010). Thus, Whitehead et al. (2010) define that Collaborative Software Engineering (CoSE) is ''about creating the organisational structures, reward structures, and work breakdown structures that afford effective work towards a goal''.

CoSE is closely tied to the 3C Collaboration model, which defines collaboration as integration of communication, coordination, and cooperation (Conchuir et al., 2009; Franzago et al., 2018; Steinmacher, Chaves, & Gerosa, 2010; Tamburri et al., 2012). The 3C Collaboration model is not only about software development team members but also embraces external, non-technical stakeholders, such as customers and end users (Davis & Daniels, 2016).

CoSE empowers the prominence of outsourcing, open-source software projects, distributed agile

methods, and global software engineering processes (Franzago et al., 2018). Accordingly, DevOps and DevSecOps, as contemporary software development paradigms within the GSE/GSD context, are grounded in the foundational principles of CoSE. They deal with methods, processes and tools for enhancing collaboration (i.e., the union of communication, coordination, and cooperation) among roles and stakeholders (Conchuir et al., 2009; Franzago et al., 2018; Steinmacher, Chaves, & Gerosa, 2010; Tamburri et al., 2012). The foundational idea of DevOps and DevSecOps is to reduce functional silos and foster collaboration among Development, Operations, and Security teams (Rajapakse & Szabo, 2024). Luz, Pinto and Bonifácio (2018) emphasise that practitioners should focus on building a collaborative culture in DevOps adoption, which also enables collective decision-making about adopting tools and automation for the DevOps environment. This is also critical for DevSecOps adoption, as it enables fluid boundaries between various functional setups involved in developing and delivering software for the production environment. Hence, a conceptual model that guides DevSecOps adoption helps determine SE practices, tools, and development infrastructure.

In GSE/GSD, geographical, temporal, linguistic, and cultural distances make the achievement of the 3C particularly difficult (Steinmacher, Chaves, & Gerosa, 2010). Developing a shared understanding is more challenging because product ideas are dynamic and spontaneous, and the geographical distribution of teams limits rapid communication. Additionally, different national and organisational cultures may amplify the challenge of communication (Whitehead et al., 2010). Therefore, the 3C implementation may encounter magnified challenges when adopting DevOps and DevSecOps in GSE contexts. Evidences state that both academia and industry are increasingly researching and adopting DevOps practices in the GSE context (Cico et al., 2021; Grande, Vizcaino, & Garcia, 2024). As a security-oriented expansion of DevOps, DevSecOps in GSE also warrants an academic investigation.

2.2 Literature Review

This section presents a brief literature review, which was preliminarily conducted to capture the existing related work and identify the research gap.

2.2.1 Related Work on DevSecOps

Several secondary studies have explored DevSecOps, identifying its definitions, benefits, challenges, practices, tools, metrics, and applications. A summary and comparison among the existing secondary studies are presented in **Table 2. 1**, including three Multivocal Literature Review (MLR) studies (Akbar, Smolander, et al., 2022; Myrbakken & Colomo-Palacios, 2017; Prates et al., 2019), two Systematic Literature Reviews (SLR) studies (Rajapakse et al., 2022; Sanchez-Gordon & Colomo-Palacios, 2020), a Grey Literature Review (GLR) study (Mao et al., 2020), and a Systematic Mapping Study (SMS) (Mohan & Othmane, 2016). By reviewing those studies chronologically, the development and evolution of DevSecOps research have been discussed, and the corresponding research gap has been identified.

Table 2. 1 - Comparison among review papers

Reference	Research methods	Search sources	Included studies	Aspects involved
(Mohan & Othmane, 2016)	Mapping study	Google Scholar, IEEE, OWASP, and RSA conferences	5 WL + 3 Presentations	Definition, Practices, Compliance, Automation, Tools, Configuration management, Team collaboration, Availability of activity data, Information secrecy
(Myrbakken & Colomo-Palacios, 2017)	MLR	Google Scholar, Google	2 WL + 50 GL	Definitions, Characteristics, Benefits, Challenges, Evolution
(Prates et al., 2019)	MLR	ACM, IEEE, Scopus, Google Scholar, Google	2 WL + 11 GL	Metrics
(Sanchez-Gordon & Colomo-Palacios, 2020)	SLR	Google Scholar	11 WL	Cultural aspects
(Mao et al., 2020)	GLR	Google	141 GL	Security risks, Practices
(Akbar, Smolander, et al., 2022)	MLR + Survey	ACM, IEEE,	46 WL + 41 GL	Challenges

		Wiley, Springer Link, Science Direct, Google Scholar, Google		
(Rajapakse et al., 2022)	SLR + TA	ACM, IEEE	54 WL	Challenges, Solutions
Ours (Zhao, Clear, & Lal, 2024b) (also presented in Chapter 3)	MLR + TA + Survey	ACM, IEEE, Scopus, Google	104 WL + 43 GL	Definitions, Challenges, Practices, Tools, Metrics, Global applications

Early research on DevSecOps was relatively sparse. Mohan and Othmane (2016) conducted a mapping study to capture multiple aspects of DevSecOps, including the definition, practices, compliance requirements, automation, tools, configuration management, team collaboration, availability of activity data, and information secrecy. This study is considered the earliest secondary investigation in the DevSecOps domain. However, it is based on a limited body of work (five academic papers and three presentations), which constrains the generalizability of its findings. The findings are relatively limited, making it challenging to draw upon them to shape SE environments and practices for security-driven development.

A subsequent study by Myrbakken and Colomo-Palacios (2017) expanded the investigation of DevSecOps by conducting a Multivocal Literature Review (MLR) study to identify its definition, characteristics, benefits, challenges, and evolution. This study was undertaken when DevSecOps was emerging as a novel approach to support the adoption of CI/CD practices. Therefore, the study was limited by the scarcity of academic research publications as data sources, relying on only two academic papers. However, the fifty grey literature items included in that study broaden the findings and demonstrate the practical adoption of DevSecOps, providing valuable insights, despite the absence of a fully balanced perspective between academic and industry evidence. Similar to the previous study by Mohan and Othmane (2016), this study's findings remain insufficient to thoroughly guide the adoption of SE practices for capturing and implementing security requirements.

Subsequently, Prates et al. (2019) conducted an MLR study to identify nine DevSecOps metrics from two academic papers and eleven grey literature articles. Compared with the two reviews by

Mohan and Othmane (2016) and Myrbakken and Colomo-Palacios (2017), this study focuses on a specific aspect of DevSecOps – Measuring and Metrics, and the findings make significant contributions to this unexplored DevSecOps dimension.

Nonetheless, both MLR studies (Myrbakken & Colomo-Palacios, 2017; Prates et al., 2019) state a limitation in common: the selection of the MLR method to include grey literature due to the lack of academic studies at that time. However, employing the MLR method nowadays is essential to incorporate practitioner perspectives, so that the MLR method is an active choice, rather than seen as a compromise. Compared to Systematic Literature Review (SLR) and Systematic Mapping Study (SMS), the MLR method covers both the researcher-oriented and practitioner-oriented sources to provide a reliable and valuable knowledge of SE approaches from dual perspectives for the application in practice. Importantly, SE approaches such as DevOps and DevSecOps originate from practices, so industry views cannot be ignored.

Since 2020, the DevSecOps domain has seen an increasing number of both white and grey literature, leading to more specific and in-depth secondary studies being undertaken compared to the earlier work. Sanchez-Gordon and Colomo-Palacios (2020) conducted an SLR study to review DevSecOps from a cultural and human-factors perspective. Their findings state the culture as a crucial component of DevSecOps, and characterise it using themes such as collaboration, communication, knowledge sharing, feedback, continuous improvement, responsibility, trust, experimentation, leadership, commitment, agreement, blamelessness, transparency, and new personnel hiring. DevSecOps culture offers a different way of working, which emphasises cross-team collaboration with a focus on security. Rafi et al. (2020) presented an SLR study to identify eighteen DevSecOps challenges and surveyed experts to prioritise the findings. The study helps practitioners focus on key challenges that require further improvement to secure DevOps. A Grey Literature Review (GLR) by Mao et al. (2020) identified DevSecOps practices from the industrial source, and categorised the findings in terms of Process, Infrastructure, and Collaboration. In comparison with the two SLRs by Sanchez-Gordon and Colomo-Palacios (2020) and Rafi et al. (2020), that GLR provides a purely practitioner-oriented review, showing the state-of-the-practice of DevSecOps in industry.

Those studies in 2020 (Mao et al., 2020; Rafi et al., 2020; Sanchez-Gordon & Colomo-Palacios,

2020) have made respective contributions to a single viewpoint or a specific aspect of DevSecOps. They need to be combined to give a collective view and a comprehensive understanding of the field. In comparison, this research regards DevSecOps as a broad and multifaceted concept, thereby aiming to conduct a comprehensive secondary study, which covers the entire software development life cycle rather than a specific step, and takes inspiration from multiple perspectives, such as academic, industrial, cultural, organisational, technological, managerial, and business.

With the rapid development of DevSecOps, research has been exploring the interrelation of multiple DevSecOps aspects to provide a comprehensive understanding. Rajapakse et al. (2022) conducted an SLR study to identify a set of DevSecOps challenges and corresponding solutions by applying thematic analysis, further classifying them in four categories: People, Practices, Tools, and Infrastructure. Akbar, Smolander, et al. (2022) presented an MLR study and employed thematic analysis to reveal eighteen DevSecOps challenges and grouped them in ten categories. According to Cruzes and Dyba (2011a), there are four levels of interpretation in Thematic Analysis (TA): Text, Code, Themes, and Model. These two review papers (Akbar, Smolander, et al., 2022; Rajapakse et al., 2022) have contributed to the correspondence between the challenges and practices of DevSecOps, indicating what practices and tools can be adopted to overcome the corresponding challenges. However, they both stopped at the third level of TA interpretation – Themes; the final level – Model, had not been reached.

Overall, the findings of earlier secondary studies are basic, premature, and unshaped due to the scarcity of data sources in this emerging area, with most focusing on a single perspective or aspect of DevSecOps. With the rapid development of DevSecOps, the research trends have become distinct and taken shape. The majority of studies focus on the challenges and practices of DevSecOps adoption. The recent trend is to explore the interrelation of challenges, practices, and tools, further to build theoretical frameworks for DevSecOps. However, a substantial body of academic research on the topic has not been fully built, and the existing literature lacks comprehensive guidelines and reference models for DevSecOps adoption.

2.2.2 Related Work on DevSecOps/DevOps in GSE

To date, no academic studies have explicitly investigated the adoption of DevSecOps in the Global Software Engineering (GSE) context. However, several studies have investigated global DevOps

practices. Therefore, an investigation into DevSecOps within the GSE context is necessary to explore how security requirements and considerations can be effectively integrated into globally distributed development environments.

Diel, Marczak and Cruzes (2016) employed exploratory observations and interviews to identify seven themes of communication challenges and communication strategies in distributed DevOps. The identified challenges were categorised into three factors: Geographical distance, Socio-cultural distance, and Temporal distance, while strategies were grouped into four facets: Frequency, Direction, Modality, and Content. That paper presents how two-way communication happens in global DevOps, and states that communication issues are a recurrent problem. Some recommendations are provided for better communication in global DevOps, for instance, creating exchange programs between teams to overcome geographical distance, getting teams more involved with others' cultures to mitigate cultural issues, and promoting face-to-face meetings quarterly to address temporal distance.

Hussain, Clear and MacDonell (2017) presented the results from an investigation based on online job advertisements and interviews. The investigation identified the Knowledge areas, Skills, and Capabilities (KSCs) for DevOps roles in New Zealand. It also revealed that the global dimension of DevOps roles was apparent in most job advertisements, sometimes by explicit mention (16% job postings explicitly mentioned global aspects) but more often by implication, reflecting that global DevOps was involved in the New Zealand SE industry, but not to any great extent. However, none of the identified KSCs mentioned security aspects. That paper presented the practical requirements and accountabilities of DevOps roles, which were valuable to employers, job seekers, researchers, educators, and policy makers at that time (2017). On the contrary, at present, security has become one of the indispensable KSCs for DevOps roles (Puppet, 2025).

Gupta, Venkatachalapathy and Jeberla (2019) presented an empirical study on a global DevOps project across India, the USA, and Germany, that had successfully established continuous delivery and short release cycles with agile. The authors brought their practical insights from experiences as a Project Manager, a Quality Manager, and an Architect, who was an integral part of their project. That paper provides actual challenges and practices of global DevOps, and presents valuable lessons learned and recommendations from the perspectives of different stakeholders,

i.e., project manager, quality manager, and architect. The scarce information about security is only mentioned by the author, who is an architect of their global DevOps project, pointing out the importance of security in the cloud environment, and suggesting using microservice architecture to help in updating security patches frequently.

Grande, Vizcaino and Garcia (2024) recently presented a Systematic Mapping Study (SMS) on the application of DevOps in Global Software Development (GSD). The paper reviewed the definition of DevOps in global settings; captured the goals of adopting DevOps in GSD along with five motivating issues; identified eleven benefits, nine challenges, and fifteen practices of DevOps in GSD; mapped the identified challenges with a list of well-known GSD risks; also mapped the links among the motivating issues, benefits, challenges, and practices. The study has contributed to a review of the current status of DevOps adoption in GSD. It has identified several research gaps, including the lack of a security-related framework for DevOps in GSE. That study helps practitioners evaluate the challenges and benefits of implementing DevOps practices in GSE contexts, providing them with sets of practices and tools that can be employed in conjunction with those currently used.

From a practical viewpoint, scaling agile development has become a cornerstone in managing large, diverse, and distributed development teams (Ononiwu et al., 2023). The adoption of scaling agile frameworks such as Disciplined Agile® Delivery (DAD) and Scaled Agile Framework (SAFe®) enables organisations to coordinate activities of multiple teams cooperating to produce a large software product (Beecham et al., 2021; Ononiwu et al., 2023; Senapathi & Strode, 2025). DevOps, as an important part of agile product delivery competency, has been hybridised with such frameworks, e.g., the Disciplined DevOps in the DAD (PMI, 2024) and DevOps series in the SAFe (SAFe, 2025). Security aspects of DevOps are also involved in those frameworks. For example, Disciplined Agile® Enterprise (DAE) presents a Disciplined DevOps process, comprising six blades: Disciplined Agile® Delivery (DAD), Security, Data management, Release management, Support, and IT operations. Although several security strategies are provided to support the Disciplined DevOps (PMI, 2025), there is no model for DevSecOps adoption. The SAFe® framework focuses on security, compliance, and privacy requirements, which do not have strong intersections with its DevOps series, though (Alsaqaf, Daneval, & Anish, 2021; SAFe, 2025).

By reviewing the above related work, it becomes apparent that the security aspect of DevOps in the GSE/GSD context is scarcely mentioned and addressed. Hence, the research gap between DevSecOps and GSE has been revealed. Although some existing industrial frameworks guide organisations through scaling agile development with DevOps, they lack reference models and comprehensive guidelines for DevSecOps adoption. As an expansion of DevOps, the adoption of DevSecOps in GSE requires an academic investigation.

2.3 Chapter Summary

To sum up, the related work demonstrated that DevOps, DevSecOps, and GSE are growing areas of interest to both the SE academia and industry. However, the existing secondary studies are relatively early or focus on a single perspective or aspect of DevSecOps, and little is known about its application in the GSE context. Hence, a research gap is identified: a substantial body of academic research on DevSecOps has not been entirely established, and the existing literature remains scarce in guidelines or reference models for DevSecOps adoption in the GSE context.

Given all this, the following Chapter 3 presents a Multi-vocal Literature Review (MLR) study on DevSecOps and its adoption in GSE to identify, review, and analyse the current state of the art, and further consolidate, update, and add value to the existing literature on this research direction. The MLR study aimed to provide a comprehensive review of the DevSecOps research and development over the decade (2012–2022), reporting new findings from academia and industry for this fast-moving field and building a conceptual framework for DevSecOps adoption based on those findings.

Hence, the following research/review objectives and questions of the MLR study were established. They would be addressed in the next Chapter.

Research/Review Objectives:

- To explore the current state of DevSecOps in the existing white and grey literature.
- To explore the adoption of DevSecOps in GSE from the existing white and grey literature.
- To establish a conceptual framework of DevSecOps based on the existing literature.

Research/Review Questions:

- ***RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect, and their links) in the existing white and grey literature?***

  Sub-question 1.1: What aspects of DevSecOps can be found in the existing white and grey literature?

  Sub-question 1.2: What themes do these aspects contain?

  Sub-question 1.3: How do the identified aspects and themes link to each other?

- ***RQ2: How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?***

3 Chapter 3: Multivocal Literature Review

The aim of Chapter 3 is to provide a comprehensive review of the DevSecOps research and development over the past decade, reporting new findings from academia and industry for this fast-moving field, and building a conceptual framework for DevSecOps adoption based on these findings. It presents a summary of a Multivocal Literature Review (MLR) study on DevSecOps and its application in GSE, encompassing both white (104 studies) and grey (43 studies) literature from 2012 to 2022. The Thematic Analysis (TA) approach was employed to identify, synthesise, and analyse the themes within the data for reporting the MLR results.

This MLR work has been published as: Zhao, X., Clear, T., & Lal, R. (2024). Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. Journal of Systems and Software, 214, 112063. https://doi.org/10.1016/j.jss.2024.112063. I acknowledge that I reformatted the published journal article for this chapter and rewrote certain paragraphs for several other chapters of the thesis.

In addition to the published journal article, the associated materials are available in an open repository at zenodo.org: Zhao, X., Clear, T., & Lal, R. (2024). Identifying the Primary Dimensions of DevSecOps: A Multi-vocal Literature Review, https://doi.org/10.5281/zenodo.10668696. The associated materials include:

- MLR protocol
- List of included papers along with quality assessment score
- Raw data/text and codes (definitions, challenges, and practices)
- Thematic synthesis for white and grey literature
- Thematic analysis tables (first edition)
- Thematic analysis tables (completed edition)
- A conceptual framework named “DevSecOps CPTM Model” (Version 1.0)

These associated materials are also provided in the Appendices section of the thesis.

In Section 3.1, the review method (i.e., Multivocal Literature Review) is introduced. Next, the development of the review protocol and the implementation of the MLR are presented in Section 3.2. Section 3.3 presents the findings of the MLR study. It identifies five aspects of DevSecOps (Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement), collects related themes for each, and generates a conceptual framework named "Challenge-Practice-Tool-Metric (CPTM) Model (Version 1.0)" by integrating the themes of the latter four aspects. Additionally, an unexplored research dimension related to the global application of DevSecOps has been identified. Section 3.4 finally concludes the chapter. The DevSecOps CPTM Model (Version 1.0) is regarded as the key outcome of this MLR study. It reveals the current state of DevSecOps and establishes a basis for the empirical study presented in Chapter 4.

3.1 Review Method – Multi-vocal Literature Review

A Multi-vocal Literature Review (MLR) was selected for the secondary study in this doctoral research, which aimed to present a comprehensive review of DevSecOps over the decade by reporting new findings from academia and industry in this fast-moving field and building a conceptual framework for DevSecOps adoption based on those findings. By doing so, a solid foundation could be built for the follow-up empirical investigation.

3.1.1 Multi-vocal Literature Review (MLR)

A Multi-vocal Literature Review (MLR) is a special form of the Systematic Literature Review (SLR). It not only uses the formally peer-reviewed and commercially published literature (called white literature, e.g., journal and conference papers) but also includes the unpublished literature work (called grey literature, e.g., industrial articles, technical reports, websites, magazines, blogs, etc.) (Garousi, Felderer, & Mäntylä, 2019).

Schopfel (2010), who is member of GreyNet, defines that "Grey literature stands for manifold document types produced on all levels of government, academia, business and industry in print and electronic formats that are protected by intellectual property rights, of sufficient quality to be collected and preserved by libraries and institutional repositories, but not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body".

There is no need to define white literature (WL) as it is a relative concept to GL.

The distinction between white and grey literature has not been clearly recognised by researchers though, the most generally accepted boundary is whether it is controlled by commercial publishers, rather than whether it is academic or industrial work (Farace & Schopfel, 2010; Schopfel, 2010). Books can be either WL or GL, depending on their types. Doctoral/master's theses are sometimes classified as WL because they are considered academic artifacts. Whereas, according to the GL definition, they belong to GL, as they are not peer-reviewed or controlled by commercial publishers. In this MLR, no theses were actually selected because they are not included in commercial publishers' databases, such as ACM, IEEE, and Scopus. A few theses were identified through snowballing, but were eventually excluded, because they did not meet the criteria of paper selection or quality assessment and were deemed unsuitable or unworthy for this research.

3.1.2 Motivation Behind Conducting MLR

The secondary study in this PhD research aimed to present a comprehensive review of the DevSecOps research and development over the decade, report new findings from academia and industry for this fast-moving field, and build a conceptual framework for DevSecOps adoption in the context of Global Software Engineering (GSE). To achieve this, the research objectives and research questions were established, as stated at the end of Chapter 2 (on Page 26). This was followed by an MLR study based on the review protocol, which led to the development of an initial DevSecOps conceptual framework, i.e., the DevSecOps CPTM Model (Version 1.0). Then, an empirical investigation was undertaken through three rounds of a Delphi-AHP study to evaluate and validate the MLR findings and further refine and upgrade the DevSecOps CPTM Model from Version 1.0 to 2.0 by incorporating additional GSE aspects.

Besides MLR, Systematic Literature Review (SLR) and Systematic Mapping Study (SMS) were also evaluated for the suitability for this PhD research. SLR is a means of identifying, evaluating, and interpreting all available research relevant to a particular research question, or research area, or phenomenon of interest (Kitchenham, 2007). This method is widely used and contributes to the academic viewpoint of DevSecOps, based solely on white literature, e.g., (Rajapakse et al., 2022; Sanchez-Gordon & Colomo-Palacios, 2020).

An SLR of academic literature on DevSecOps required developing and following a protocol to

conduct the review, which enabled the identification, evaluation, synthesis, and presentation of findings, and provided transparency and rigour in the identification, selection, and review of the DevSecOps literature. This was crucial in developing the initial conceptual framework. However, a conceptual framework derived solely from academic literature would fail to capture the practical realities. DevSecOps is an emerging and rapidly evolving field (10 – 15 years) primarily driven by industry needs. Practitioners from industry are the first to adopt emerging approaches and constantly produce industrial articles, technical reports, feedback, and reviews, which serve as an important supplement to the academic literature. This broader perspective, achieved through the integration of both white and grey literature, formed the foundation of the MLR study.

A Systematic Mapping Study (SMS) is a means of drawing a "map" of a research area by classifying papers and results into relevant categories and counting the frequency of work within each category, aimed at laying a foundation for future research and educating the community. It applies to more emerging research areas (less than 10 years old) with limited systematic review publications, and it is inapplicable for answering specific research questions (Felderer & Carver, 2017). For example, Mohan and Othmane (2016) presented an SMS, which is considered the earliest secondary study on DevSecOps. The generalizability of its findings was constrained by the scarcity of relevant publications in 2016, when DevSecOps was emerging as a novel approach. Since 2020, the DevSecOps domain has seen an increasing number of both white and grey literature. Hence, SMS was considered unsuitable for this research, as the aim was to go beyond identifying and classifying findings based solely on academic publications on DevSecOps to develop a conceptual framework grounded in an in-depth understanding of theoretical propositions and industry practices.

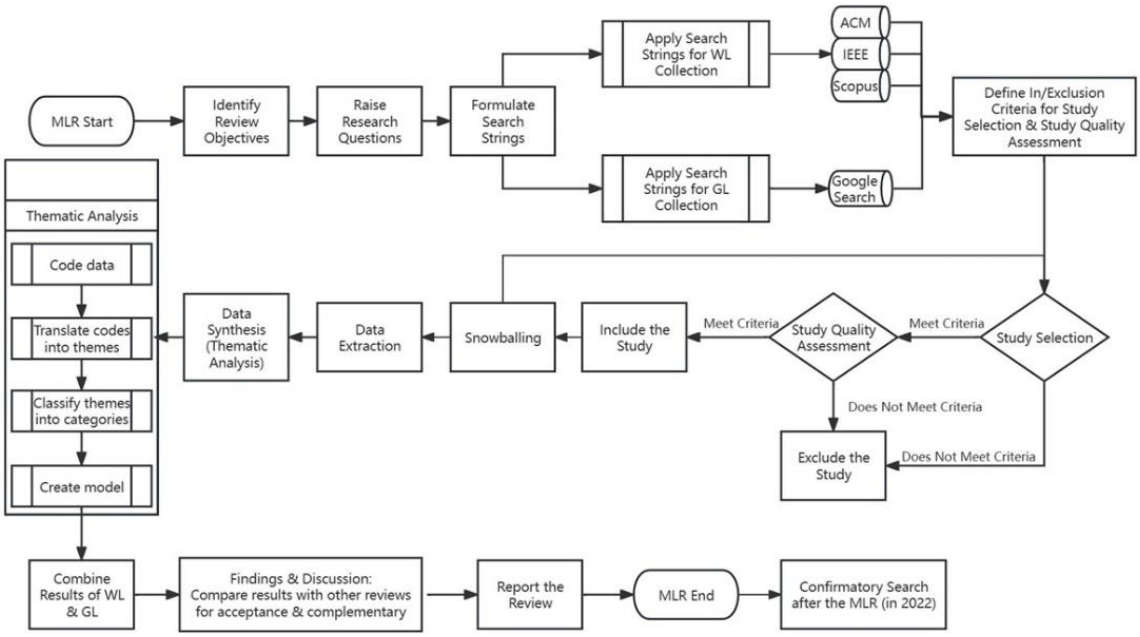
Unlike early MLR studies on DevSecOps (Myrbakken & Colomo-Palacios, 2017; Prates et al., 2019), which included grey literature due to the lack of relevant white literature, using MLR was an active choice for this research rather than a compromise. The MLR method encompasses both researcher-oriented and practitioner-oriented sources, providing a reliable and valuable understanding of SE approaches from dual perspectives for practical application.

3.1.3 MLR Process and Philosophical Stance

The research design of the MLR study incorporates both positivist and interpretive paradigms,

employing a pragmatic strategy as recommended by Hoda (2021) and Cruzes and Dyba (2010), which is driven by pragmatic demands around predetermined information needs. "Although combined approaches are methodologically challenging" (Cruzes & Dyba, 2010). **Figure 3. 1** illustrates an overview of the MLR process, adapted from the guidelines presented by Garousi, Felderer and Mäntylä (2019), which also incorporates the Thematic Analysis (TA) method (Cruzes & Dyba, 2011a) for data synthesis and results reporting.

Figure 3. 1 - Overview of the MLR process



In **Figure 3. 1**, earlier steps of the MLR, from beginning to data extraction, belong more to the positivist perspective (Kitchenham, Dyba, & Jorgensen, 2004), which aims to derive an objective reality from pure data by using quantitative analysis, without undue influence of researchers' interpretation (Alharahsheh & Pius, 2020). Although biases in study selection, quality assessment, and data extraction were inevitable, they could be mitigated by the rigorous formulation and strict implementation of the review protocol. A literature review considers synthesis and interpretation to be mandatory (Rowe, 2014). Thus, steps since data synthesis can be addressed through an interpretivist stance, which advocates a relativist ontology and a subjective epistemology, and employs qualitative methods (Alharahsheh & Pius, 2020). In this MLR, the white and grey literature were collected in strict accordance with the search strategy, selection criteria, and quality assessment criteria. The reflexive Thematic Analysis was employed as the data synthesis method

to qualitatively analyse the extracted data through coding, theming, and modelling (Braun & Clarke, 2021).

3.2 Protocol Development and MLR Implementation

Prior to undertaking the MLR, the process illustrated in **Figure 3. 1** was applied to structure a review protocol, which defined the procedures, specified the methods, and described the conduct of the proposed MLR (Garousi, Felderer, & Mäntylä, 2019; Kitchenham, 2007). Although the protocol development was the planning stage of the MLR study, it covered most activities in **Figure 3. 1** and involved iterations, e.g., the selection of search sources, the formulation of search strings, selection criteria and quality assessment for primary studies, the data extraction form, and the data synthesis method. These activities were initiated during protocol development and refined when the MLR was properly conducted (Kitchenham, 2007).

Based on the drafted protocol, a pilot MLR round with a small number of papers was conducted throughout the process, including WL and GL search, paper selection, quality assessment, data extraction, and a little data synthesis. Furthermore, the drafted protocol was presented to supervisors for review and criticism. By doing so, the MLR protocol was iteratively tested, evaluated, modified, and updated over the research timeline to ensure it remained relevant.

Several improvements were made:

- The selection and combination of databases were adjusted.
- Additional keywords were embedded into search strings.
- Snowballing searches were added to locate relevant papers
- A continuous confirmatory process named “Confirmatory Search” was supplemented after the MLR to identify the latest literature, thereby avoiding staleness and continuing validation.

Once the protocol development had been completed, the MLR process was ready to be implemented. The latest version of the MLR protocol has been published at zenodo.org (https://doi.org/10.5281/zenodo.7959584), including the data extraction form and the database search records. It is also provided in Appendix B.1, “MLR Protocol” (Page 295).

3.2.1 Research/Review Objectives and Questions

The research/review objectives and questions of the MLR study were established based on the previous literature review presented in Chapter 2.

Research/Review Objectives:

- To explore the current state of DevSecOps in the existing white and grey literature.
- To explore the adoption of DevSecOps in GSE from the existing white and grey literature.
- To establish a conceptual framework of DevSecOps based on the existing literature.

Research/Review Questions:

- ***RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect, and their links) in the existing white and grey literature?***

  Sub-question 1.1: What aspects of DevSecOps can be found in the existing white and grey literature?

  Sub-question 1.2: What themes do these aspects contain?

  Sub-question 1.3: How do the identified aspects and themes link to each other?

- ***RQ2: How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?***

3.2.2 Search Strategy

Based on the research questions, the search strategy was defined, discussed, and refined, including the selection of search sources and the formulation of search strings.

Search Sources:

- Three digital databases with advanced or constructed search features were used for the White Literature (WL) collection: ACM Digital Library, IEEE Xplore, and Scopus.

  The combination of these databases was considered to comprehensively cover publications in the fields of computing and software engineering. Scopus and ScienceDirect are both owned by Elsevier and somewhat overlap. ScienceDirect was not selected as it is more suitable for searching journal papers on specific topics. The initial attempt to search using

ScienceDirect did not yield sufficient results. Hence, Scopus was selected, as it is the largest abstract and citation database, providing a wide range of literature from multiple publishers. Similarly, Springer was not selected as it did not produce sufficient search results.

- Google search was used to identify and collect the Grey Literature (GL).

Search Strings:

- ***Search String 1 = (devops AND (security OR secure OR safe)) OR secdevops OR devsecops.***

  'DevSecOps', 'SecDevOps', and 'DevOps' were embedded in the search string. Keywords 'Secure' and 'Safe' were not enclosed in double-quotes, so their variations, such as 'Securely', 'Safely' and 'Safety', were searched too.

  Some dictionaries define security in terms of safety, and vice versa. Languages such as Chinese, Germanic, and Romance do not even distinguish them (Burns, McDermid, & Dobson, 1992). Although the definitions of 'Security/Secure' and 'Safety/Safe' share many similarities, they are not identical but rather complementary. 'Safety' focuses on the undesirable effects that are unintentional; 'Security' focuses on the undesirable effects that are caused by malicious parties outside the system (Line & Rostad, 2006). Burns, McDermid and Dobson (1992) distinguish them by causalities and failure consequences: a system is "safety-critical" if failure could immediately and directly cause absolute harm; a system is "security-critical" if failure could only cause relative harm, or could raise the possibility of harm. Thus, the connotations of security and safety are not separate. Security flaws compromise safety, while safety breaches make security impossible. Considering the above, 'Safe/Safety' was embedded in the search string.

- ***Search String 2 = (devops AND (security OR secure OR safe) OR secdevops OR devsecops) AND ("global software engineering" OR "global software development" OR gse OR gsd OR "globally distributed" OR "distributed software development" OR "distributed software engineering" OR "multi-site" OR "multi-nation" OR "transnational" OR "remote work").***

  After applying Search String 1, the results did not include any studies involving the adoption of DevSecOps in GSE settings, so Search String 2 was added to narrow the search to a global

orientation by specifying additional keywords related to GSE, because Search String 1 retrieved hundreds of papers from each database, which potentially led to accidentally missing a few GSE-related papers.

Search strings needed to be adapted due to differences between databases and the acceptability of Boolean operators. Search results were sorted by relevance, enabling the researcher to know when the relevance was too weak to stop the search. Search strings were also applied to Google Search for the GL collection. After eliminating advertising, the first 18 results pages (180 GL items) were browsed, as the relevance became extremely weak after page 19. A pre-selection was applied to identify appropriate literature by reading titles, abstracts (for WL), and summaries (for GL).

3.2.3 Study Selection and Quality Assessment

Study selection criteria, i.e., inclusion and exclusion criteria, were defined to ensure that selected studies provided data to answer the research questions. Those criteria were for WL selection only. Because, in practice, GL selection criteria usually overlap and are integrated with a quality assessment guide (Garousi, Felderer, & Mäntylä, 2019).

Inclusion Criteria:

- The study identifies one or more primary aspects of DevSecOps, e.g., definition, challenges, practices, activities, solutions, tools, technologies, metrics, measurement, and global applications.
- The study is written in English.
- The study has been published from 2012 to date.
- The study has a clearly stated methodology or research design.
- The study has a credible source.

Exclusion Criteria:

- The study does not have a full-text version.
- The study is external to the subject area of computer science and software engineering.
- The study lacks a rigorous research method to validate its findings.

- Duplicate studies.
- Secondary studies.

Quality Assessment:

This step was applied to ensure compliance with additional selection criteria, to determine the validity of sources, assess the importance of studies, and minimise selection bias (Kitchenham, 2007).

Table 3. 1 lists the quality assessment (QA) criteria for WL and GL selection, which were adapted from the guidelines by Garousi, Felderer and Mäntylä (2019) and Kitchenham (2007). Garousi, Felderer and Mäntylä (2019) applied QA criteria to GL only. This MLR adapted and extended the QA criteria to cover WL. The first 14 questions, which pertained to six criteria, were answered with a "Yes" or "No" and were marked as 0 or 1 accordingly. According to Garousi, Felderer and Mäntylä (2019), literature types can be banded based on their credibility. This MLR rated "Literature Type" on a scale from 0 to 4. Thus, a full mark was set as 18 (14 + 4), and 11 (60% of 18) was set as the passing score.

Table 3. 1 - Quality assessment criteria

Criteria	Questions
Authority of the producer (Measure = 0 or 1)	• Is the author or the publishing organisation reputable? • Has the author published other work in the field? • Does the author have expertise in the area?
Methodology (Measure = 0 or 1)	• Does the work have a clearly stated aim? • Does the work have a stated methodology? • Does work have authoritative and contemporary references? • Are any limits clearly stated?
Objectivity (Measure = 0 or 1)	• Does the work provide objective statements or credible findings? • Is there a vested interest? E.g., a tool comparison by authors who are working for a particular tool vendor. • Do the data support the conclusion?
Publication Date (Measure = 0 or 1)	• Does the work have a clearly stated date?
Novelty (Measure = 0 or 1)	• Does the work have a novel idea or something unique? • Does the work strengthen or refute a current position?
Impact (Measure = 0 or 1)	• For WL, is the author's work cited often? / For GL, is the source viewed/shared/discussed often?
Literature Type	• WL: peer-reviewed academic papers or book chapters (Measure = 4).

(Measure = 0 to 4)	• GL: PhD and Master's theses (Measure = 3). • GL with high credibility, e.g., books, magazines, specialised databases, white papers, method creators, and consultants' websites and case studies (Measure = 2). • GL with medium credibility, e.g., technical reports, news, Q/A sites, blogs, presentations, and videos (Measure = 1). • GL with low credibility, e.g., ideas/opinions/thoughts/commentaries without evidence (Measure = 0).

Table 3. 2 provides several QA samples for WL and GL. Full QA results are provided in Appendix E.1, "MLR Included Papers and Quality Assessment Scores" (Page 363).

Table 3. 2 - Samples of quality assessments

White Literature								
Paper ID	**Criteria**							
	Overall (18)	**Authority (3)**	**Methodology (4)**	**Objectivity (3)**	**Novelty (2)**	**Impact (1)**	**Publication Date (1)**	**Literature Type (4)**
S1-ACM-01	15	2	3	3	2	0	1	4
S1-ACM-02	14	1	3	3	2	0	1	4
S1-IEEE-02	11	1	2	2	1	0	1	4
S1-IEEE-03	11	1	2	2	1	0	1	4
S1-SC-01	13	2	2	2	1	1	1	4
Grey Literature								
Paper ID	**Criteria**							
	Overall (18)	**Authority (3)**	**Relevance (7)**		**Novelty (2)**	**Impact (1)**	**Posted Date (1)**	**Literature Type (4)**
S1-GL-01	12	2	5		1	1	1	2
S1-GL-02	11	1	6		1	1	1	1
S1-GL-03	11	1	6		1	1	1	1
S1-GL-04	14	3	6		1	1	1	2
S1-GL-05	12	1	6		1	1	1	2

- "Authority of the producer" was assessed to ensure the credibility and expertise that a content creator has. For WL, publishers' reputation was judged through journal/conference rankings, and authors' authority was assessed through their expertise and other relevant publications. For GL, authors' names were searched on LinkedIn, and their work experiences were used to gauge authority. When a GL work lacked an author, the authority of the posting organisation was assessed instead. Several large global companies, such as Microsoft, Google, and Red Hat, do have high authority, though they post articles without the names of

authors.

- "Methodology" and "Objectivity" were assessed to check whether a WL work clearly stated its research methodology and objectivity. These two criteria were not applicable in GL assessment, thus were replaced with "Relevance", which was relatively subjective.
- To assess the "Impact" of WL, the names of the first and corresponding authors were searched on Google Scholar, and the number of citations served as the indicator. For GL's impact, the number of views, shares, and comments was measured.
- "Novelty" was also a relatively subjective criterion that evaluated the extent to which the work provided new findings, original ideas, innovative methodologies, or unique perspectives.
- "Literature Type" was rated on a scale from 0 to 4, covering WL and four tiers of GL in terms of credibility.

3.2.4 Snowballing

In addition to the database searches, the snowballing technique was applied to locate relevant studies. Initially, database searches were performed, and later complemented by snowballing searches. MLR and SLR as forms of secondary studies should exclude the other secondary studies to ensure credibility, but six key review papers (Akbar, Smolander, et al., 2022; Mohan & Othmane, 2016; Myrbakken & Colomo-Palacios, 2017; Prates et al., 2019; Rajapakse et al., 2022; Sanchez-Gordon & Colomo-Palacios, 2020) were identified to validate the dependability of the MLR findings, as listed in **Table 2. 1**, Chapter 2 (on Page 20). Intentional replication in MLR/SLR can validate the review results by comparing similar work on the same topic with different research questions, search strategies, paper selections, and analytical methods (Wohlin et al., 2022). Hence, snowballing was applied to those secondary studies.

The primary objective of snowballing in this MLR was to identify sources and validate the reliability of relevant studies, with a secondary objective of identifying additional papers. Database searches had already identified an extensive collection of papers, and the snowballing search was unlikely to yield any additional papers beyond those identified in the database searches. Thus, backward snowballing was mainly used, rather than forward snowballing.

Snowballing in the WL search primarily relied on Google Scholar, with a few additional databases also included, such as Springer, ScienceDirect, and the university library's databases. Similar to snowballing, "backlinks" in Google were navigated forward or backward to identify additional GL articles and validate their reliability.

3.2.5 Search Execution

Table 3. 3 summarises the MLR search execution. The number of collected studies was counted during the implementation of the search procedure.

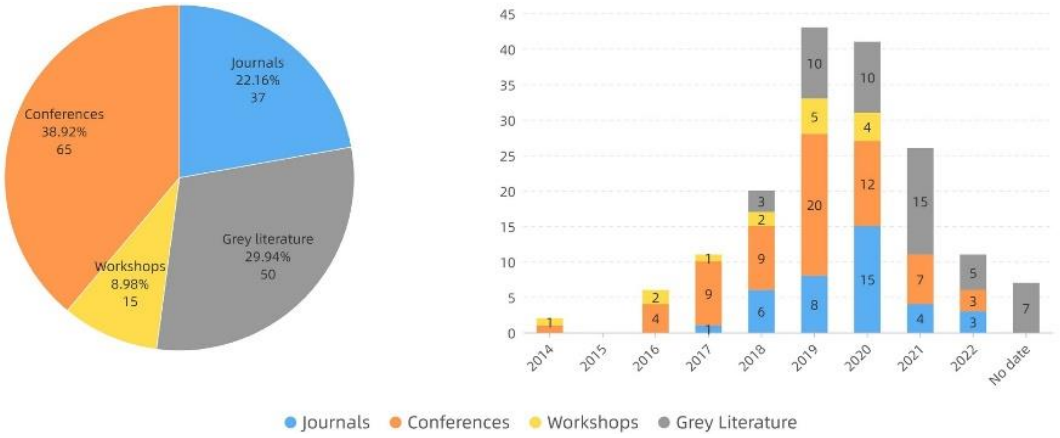
Table 3. 3 - MLR search execution

Search steps	Search string 1 results (WL/GL)	Search string 2 results (WL/GL)
Apply search string	692 (acm-416, ieee-100, scopus-176)/400 m studies	216 (acm-97, ieee-27, scopus-92)/150k studies
Study pre-selection	327 (acm-113, ieee-90, scopus-124)/180 studies	66 (acm-35, ieee-21, scopus-10)/100 studies
Study selection	238 (acm-101, ieee-88, scopus-49)/56 studies	8 (acm-7, ieee-0, scopus-1)/ 3 studies
Study QA	96 (acm-26, ieee-39, scopus-31)/43 studies	2 (acm-2, ieee-0, scopus-0)/0 study
Snowballing	102 (acm-26, ieee-45, scopus-31)/43 studies	2 (acm-2, ieee-0, scopus-0)/0 study

After the work of paper collection and selection had been completed, a continuous confirmatory process named "Confirmatory Search" was supplemented to identify the latest literature, thereby avoiding staleness and continuing validation. The same search strings were applied, along with additional databases, such as ScienceDirect and Springer. The selection criteria were broadened to include recent secondary studies for validation, rather than continuing the MLR. By 2022, 13 WL papers and 7 GL articles were newly added. However, the latest literature from the "Confirmatory Search" was not included in the thematic analysis to avoid affecting the original MLR results.

Figure 3. 2 breaks down the number of included articles based on source types and published years. 147 articles were collected from the MLR study, and 20 new articles were collected from the "Confirmatory Search" after the MLR study.

Figure 3. 2 - Number of included papers based on source types and published years



3.2.6 Data Extraction and Data Synthesis (Thematic Analysis)

Data Extraction:

Data extraction was a step to gather relevant information from the included studies, using a data extraction form attached to the review protocol as an appendix (see Appendix B.1 on Page 295).

Data Synthesis:

Data synthesis was performed to collate and summarise the results of data extraction (Kitchenham, 2007). In this MLR, data synthesis was conducted by using the Thematic Analysis (TA) method.

Thematic Analysis:

Thematic Analysis (TA) is a data analysis method for identifying, analysing, and reporting themes with data, combining both qualitative (text segments, codes, themes) and quantitative (frequency statistics) evidence (Braun & Clarke, 2006). TA is one of the most frequently used methods for data synthesis in the SE research; 2/3 of the systematic reviews in SE employed TA to synthesise the data from primary studies (Cruzes & Dyba, 2011b).

Compared to other synthesis methods, TA is said to be relatively easy to learn and perform, thereby being accessible to inexperienced researchers (Braun & Clarke, 2006). Additionally, flexibility is a key advantage of the TA method, enabling researchers to provide a wide range of analytic options (Braun & Clarke, 2006). The significant distinction between TA and another classic synthesis method – Grounded Theory (GT) is that GT uses an ongoing process of data coding throughout data collection. In contrast, TA is applied after data collection (Cruzes & Dyba,

2011b). Another distinction is that GT aims to create new theories, but TA is used to capture themes and summarise key features based on existing frameworks. Hence, the latter was a more appropriate choice for this study.

TA has three types: coding reliability, codebook, and reflexive. This study employed the reflexive TA method, which fully embraces qualitative research values and researchers' subjective skills, thereby fitting an experiential (e.g., critical realist, contextualist) and critical (e.g., relativist, constructionist) framing of language, data, and meaning (Braun & Clarke, 2021). The reflexive TA is a situated interpretative reflexive process that can be conducted inductively or deductively. Coding is open and organic, without a predefined coding framework. Themes emerge from data coding and iterative theme development (Braun & Clarke, 2020).

Compared to the reflexive TA, which is informed by interpretivism and advocates a relativist ontology and subjective epistemology, the coding reliability TA is informed by positivism and employs unbiased coding to achieve an objective reality from pure data without human interpretations; the codebook TA is informed by pragmatism and uses structured codebooks or coding frameworks with coding reliability approaches (Alharahsheh & Pius, 2020).

According to Braun and Clarke (2021), agreement between researchers and inter-rater reliability are not required as measures of quality for the reflexive and codebook TA, except for the coding reliability TA. They argue that it is "illogical, incoherent, and ultimately meaningless" to require coding reliability and bias suppression in reflexive TA, because meaning and knowledge are understood as situated and contextual, and researchers' subjectivity is conceptualised as a resource for knowledge production, rather than a threat to credibility. In other words, the reflexive TA is more applicable than the other two types for an individual researcher, so it was selected.

Model Creation:

The TA method had four levels of interpretation and abstraction: Text, Code, Themes, and Model (Cruzes & Dyba, 2011a). The PhD researcher read the included papers, identified text segments and labelled them with codes. Codes were translated into themes by removing overlaps, and themes were further classified into categories.

The TA process initially followed an inductive approach (coding/theming was directed by the

content of the data) (Braun & Clarke, 2020). After generating sets of codes/themes, it was supplemented with a deductive approach (coding/theming was directed by existing concepts) (Braun & Clarke, 2020).

Because "TA has limited interpretative power beyond mere description if it is not used within an existing theoretical framework" (Cruzes & Dyba, 2010), in addition to the inductively derived themes and categories, the thematic dimensions of the developing model were complemented with a DevSecOps lifecycle model by Gartner (MacDonald & Head, 2016), so that themes were deductively mapped to each applicable lifecycle phase to derive the conceptual model.

The TA outputs were reviewed by the supervisors of the PhD researcher through regular meetings. TA samples are provided in Appendix C (Page 350), and are also available at zenodo.org (https://doi.org/10.5281/zenodo.7959584).

3.2.7 Trustworthiness Assessment

To ensure the trustworthiness of the findings, TA tasks were reviewed using Braun's checklist (Braun & Clarke, 2021). In addition, the four components of trustworthiness (credibility, confirmability, dependability, and transferability) were assessed (Cruzes & Dyba, 2011a).

- Credibility is significantly concerned with the quality of selected primary studies (Cruzes & Dyba, 2011a). A quality assessment was therefore conducted on the selected white and grey literature. Another concern for achieving credibility is the suitability of text segments, particularly long segments, such as definitions of DevSecOps.
- Confirmability focuses on agreement among researchers (Cruzes & Dyba, 2011a). The PhD researcher's findings were reviewed and evaluated in consultation with his supervisors through regular meetings. Both supervisors of the PhD researcher are experienced scholars, and their recognition could also be a factor of confirmability.
- Dependability refers to the stability of findings (Cruzes & Dyba, 2011a). The review results were compared with the findings of other secondary studies on the same topic. According to Wohlin et al. (2022), intentional replication in MLR/SLR can validate findings by comparing similar work on the same topic with different research questions, search strategies, paper selection criteria, and analytical methods.

- Transferability is the extent to which the findings can be transferred to other settings (Cruzes & Dyba, 2011a). The subsequent empirical investigation using Delphi-AHP methods has validated the transferability of the conceptual framework, i.e., the artefact developed based on the MLR findings (presented in Chapters 4 – 8).

3.3 Results and Discussion

Results are reported and discussed in this section to answer the research questions and associated sub-questions. To validate and complement the MLR findings, they were further compared with the findings of six secondary studies on DevSecOps, i.e., (Akbar, Smolander, et al., 2022; Mohan & Othmane, 2016; Myrbakken & Colomo-Palacios, 2017; Prates et al., 2019; Rajapakse et al., 2022; Sanchez-Gordon & Colomo-Palacios, 2020).

3.3.1 RQ1 – Current State of DevSecOps

To answer RQ1 regarding the current state of DevSecOps in the existing literature, Search String 1 was applied to the ACM, IEEE, and Scopus databases, yielding an initial collection of 327 white publications. After pre-selecting and eliminating duplicates, 238 WL papers remained. After performing the study selection, quality assessment and snowballing, 102 WL papers were finally included. Identifiers of WL were created by mixing the search string number, the database, and the paper number. For instance, S1-ACM-01 represents that this paper is identified by Search String 1 and is from the ACM database.

Search String 1 was also applied on Google to search for GL work. The first 18 pages (resulting in 180 search results, as the results were relevant up to page 18) were browsed, allowing for the collection of 56 GL articles. After the quality assessment, 43 GL articles were finally included. Identifiers of GL were created by combining the search string number, "GL" for grey literature, and the paper number, e.g., S1-GL-01.

3.3.1.1 Five Aspects of DevSecOps Research

To answer Sub-question 1.1 *"What aspects of DevSecOps can be found in the existing literature?"*, five aspects of the DevSecOps topic have been identified: Definitions, Challenges, Practices, Measurements/Metrics, and Technologies/Tools.

Table 3. 4 defines the five aspects, and **Table 3. 5** lists the included WL and GL relating to each aspect. When the wording of studies was different, synonyms would be considered. For example, "Meanings", "Perceptions", and "Concepts" were categorised as fitting the "Definitions" aspect. "Problems", "Issues", and "Concerns" were categorised as fitting the ''Challenges'' aspect. "Activities", "Approaches", "Solutions", and "Strategies" were categorised as fitting the "Practices" aspect. Several minor aspects were omitted and regarded as part of the major aspect. For instance, "Characteristics" and "Benefits" of DevSecOps were always mentioned in definitions, thereby being considered codes or themes under the "Definitions" aspect.

Table 3. 4 - Five aspects of DevSecOps research

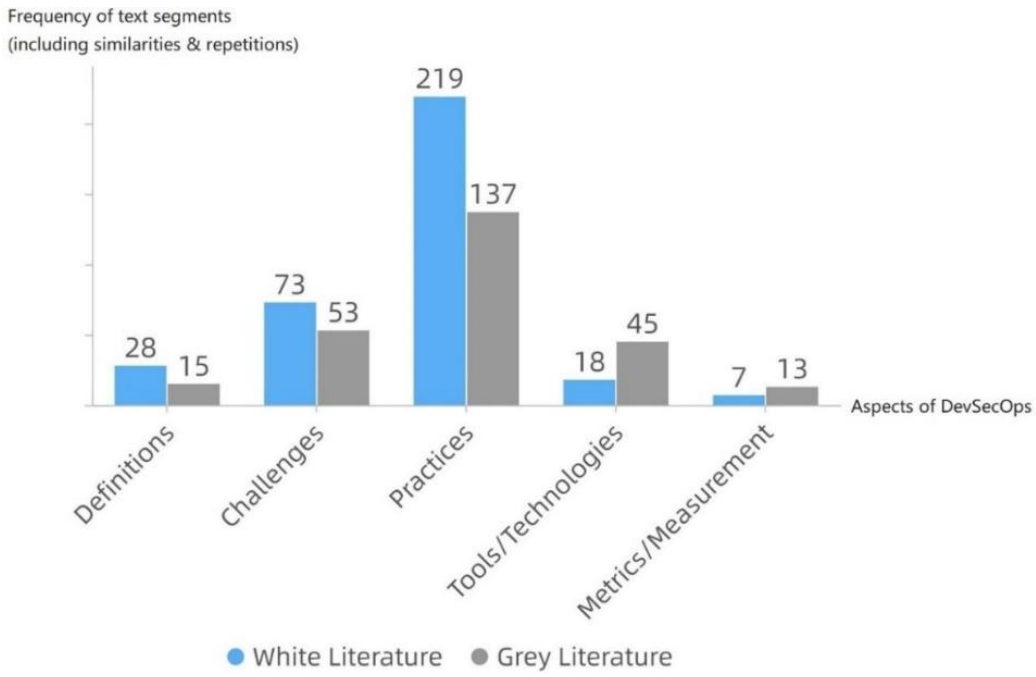
Definitions	The Definitions for the term 'DevSecOps' and equivalent terms. "Meanings", "Perceptions", and "Concepts" were categorised as fitting the "Definitions" aspect. "Characteristics" and "Benefits" were always mentioned in definitions, thereby being considered codes or themes under the "Definitions" aspect.
Challenges	The obstacles and uphill tasks encountered when adopting DevSecOps require ongoing efforts to overcome. "Problems", "Issues", and "Concerns" were categorised as fitting the ''Challenges'' aspect.
Practices	DevOps and security activities that are suited for DevSecOps. "Activities", "Approaches", "Solutions'' and "Strategies" were categorised as fitting the "Practices" aspect.
Tools/Technologies	Specific tools and technical approaches that are used for DevSecOps practices. They could be part of a subset of DevSecOps practices, particularly those related to technology.
Metrics/Measurement	The means to track progress, facilitate decision-making, and improve the performance of DevSecOps practices by measuring implementation, effectiveness, efficiency, and impact.

Table 3. 5 - WL and GL work relating to each aspect

Aspects	Related WL work (Paper ID)	Related GL work (Paper ID)
Definitions	S1-ACM-04, 07, 45, 50, 68, S1-IEEE-03, 05, 06, 08, 10, 12, 21, 22, 24, 26, 44, S1-SC-01, 02, 03, 04, 09, 10, 11, 14, 21, 22, 31, CS-ACM-01, 02, 03, 04, CS-IEEE-01, 02, 03, 04, 05, 06, CS-SC-01, 02, 03	S1-GL-01, 02, 04, 05, 10, 11, 12, 13, 15, 16, 19, 23, 26, 27, 33, CS-GL-01, 02, 03, 04, 05, 06, 07
Challenges	S1-ACM-01, 05, 06, 19, 52, 59, 64, 66, 95, S1-IEEE-01, 04, 06, 07, 08, 11, 12, 16, 25, 28, 33, 39, 42, S1-SC-08, 26, CS-ACM-01, 02, 04, CS-IEEE-01, 06, CS-SC-01, 02, 03	S1-GL-13, 15, 17, 18, 19, 20, 24, 28, 29, 30, 37, 38, 39, 40, CS-GL-07
Practices	S1-ACM-01, 02, 03, 08, 09, 15, 45, 49, 50, 52, 69, 71, 72, 81, 95, S1-IEEE-02, 04, 06, 07, 09, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 24, 26, 29, 30, 31, 33, 34, 36, 38, 39, 40, 41, 43, 52, 54, 55, 57, 61, 71, 84, 86, S1-SC-07, 08, 09, 11, 15, 17, 18, 20, 22, 26, 27, 32, 34, 36, 38, 40, 41, 42, CS-ACM-01, 03, 04, CS-IEEE-01, 02, 03, 04, 05, 06, CS-SC-02, 03	S1-GL-02, 04, 06, 08, 09, 10, 11, 13, 14, 15, 17, 18, 19, 22, 23, 24, 25, 28, 30, 31, 32, 35, 36, 41, CS-GL-01, 02, 03, 04, 06, 07
Tools/Technologies	S1-ACM-52, 76, 89, 95, 99, S1-IEEE-06, 07, 18, 31, 33, 39, 55, S1-SC-01, 09, 12, 18, 20, 26, 29, 34, 42, 45, 48, CS-ACM-01, CS-IEEE-01, 02, 03, 04, 05, 06, CS-SC-02	S1-GL-01, 03, 04, 10, 21, 23, 42, CS-GL-01, 02, 03, 05
Metrics/Measurement	S1-IEEE-06, 57, CS-ACM-03, CS-SC-02	S1-GL-01, 18, 43

Figure 3. 3 illustrates the distribution of the five identified DevSecOps aspects, detailing the total frequency of initially identified text segments for each aspect, including similarities and repetitions, which would be further coded and themed. Those initial text segments, such as phrases, clauses, and long sentences, could reflect the result more realistically than the codes and themes, which had been artificially processed.

Figure 3. 3 - Aspects of DevSecOps



As shown in **Figure 3. 3**, "Practices" is the most widely focused aspect of DevSecOps in the literature, while "Metrics/Measurement" has the least coverage. WL provides more results on definitions, challenges and practices, while GL focuses mainly on tools and metrics. This suggests that DevSecOps investigations from academia and industry are equally essential and complementary for both research and practice.

3.3.1.2 Themes and Classification

DevSecOps Definitions:

This MLR study has identified and reviewed as many DevSecOps definitions as possible in the existing white and grey literature, including 28 definitions from WL and 15 definitions from GL

(similarities and repetitions included). 74 codes were labelled from those definitions, and they were translated into 21 themes. **Table 3. 6**. lists the themes and codes, along with their categories, frequencies, and sources.

Table 3. 6 - Thematic synthesis of DevSecOps definitions

Categories	21 Themes (Freq)	74 Codes [Papers contributed to the code] (Freq)
Organisation, People and Culture (OPC)	Expansion to DevOps (4)	expansion to DevOps [S1-IEEE-08, S1-SC-21] (2)
		extension to DevOps [S1-SC-01] (1)
		extension of the DevOps [S1-GL_33] (1)
	Dev, Sec & Ops (10)	development, operations and security teams [S1-IEEE-05, 08, 12, S1-SC-10, 21, S1-ACM-68, S1-GL-15, 19, 27] (9)
		dev/sec/ops [S1-IEEE_26] (1)
	Culture (8)	culture [S1-ACM-45, S1-GL-10, 13, 26] (4)
		cultural approach [S1-IEEE-26] (1)
		cultural shift [S1-ACM-50, S1-GL-11] (2)
		shift the mindset [S1-IEEE-10] (1)
	Collaboration (9)	collaboration/collaborate [S1-IEEE-08, 12, 26, S1-SC-10, 21, S1-ACM-45, 68, S1-GL-26] (8)
		team work [S1-GL-02] (1)
	Breaking silos of security (4)	breaking silos of security [S1-IEEE-08, 24, 26] (3)
		break down the barrier [S1-IEEE-22] (1)
	Sharing knowledge (3)	sharing that knowledge [S1-IEEE-08] (1)
		giving that knowledge to the different teams [S1-IEEE-24, 26] (2)
	Shared responsibility (6)	shared responsibility [S1-GL-10, 33] (2)
		everyone's responsibility [S1-GL-10] (1)
		security is a part of everyone's job [S1-GL-12] (1)
		make everyone accountable for security [S1-GL-27] (1)
		at the top of every developer's mind [S1-GL-12] (1)
	Philosophy (3)	philosophy [S1-GL-02, 19, 26] (3)
	Communication (1)	communication [S1-GL-19] (1)
	Combination of DevOps and SecOps (1)	combination of DevOps and SecOps [S1-GL-13]
Process Capabilities (PC)	Integration of security into DevOps (21)	incorporating security practices in the DevOps processes [S1-IEEE-08, S1-SC-21] (2)
		incorporation of security practices in a DevOps environment [S1-SC-10, 11] (2)
		IT processes with security approach [S1-ACM-04, S1-IEEE-21] (2)
		integration of security with development and operation [S1-SC-09] (1)
		integrating security principles [S1-IEEE-12] (1)
		integration of security processes and practices [S1-IEEE-10, S1-GL-10] (2)
		introduction of more security-oriented processes [S1-SC-22] (1)
		integrates continuous security into the original DevOps process [S1-IEEE-03] (1)
		injection of security principles and controls into the DevOps [S1-ACM-50] (1)
		integrating secure development best practices and methodologies into development and deployment processes [S1-IEEE-44] (1)

		integrating the software development and operation processes considering security and compliance requirements [S1-SC-11] (1)
		integrating security methods into a DevOps process [S1-GL-02] (1)
		integrating security practices within the DevOps process [S1-GL-26] (1)
		adding security components to each step of the DevOps [S1-GL-23] (1)
		bake security into the rapid-release cycles [S1-GL-11] (1)
		integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline[S1-GL-16] (1)
		built-in security [S1-GL-04] (1)
	Agile (4)	Agile [S1-ACM-45, S1-IEEE-03, S1-GL-05] (3)
		smart and lightweight approach [S1-SC-31] (1)
	Security is the main concern throughout the SDLC (7)	security is the main emphasis [S1-SC-14] (1)
		security is given high priority throughout the SDLC [S1-ACM-07] (1)
		a key concern throughout all phases of the development lifecycle and even post deployment [S1-SC-31] (1)
		security practices are implemented at each stage of the cycle [S1-ACM-07] (1)
		security is implemented at the right level and at right time [S1-IEEE-24] (1)
		emphasises the importance of sound information security practices [S1-GL-01] (1)
		adoption of security through the entire SDLC [S1-GL-19] (1)
	Shifting security to the start (8)	puts security at the forefront of requirements [S1-IEEE-24] (1)
		shifting security to the early stages [S1-IEEE-06] (1)
		security from the start/beginning [S1-GL-04, 15, 33] (3)
		integrate security objectives as early as possible [S1-GL-10] (1)
		placing security practices early during the SDLC [S1-GL-05] (1)
		avoids any risk of security being an afterthought [S1-GL-01] (1)
	Time reduction & Efficiency improvement (4)	time reduction [S1-ACM-04, S1-IEEE-21] (2)
		increase deployment rates [S1-IEEE-22] (1)
		shorten the SDLC [S1-GL-23] (1)
	Security assurance (3)	maintaining a secure operational atmosphere [S1-IEEE-22] (1)
		identifying security vulnerabilities [S1-SC-31] (1)
		responsible for application security [S1-IEEE-05] (1)
Technology	Tooling (2)	reliance on operational tools [S1-ACM-45] (1)
		tooling [S1-GL-10] (1)
	Automation (2)	automation/automating [S1-GL-04, 19] (2)
	Security as code (1)	security as code [S1-GL-26] (1)
Business	High quality (4)	without lost quality [S1-ACM-04, S1-IEEE-21] (2)
		quality affirmation [S1-SC-14] (1)
		high software quality [S1-GL-23] (1)

Some definitions of DevSecOps were repeatedly quoted or paraphrased by the included papers. Snowballing was applied with WL to trace the sources of those definitions, while GL had no references to enable snowballing. However, all the traced sources were secondary studies and thus

were excluded. **Table 3. 7** identifies the sources of those cited definitions and the corresponding included papers that quote or paraphrase the definition. The definition by Mohan and Othmane (2016) is the most cited (by 9 papers): ‘‘DevSecOps refers to incorporating security practices in the DevOps processes by promoting collaboration between the development, operations and security teams.’’

Table 3. 7 - Sources of common definitions

Sources of common definitions	Papers that quote or paraphrase the definition	Frequency
(Mohan & Othmane, 2016)	S1-ACM-45, S1-ACM-68, S1-IEEE-08, S1-IEEE-26, S1-SC-09, S1-SC-10, S1-SC-11, S1-SC-21, S1-SC-22	9
(Rahman & Williams, 2016)	S1-IEEE-08, S1-IEEE-12, S1-IEEE-44, S1-SC-22	4
(Carter, 2017)	S1-IEEE-24, S1-IEEE-26	2
(Carturan & Goya, 2019)	S1-IEEE-21, S1-ACM-04	2
(Myrbakken & Colomo-Palacios, 2017)	S1-IEEE-10	1
(Mohan, Othmane, & Kres, 2018)	S1-SC-11	1

Classification:

Themes were further classified into four categories: Organisation, People and Culture (OPC), Process Capabilities (PC), Technology, and Business, as defined in **Table 3. 8**. These four categories were elicited from the data synthesis of DevSecOps definitions as shown above in **Table 3. 6**, revealing four different perspectives and ways to define and understand the DevSecOps approach.

Table 3. 8 - Four categories of themes

Organisation, People and Culture (OPC)	The category encompasses themes related to organisational structure, people management, and cultural strategies, including breaking silos, collaboration, communication, sharing, training, and recruiting.
Process Capabilities (PC)	The category encompasses themes related to the capabilities of the DevSecOps process, including the integration of security, security-left, continuous activities, risk management, and a faster lifecycle.
Technology	The category encompasses themes related to technological approaches, software, and hardware tools, including automation, cloud, containerisation, testing techniques, and associated tools.
Business	The category encompasses themes related to business benefits, customers, quality of product and service, e.g., increasing value, higher quality, fewer impacts to users, etc. The reason for adding this category was that the MLR, especially the GL results, showed a business perspective on DevSecOps.

The literature on DevSecOps has two common taxonomies. Smeds, Nybom and Porres (2015) define DevSecOps with three categories: "Capabilities, Cultural Enablers, and Technological Enablers". Humble and Molesky (2011) present a taxonomy for DevSecOps, grouped into four categories: "Culture, Automation, Measurement, and Sharing (CAMS)", which is more widely cited and used. However, both taxonomies are derived from DevOps definitions and principles, thereby not fully capturing the DevSecOps approach (e.g., DevSecOps technology includes but is not limited to automation).

Hence, based on the MLR findings in combination with those two taxonomies, this new taxonomy was derived from the four categories of emerging themes of DevSecOps definitions. It would form the basis for later grouping, i.e., grouping the other four aspects: Challenges, Practices, Tools, and Metrics.

DevSecOps Challenges, Practices, Tools, and Metrics:

To answer Sub-question 1.2 *"What themes do these aspects contain?"*, **Table 3. 9** summarises the results of the thematic analysis (TA) process, which was performed to collect, analyse, and report the related themes of each aspect (i.e., Definition, Challenges, Practices, Tools/Technologies, and Metrics/Measurement.) The TA processes for Challenges, Practices, Tools, and Metrics were similar to those for Definitions, as previously shown in **Table 3. 6**. Those lengthy tables are provided in Appendix C, "Samples of Thematic Analysis in MLR Study" (Page 350), or at zenodo.org (https://doi.org/10.5281/zenodo.7959584), rather than in the main text.

Table 3. 9 - Thematic analysis and synthesis results

Aspects	Extracted data (text segments) WL/GL	Coded data (codes)	Translated codes into themes	Classified themes into categories
Definitions	28/15 definitions	74 codes	21 themes	4 categories: OPC, PC, Technology, Business
Challenges	73/53 challenges	85 codes	23 themes	4 categories: OPC, PC, Technology, Business
Practices	219/137 practices	142 codes	56 themes	4 categories: OPC, PC, Technology, Business
Tools/Technologies	18/45 tools	56 codes	16 themes	Single category: Technology
Metrics/Measurement	7/13 metrics	20 codes	16 themes	3 categories: OPC, PC, Technology

The identified challenges, practices, tools, and metrics are listed in **Figure 3. 4**, **Figure 3. 5**, **Figure 3. 6**, and **Figure 3. 7**. The full text version is provided in Appendix E.2, "Glossary of Findings" (Page 379), along with their categories, explanations, related codes, and source papers.

Figure 3. 4 - DevSecOps challenges

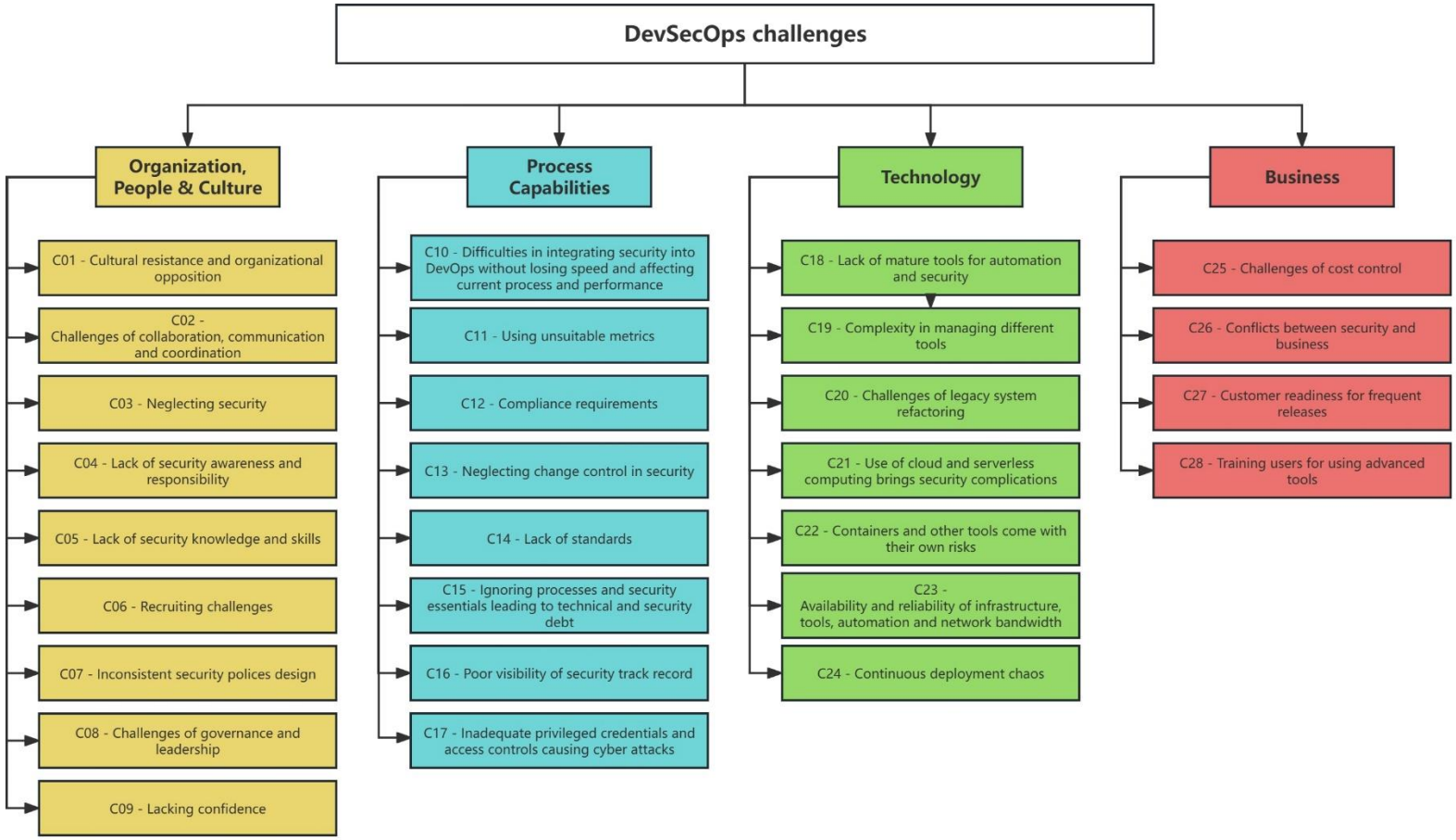


Figure 3. 5 - DevSecOps practices

Figure 3. 6 - DevSecOps tools

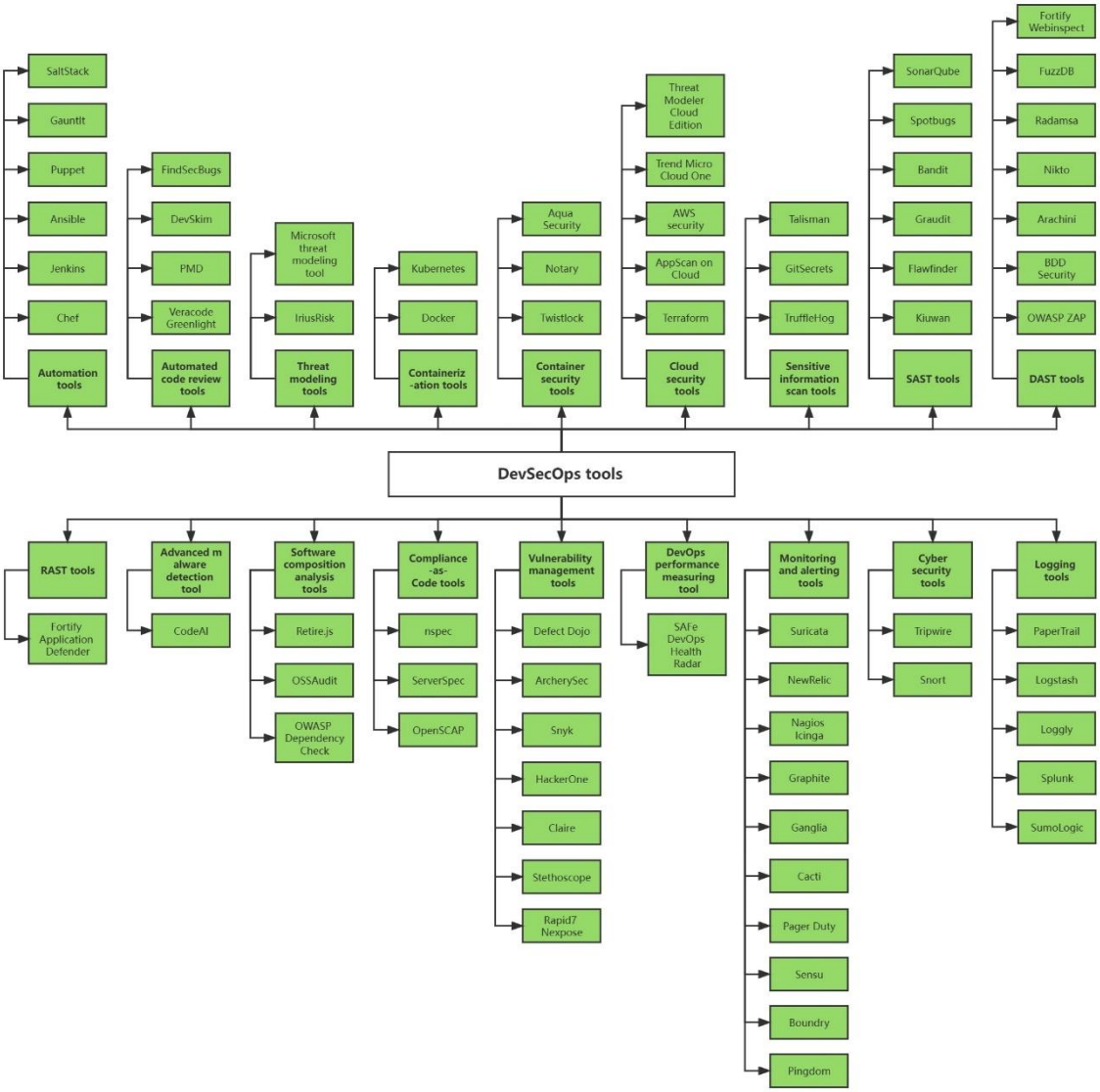
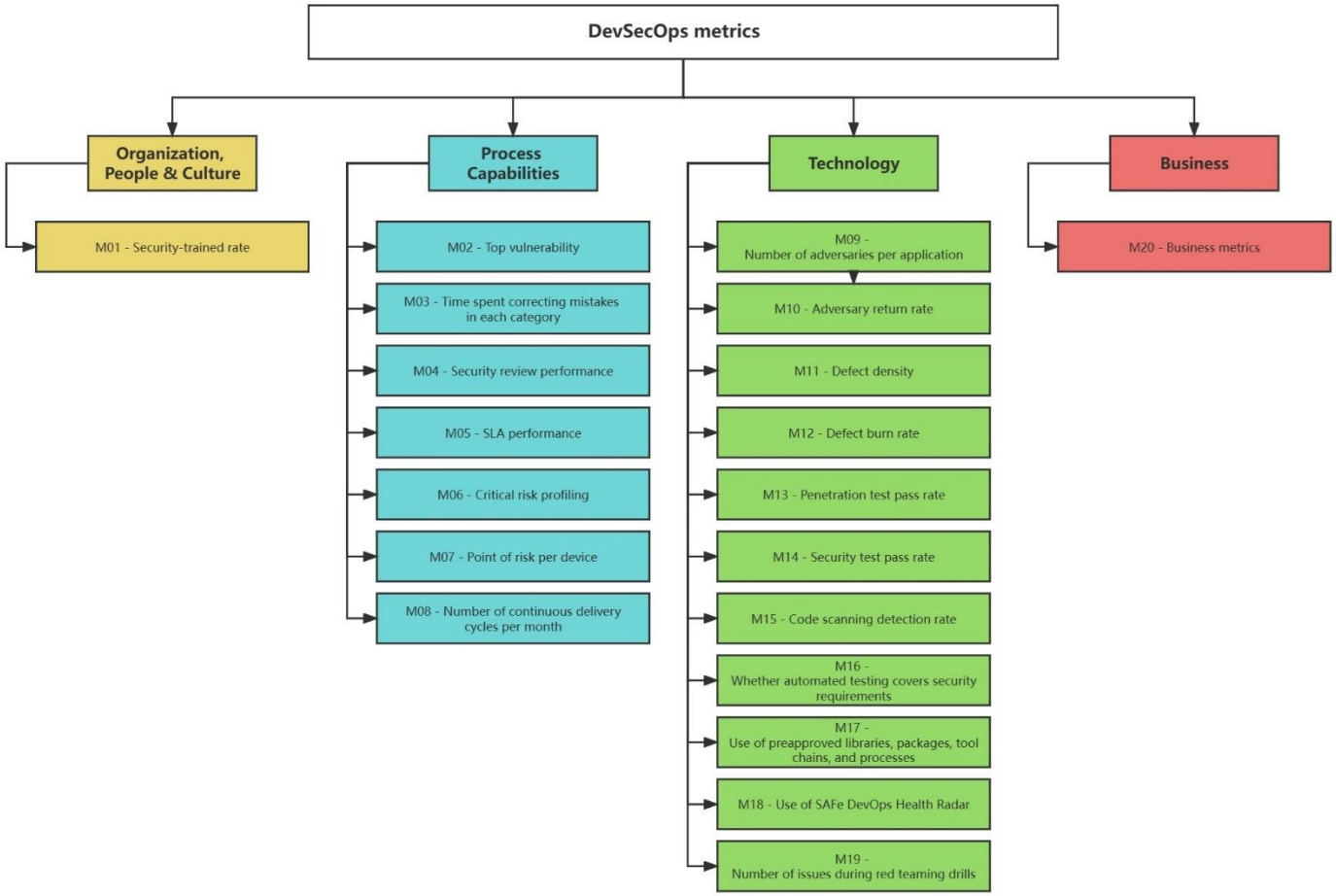


Figure 3. 7 - DevSecOps metrics



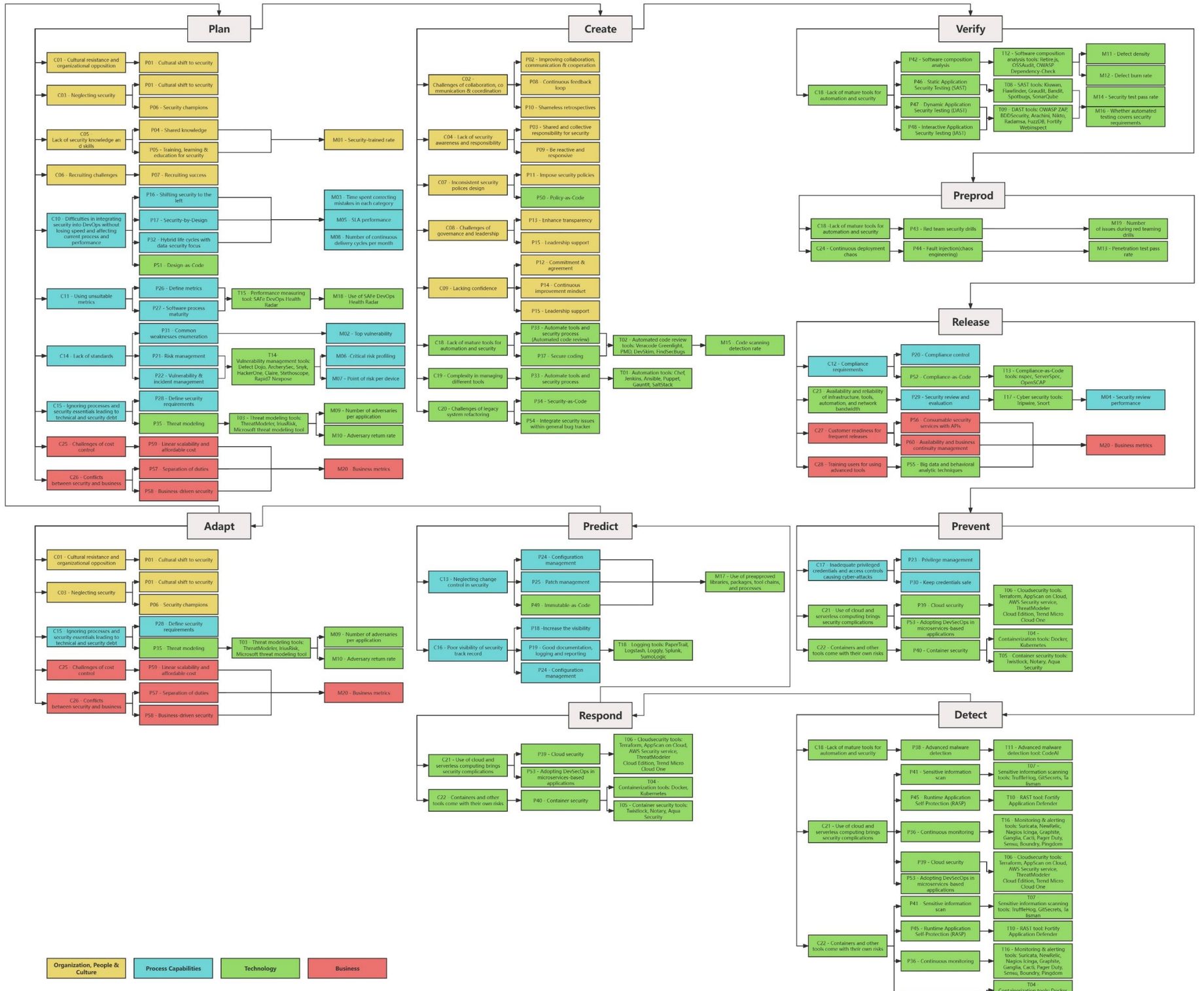
3.3.1.3 Links Between Aspects and Themes – DevSecOps CPTM Model (Version 1.0)

To answer Sub-question 1.3 *"How do the identified aspects and themes link to each other?"*, a conceptual framework named "DevSecOps Challenge-Practice-Tool-Metric (CPTM) Model (Version 1.0)" has been deduced in **Figure 3. 8**.

According to Jabareen (2009), the conceptual framework is defined as "a network or a plane of interlinked concepts that together provide a comprehensive understanding of a phenomenon or phenomena. The concepts that constitute a conceptual framework support one another, articulate their respective phenomena, and establish a framework-specific philosophy".

As shown in **Figure 3. 8**, the DevSecOps CPTM Model (Version 1.0) covers 28 challenges, 60 practices, 20 metrics, and 71 tools within 18 types, and shows the links between these four aspects/elements of DevSecOps associated with the four categories, which are: "Organization, People and Culture" is shaded in yellow; "Process Capabilities" in blue; "Technology" in green; and "Business" in red. These categories are derived from DevSecOps definitions and serve as the basis for grouping the challenges, practices, tools, and metrics associated with DevSecOps. However, definitions themselves are not involved in the CPTM Model, as they are not appropriate.

Figure 3. 8 - DevSecOps CPTM (Challenge-Practice-Tool-Metric) Model (Version 1.0)



By referring to Gartner's DevSecOps model (MacDonald & Head, 2016) (in **Figure 3. 9**), the DevSecOps lifecycle is decomposed into ten phases: Plan, Create, Verify, Preproduce, Release, Prevent, Detect, Respond, Predict, and Adapt, which are defined in **Table 3. 10**.

Figure 3. 9 - DevSecOps lifecycle by Gartner

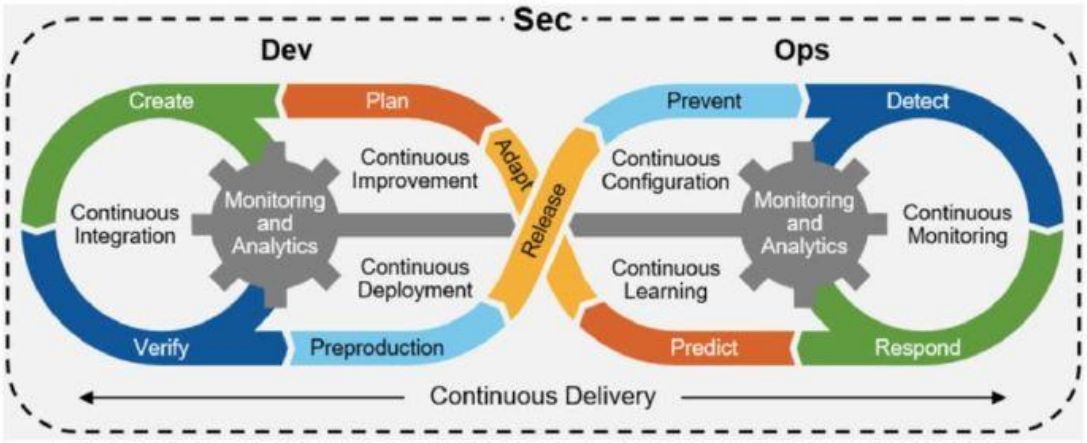


Table 3. 10 - Ten phases of the DevSecOps model by Gartner

Plan	The phase involves setting project objectives, identifying security requirements, planning security measures, defining metrics and policies, preparing organisations/teams, selecting technologies/tools, and developing budgets.
Create	The phase involves executing the plan, preparing security practices, and setting up security tools.
Verify	The phase is to conduct security practices by using appropriate (automated) tools and technologies, such as security tests (SAST, DAST, IAST) and software composition analysis (SCA).
Preproduction	The next phase involves incorporating additional security tests, including chaos engineering and red team drilling.
Release	The phase involves signing the software, preparing it for release, and integrating it into the production environment. This includes reviewing configuration, infrastructure, network bandwidth, compliance, and other relevant factors.
Prevent	The phase is to protect the runtime environment architecture.
Detect	The phase involves continuously monitoring and scanning the runtime environment architecture.
Respond	The phase is to address the vulnerabilities detected in the previous phase.
Predict	The phase involves analysing vulnerabilities to identify their causes.
Adapt	The phase aims to enhance security processes and re-plan the DevSecOps lifecycle, drawing on lessons learned from previous phases.

The ten phases of the DevSecOps lifecycle by Gartner (MacDonald & Head, 2016) are integrated into the DevSecOps CPTM Model, and all identified themes of the four elements are allocated to these lifecycle phases. These phases form a loop that starts with planning and ends with adapting/re-planning, where each iteration is completed, and the next is improved. The reason for

integrating the findings from this research into those phases is that Gartner's DevSecOps model has been widely accepted by the SE industry and academia, and appears to be one of the most popular reference models for DevSecOps adoption (Prates & Pereira, 2025; Zhao, Clear, & Lal, 2024b)

Within each lifecycle phase, there are four columns indicating the four elements of the model: Challenges, Practices, Tools, and Metrics. The connecting lines demonstrate the relationships between those four elements. It outlines the practices that can be applied to overcome the corresponding challenges, the tools that can support DevSecOps practices, and the metrics that can be used to measure the performance of these practices and tools.

In the DevSecOps CPTM Model (Version 1.0), one challenge may correspond to multiple practices, one practice may correspond to multiple challenges, and not every practice has its corresponding tools and metrics. For example, some practices, especially many in the "Organisation, People and Culture" category (yellow in the model), do not require support from tools and technologies. Besides, additional business costs for monitoring and collecting data for measurement must be considered, so that it is not necessary to define a metric for each practice. This is why the model includes 60 DevSecOps practices, but only 20 metrics and 18 tool types. It is also worth noting that a few elements appear to be cross-cutting themes across categories, and their categories potentially differ from those in the thematic analysis. For example, tools are categorised under the "Technology" category (green colour), but some may also appear in other categories within the model to match their corresponding practices.

3.3.1.4 Summary of Answer to RQ1

The DevSecOps CPTM Model (Version 1.0) reveals the current state of DevSecOps across academia and industry and captures existing experience in this area. It also provides a breakdown and a broad overview of DevSecOps, from which researchers and practitioners can select an area of focus to enhance their knowledge or practices. In comparing academia and industry (WL and GL), scholars have contributed to research studies on the phenomenon by defining key concepts and identifying associated challenges and practices. By contrast, industry practitioners have made more significant contributions to the business perspective and pragmatic implications of the DevSecOps approach, focusing on practical tools and metrics to deliver solutions.

3.3.2 RQ2 – Adopting DevSecOps in GSE

After applying Search String 1 across all search sources, the results did not include any work on adopting DevSecOps in GSE. To address RQ2, the additional Search String 2 was applied, resulting in 126 WL papers. After eliminating duplicates, 66 papers remained. However, most of them focus on global DevOps without addressing security. After study selection and QA, only two papers were included. The search results held even after Search String 2 was adjusted numerous times, e.g., by trying additional keywords such as "multi-site", "multi-national", "transnational", etc. Search String 2 and its variants were also applied on Google to search GL. After browsing the first 10 pages (100 results), no GL work involving the three terms (DevOps, security, and GSE) was found.

The results reveal a notable absence of adopting DevSecOps in GSE settings to help answer RQ2: *"How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?"*. Most of the existing literature simultaneously covers two of the three terms (i.e., DevOps, security, and GSE). For instance, some papers focus on DevOps adoption in GSE, excluding security; others cover DevOps and security (i.e., DevSecOps), excluding GSE.

Four assumptions may be presented for this result. First, there may be no significant correlation between GSE and DevSecOps. Specifically, DevSecOps lacks distinguishing characteristics whether adopted locally or globally. Assumption two may be that security is typically a centralised and control-oriented function in organisations, so global aspects are not prominent. The third assumption is that a research gap exists in this area. The fourth assumption, which may also be a limitation, is that some terminologies were overlooked in determining our search string, even though the search strings have been revised to address this oversight. The first two assumptions or possibilities have been verified through the empirical investigation presented in Chapters 4–8.

3.4 Chapter Summary

This chapter summarises the published Multivocal Literature Review (MLR) study, which reviews the existing white and grey literature on DevSecOps and its adoption in GSE contexts. The MLR study identifies five aspects of DevSecOps (Definitions, Challenges, Practices,

Tools/Technologies, Metrics/Measurement); collects related themes of each aspect using Thematic Analysis (TA); and builds a DevSecOps CPTM Model (Version 1.0) by integrating the included themes of the latter four aspects. It further identifies the missing global dimension of DevSecOps in the literature and discusses the possible reasons.

Through the MLR study, the current state of DevSecOps was reviewed, and the DevSecOps CPTM Model (Version 1.0) was established as the theoretical basis for the subsequent research. Nonetheless, Version 1.0 still had gaps that needed to be filled, e.g., the absence of global aspects and the gap between identified tools and practices.

Hence, an empirical investigation was necessary to assess and validate the MLR findings and further refine and upgrade the DevSecOps CPTM Model from Version 1.0 to Version 2.0 by incorporating additional GSE aspects. In Chapter 4, the research design of this empirical investigation, i.e., the combined Delphi-AHP study, is presented.

4 Chapter 4: Research Design – Combined Delphi-AHP Study

The aim of Chapter 4 is to present the research design of an empirical study that combined a Delphi survey with the Analytic Hierarchy Process (AHP), aimed to evaluate and validate the findings of the previous MLR study and further refine and upgrade the DevSecOps CPTM Model from Version 1.0 to Version 2.0.

Section 4.1 introduces the research paradigm, and Section 4.2 discusses the methodology. Section 4.3 describes two research methods (i.e., Delphi and AHP) and their combination. Section 4.4 presents a reflection on related research, including previous Delphi, survey-based, and AHP-based studies on DevSecOps or DevOps, as well as studies that have applied the Delphi-AHP combination in other research domains. In Section 4.5, the research design and implementation are presented in detail, divided into three phases: Preparation, Conduct, and Analysis. Section 4.6 finally concludes this chapter.

4.1 Research Paradigm

Paradigm is defined by Guba and Lincoln (1994) as the particular combination of basic belief systems or world views (called “ontology”) with associated epistemologies, so that it directs the research ontologically (i.e., the nature of reality) and epistemologically (i.e., nature and scope of knowledge) in selecting the research methodology (Guba & Lincoln, 1994).

Ontology and epistemology give rise to two key research paradigms: positivism and interpretivism. Positivism seeks to establish an objective reality by collecting pure data or observing facts, unaffected by human interpretation. Interpretivism concerns the in-depth variables and factors in a given context; it views humans as distinct from physical phenomena, as they add further depth to meaning, and assumes that human beings cannot be explored in the same way as physical phenomena (Alharahsheh & Pius, 2020).

The research design of this doctoral research involved both positivist and interpretive paradigms. Although combined approaches are methodologically challenging (Cruzes & Dyba, 2010), this

research adopted a pragmatic strategy, as recommended by Cruzes and Dyba (2010), given its complex design and methodologies. Separately, each stage of this research drew on both positivism and interpretivism.

The earlier steps of the MLR study were more in line with the positivist perspective underlying the evidence-based SE movement (Kitchenham, Dyba, & Jorgensen, 2004). Although bias in study selection, quality assessment, and data extraction was inevitable, it could be mitigated by formulating and implementing a review protocol. On the other hand, "a literature review considers synthesis and interpretation as a mandatory property" (Rowe, 2014); hence, the Thematic Analysis (TA) process for data synthesis was addressed through an interpretivist stance, which advocated a relativist ontology and subjective epistemology, using a qualitative method (Alharahsheh & Pius, 2020).

This empirical study, which employed a combination of Delphi and AHP methods, enabled a pragmatic investigation that incorporated positivist and interpretive paradigms. Delphi is recognised as a data collection method that requires expert participants and aligns with the interpretivist research paradigm. The researcher identified, selected, invited, enlisted, and classified experts as participants for the study. Hence, his subjectivity, experience, available resources, selection criteria, and even invitation techniques and connecting skills could influence the study. Meanwhile, the Analytic Hierarchy Process (AHP) is a decision-making approach that adopts a positivist stance. This study employed AHP to design the survey in a quantitative format, and AHP results were derived solely from a computational process, without the researcher's interpretation.

4.2 Research Methodology

Qualitative research methodology was employed in this research work to achieve the research aim, which was defined as "to provide an in-depth understanding of DevSecOps and its adoption in GSE by developing an empirically grounded conceptual framework". Qualitative research methodology can be conducted in a variety of ways, including conceptual papers, surveys, single-case studies, multiple-case studies, action research, interviews, grounded theory, and mixed-methods designs (Biedenbach & Müller, 2011).

For this empirical investigation, a special survey, i.e., a Delphi study, was conducted. In the previous MLR study, the TA process for data synthesis was used to draft the conceptual framework by aggregating identified codes and themes from qualitative data. The Delphi study was conducted to validate and refine this conceptual framework by soliciting opinions and feedback from carefully selected, identified experts, whom the researcher grouped. A key distinction between quantitative and qualitative research is that the latter does not require the researcher to survey a large sample to make inferences about the targeted population (Greene, 2020). Overall, this research employed Delphi as a qualitative study and also incorporated a quantitative survey that used the AHP method to yield results.

4.3 Research Methods – Delphi and AHP

This empirical study employed a combination of the Delphi survey with the AHP method. This section discusses the features and scope of both methods, along with the reasons for selecting them.

4.3.1 Delphi Survey

The Delphi method is a data collection method, also known as the expert survey method, that solicits opinions and feedback from experts in specific domains by conducting multiple rounds of interviews or surveys to generate insights and reach consensus on controversial subjects with limited information (Beiderbeck et al., 2021).

Key features and advantages of the Delphi method include:

- Participants are not selected at random but are selected for their expertise (Loo, 2002).
- The number of participants can be much smaller than what is traditionally considered sufficient to guarantee the reliability of an ordinary survey (Loo, 2002).
- Participants are anonymous to each other but not to the researcher, helping to support objectivity and independence by avoiding the influence of other participants (Hasson, Keeney, & McKenna, 2025).
- Surveys are conducted online, without geographical restrictions (Ho et al., 2018).

- The rationale for Delphi is its iterative nature and the use of participant feedback. Aggregated group opinions are not only used for data analysis but are also fed back to participants for use in subsequent rounds (Bastiaansen & Wilderom, 2021).
- Delphi can be used to collect quantitative and qualitative data (Lilja, Laakso, & Palomäki, 2011). It can therefore take various formats, such as rating scales, yes/no questions, multiple-choice questions, and open-ended questions (Bastiaansen & Wilderom, 2021).

Although this empirical investigation employed a hybrid-method approach, its core remained a Delphi survey. Beiderbeck et al. (2021) define a standard Delphi process for all disciplines, which consists of three phases: Preparation, Conduct, and Analysis. Each phase contains a set of steps. The Delphi process was adjusted by adding or removing specific steps to fit this research. The research design and implementation are presented in Section 4.5.

4.3.2 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) is a multi-criteria decision-making approach developed by Professor Thomas Saaty in 1980 (Saaty, 1982). This approach provides a natural psychophysical way for decision-makers to structure the decision hierarchically (to reduce the complexity of the goal), prioritise the criteria, show the relationships between criteria and alternatives, and make the final decision (Brunelli, 2015; Saaty, 1982, 2013).

AHP uses a hierarchical structure to synthesise information quantitatively and logically, so that a complex problem (e.g., in this research, the DevSecOps CPTM Model with numerous elements in different categories) can be broken down into hierarchies, and the elements in each level can be evaluated by providing better focus during the priority allocation (Mu & Pereyra-Rojas, 2017; Saaty, 2013). AHP is adaptable to both groups and individuals and encourages compromise and consensus-building (Packeer Mohamed et al., 2022), making it well-suited to the Delphi method. Hence, the AHP approach was selected to formulate closed-ended questions for the Delphi survey.

Another classic decision-making approach for multicriteria data analysis – the Preference Ranking Organisation Method for Enrichment Evaluations (PROMETHEE) is also applied by researchers in similar studies, such as (Rafi et al., 2020). AHP and PROMETHEE have their

strengths and weaknesses. Compared to PROMETHEE, AHP offers a distinct advantage. It decomposes complex systems into subsystems and constructs hierarchies of criteria, thereby allowing decision problems to be broken down into their smallest elements and enabling the determination of different weights between criteria. PROMETHEE lacks this characteristic (Macharis et al., 2004). Another distinction is that AHP involves much less complex mathematics than PROMETHEE (Ishak, Asfriyati, & Akmaliah, 2019). Its ease of use makes it more suitable and popular among researchers without a strong mathematical background. However, AHP has a drawback: it can be very time-consuming due to the large number of pairwise comparisons required.

In general, there are seven steps in an AHP process (Mu & Pereyra-Rojas, 2017) as shown follows. However, this Delphi-AHP study only carried out the Steps One to Three (bold), because the evaluation of the DevSecOps CPTM Model (Version 1.0) did not involve making any decision from among alternatives.

(1) Build a hierarchical structure for the decision.

(2) Derive priorities/weights for the criteria and sub-criteria.

(3) Perform consistency analysis.

(4) Derive priorities/weights for the alternatives.

(5) Synthesise the model.

(6) Perform sensitivity analysis.

(7) Make a final decision.

4.3.2.1 Step (1): Build a hierarchical structure

First, a hierarchical structure was built to decompose the complex problem into subsystems. An AHP structure typically comprises three levels: the top level represents the overall goal for a problem, the middle level encompasses various criteria (with potential additional sub-criteria), and the bottom level comprises multiple alternatives (Mu & Pereyra-Rojas, 2017).

For this research, a hierarchical structure was developed to decompose the DevSecOps CPTM Model (Version 1.0) into four associated categories (i.e., OPC, PC, Technology, and Business).

The hierarchical structure contained only the goal level, the criteria level for the four categories, and four sub-criteria levels for Challenges, Practices, Tools, and Metrics, as it did not involve decision-making among alternatives. Hence, only Steps (1), (2), and (3) were performed.

Past studies show that discarding the alternatives level and using the criteria/sub-criteria level alone as a method of evaluation or prioritisation is common across various fields and disciplines. For example, Khan and Shameem (2020) employed AHP to calculate the prioritisation weights for 16 DevOps challenging factors; Zohaib, Alsanad and Abdullah Alhogail (2024) used fuzzy AHP to prioritise 48 DevOps implementation guidelines; Packeer Mohamed et al. (2022) used AHP to determine the priorities of the evaluation criteria for an Extended Software Process Certification (ESPAC) model; and Ali et al. (2017) ranked the importance of criteria for choosing the suitable wind farm sites by using AHP, but they did not really make any decision.

Figure 4. 1, **Figure 4. 2**, **Figure 4. 3**, and **Figure 4. 4** show the original AHP structures for the Challenges, Practices, Tools, and Metrics of DevSecOps, respectively. These were the findings of the previous MLR study that needed to be validated and refined through this empirical investigation. The design of the AHP structures were revised as the Delphi iterations progressed.

Hence, the DevSecOps CPTM Model (Version 1.0) was decomposed into the AHP format, which has the following hierarchical structures:

- The top level is the “Goal level”.
- The second level is the criteria level; in this case, it is named “Categories level”, including the four categories in the DevSecOps CPTM Model: Organisation, People and Culture (OPC), Process Capabilities (PC), Technology, and Business.
- The third level is the sub-criteria level, which is respectively named “Challenges”, “Practices”, “Tools”, and “Metrics”.
- The fourth level is the alternatives level, which is not created; this study did not involve decision-making among multiple alternatives.

Figure 4. 1 - AHP structure for DevSecOps challenges

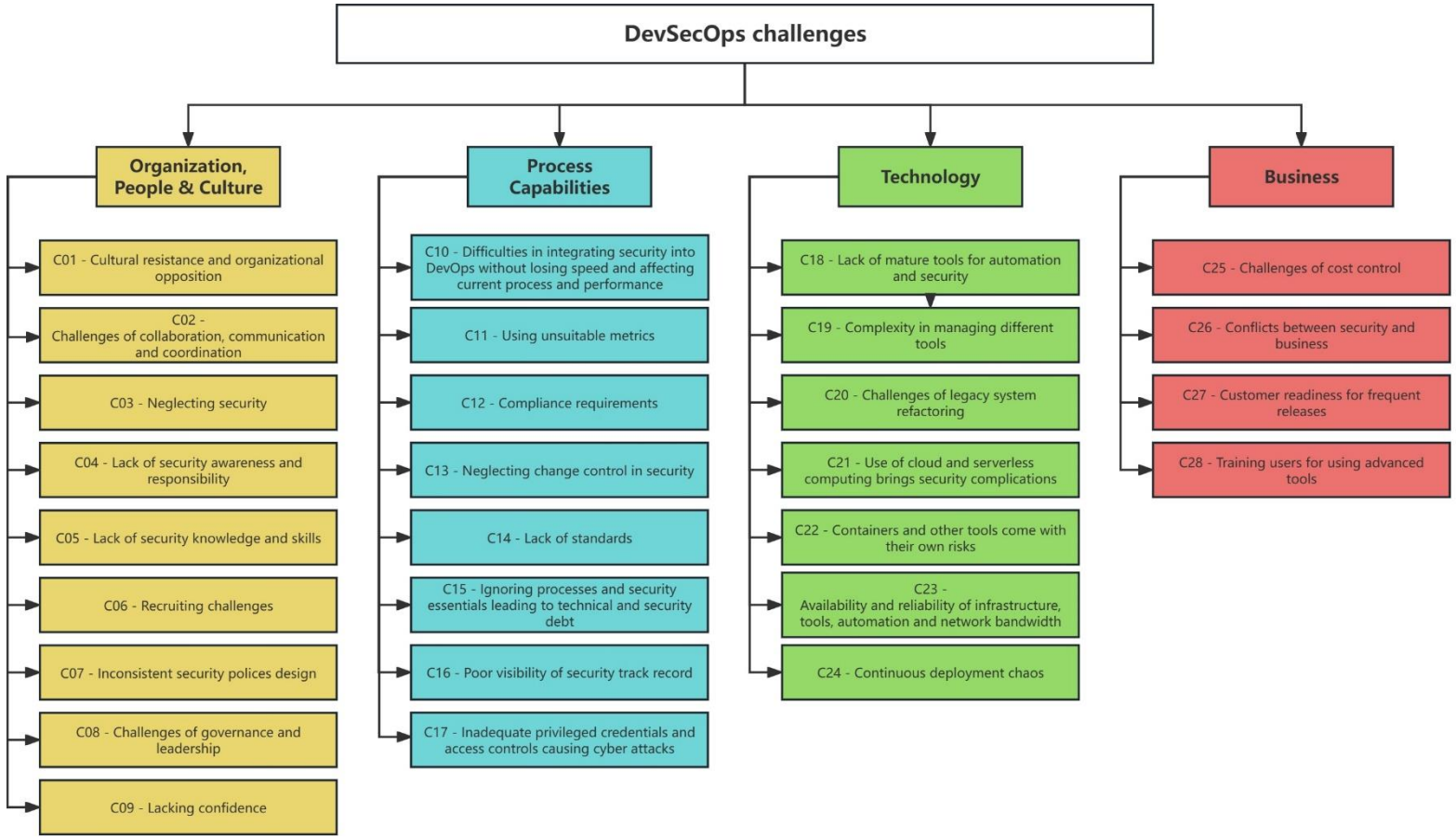


Figure 4. 2 - AHP structure for DevSecOps practices

Figure 4. 3 - AHP structure for DevSecOps tools

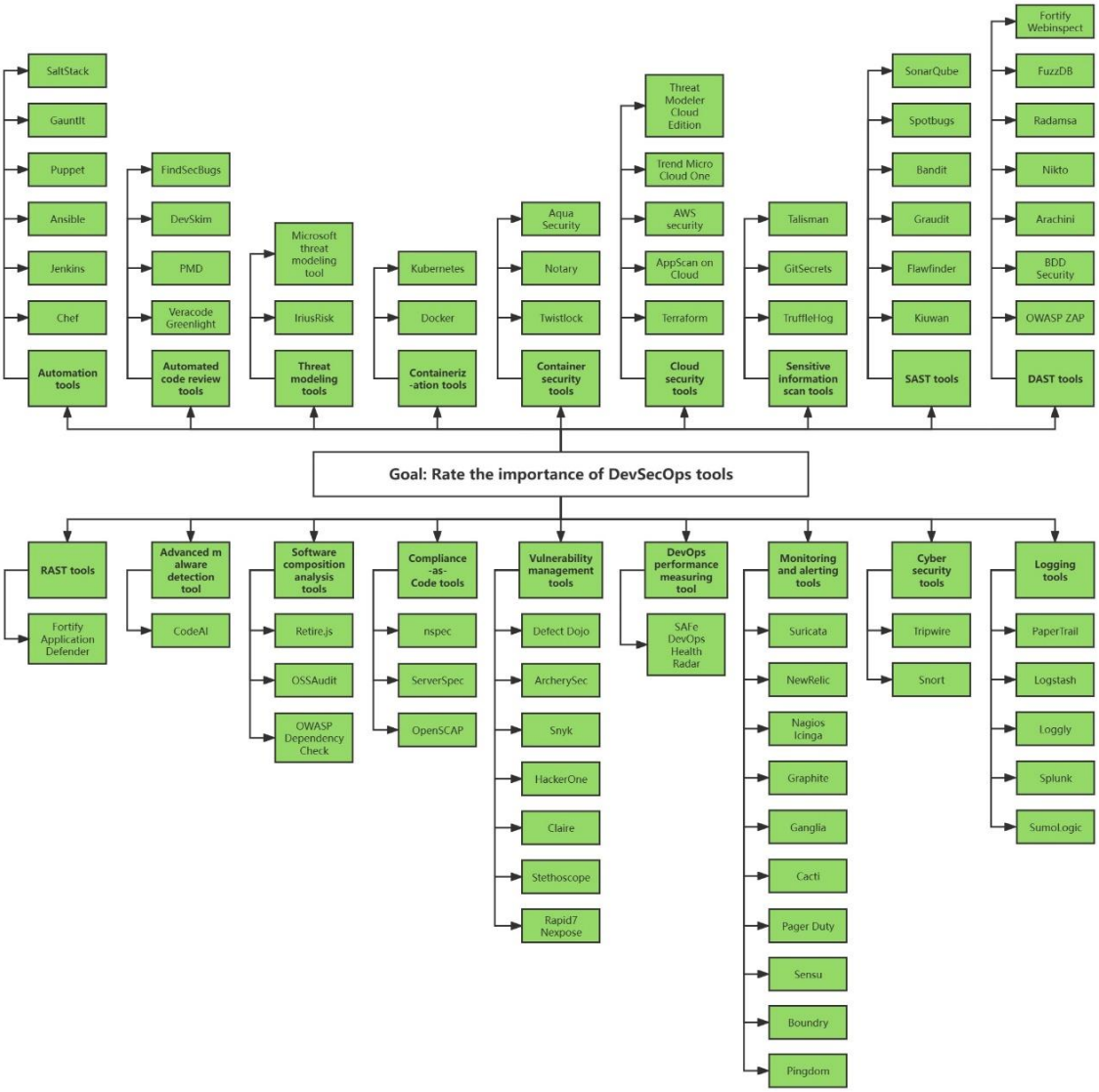
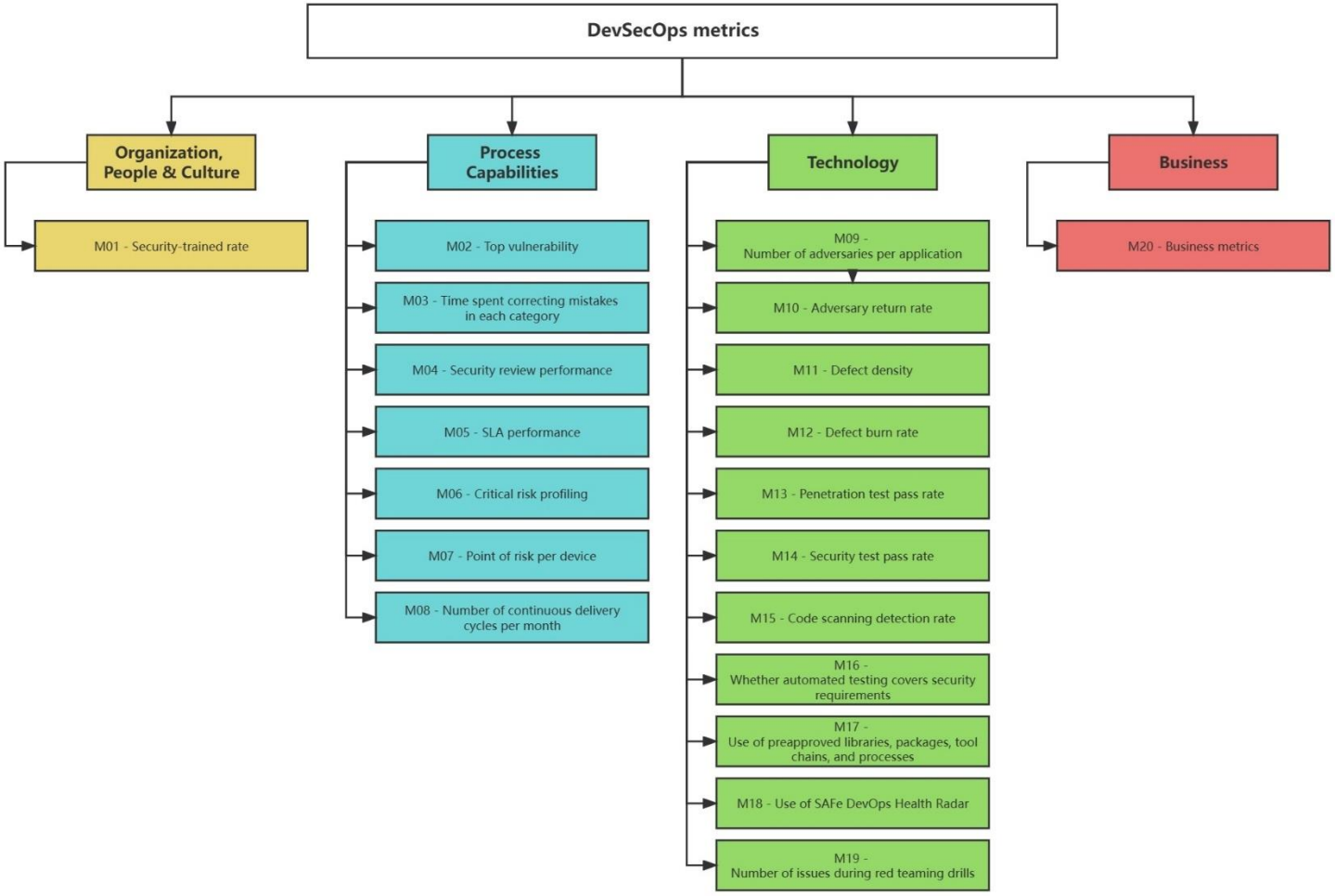


Figure 4. 4 - AHP structure for DevSecOps metrics



4.3.2.2 Step (2): Derive priorities/weights for the criteria and sub-criteria

After building the hierarchical structure, the relative priorities/weights for the criteria (four categories) and sub-criteria (C, P, T, M) were obtained through pairwise comparisons. They were referred to "relative priorities" because they were measured with respect to each other (Mu & Pereyra-Rojas, 2017). These priorities were quantified by using a standard 9-point AHP Scale (Saaty, 2013), as shown in **Table 4. 1**. The reciprocal value was used to quantify the priority of the other observation, which was compared with, and it was equal to one over the numerical score.

Table 4. 1 - 9-point AHP comparison scale

Scale	Numerical score	Reciprocal
Equally important	1	1
Moderately more important	3	1/3
Strongly more important	5	1/5
Very strongly more important	7	1/7
Extremely more important	9	1/9
Intermediate values	2, 4, 6, 8	1/2, 1/4, 1/6, 1/8

The AHP approach solves complex decision-making problems based on pairwise comparison matrices (Brunelli, 2015). Assuming that a decision maker has to consider a set of criteria c_1, c_2, …, c_n, which have characteristics making one alternative preferable to another with respect to a given goal; and each criterion has its weight w_1, w_2, …, w_n, respectively. The pairwise comparisons can be collected in a pairwise comparison matrix **A**:

Equation (1)

$$\mathbf{A} = (a_{ij})_{n\times n} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix},$$

where $a_{ij} > 0$ expressing the degree of w_i to w_j. Each entry is supposed to approximate the ratio between two weights: $a_{ij} \approx \frac{w_i}{w_j}, \forall i, j$. Thus, the matrix **A** can be expressed as:

Equation (2)

$$\mathbf{A} = \left(\frac{w_i}{w_j}\right)_{n\times n} = \begin{pmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{pmatrix}.$$

There is a condition of multiplicative reciprocity: $a_{ij} = \frac{1}{a_{ji}}$, $\forall i, j$, in this case, the matrix **A** can be rewritten as **Equation (3)**, which illustrates the pairwise relative priorities for the criteria.

Equation (3)

$$\mathbf{A} = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ \frac{1}{a_{12}} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{1n}} & \frac{1}{a_{2n}} & \cdots & 1 \end{pmatrix}.$$

The priority/weight for each criterion is determined using the priority/weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$, which is calculated using the Eigenvector method (Saaty, 2013). Multiplying **Equation (2)** by the matrix **w** obtains **Equation (4)**:

Equation (4)

$$\mathbf{Aw} = \begin{pmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} nw_1 \\ nw_2 \\ \vdots \\ nw_n \end{pmatrix} = n\mathbf{w},$$

where n and **w** are an eigenvalue and an eigenvector of **A**, respectively.

According to Saaty (2013), **Equation (4)** can be extended to all pairwise comparison matrices by replacing n with λ_{max}, when n is the largest eigenvalue of **A**.

Thus, the priority/weight vector **w** can be obtained from any pairwise comparison matrix **A** as the solution of following equation set **Equation (5)**:

Equation (5)

$$\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^T\mathbf{1} = 1 \end{cases},$$

where λ_{max} is the maximum eigenvalue of A, and $1 = (1, \ldots, 1)^T$.

Although various computational programs can substitute for manual calculations, it is still important to understand the fundamentals of the AHP method.

4.3.2.3 Step (3): Perform consistency analysis

In this step, the consistency of the pairwise comparison matrix was evaluated by considering the consistency ratio (CR) (Saaty, 2013). AHP calculates a Consistency Ratio (CR) by comparing the Consistency Index (CI) of the matrix versus the Consistency Index of a Random-like Matrix (RI), as shown in **Equation (6)**:

Equation (6)

$$CR = \frac{CI}{RI} .$$

The value of CI can be obtained using **Equation (7)**:

Equation (7)

$$CI = \frac{\lambda_{max} - n}{n-1},$$

where λ_{max} is the largest eigenvalue of the matrix **A** and n is the size of the matrix **A**. The value of RI depends on the matrix size n. Saaty (2013) provides the RI values for matrices of different sizes, as listed in **Table 4. 2**.

Table 4. 2 - RI values for the matrices of different sizes

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49	1.52	1.54	1.56	1.58	1.59

According to Saaty (1982), ideally, the pairwise comparison matrix is considered entirely consistent if the CR value is zero. The “reasonable” value of CR is lower than 0.2, and the “acceptable” value is lower than 0.1. It is suggested that the pairwise comparison process should be refined and repeated if its CR exceeds 0.2. This suggestion was given in 1982, when the AHP approach initially emerged. Over the years, the majority of research uses the acceptable value 0.1 as the borderline (Saaty, 2013); exceptions are preferring 0.2, e.g., (Kim et al., 2022). On the contrary, Ishizaka and Lusti (2004) argue that evaluating CR is unnecessary, as consistency cannot be achieved through human subjective judgment.

Some researchers refer to CR as “Inconsistency Ratio” rather than “Consistency Ratio”, since higher values indicate lower consistency. However, while expressed differently, the interpretation of the results ought to be the same.

In this research, the term “Consistency Ratio (CR)” was used, and the acceptable value of 0.1 was

employed to evaluate consistency, with sporadic values between 0.1 and 0.2 tolerated. This study combined the AHP method with the Delphi survey. Following the Delphi method, this research employed multiple rounds of surveys, which allowed participants and the researcher to amend and repeat the pairwise comparison process if the CR was deemed unacceptable. The survey results and CR from each round would be communicated to the participants to help them complete subsequent rounds more accurately.

4.3.2.4 AHP Tool – SuperDecisions

With the widespread adoption of AHP across numerous fields and disciplines, an increasing number of software and tools have been developed to analyse, synthesise, and justify complex decisions. In this study, SuperDecisions (2023) was selected as the AHP tool due to its user-friendly design, helpful tutorials, technical support, and free accessibility. This tool enabled the researcher to make comparisons between elements (nodes) by generating an AHP comparison matrix, a questionnaire, or a graph, thereby saving calculation time and reducing the risk of errors.

Refer to the AHP design in **Figure 4. 1**, **Figure 4. 2**, **Figure 4. 3**, and **Figure 4. 4**, SuperDecisions was employed to decompose the DevSecOps CPTM Model (Version 1.0) into hierarchical structures, as shown in **Figure 4. 5**, **Figure 4. 6**, **Figure 4. 7**, and **Figure 4. 8**. The alternatives level was not created, as this study did not involve decision-making among multiple alternatives.

Figure 4. 5 - AHP structure for DevSecOps challenges level by SuperDecisions

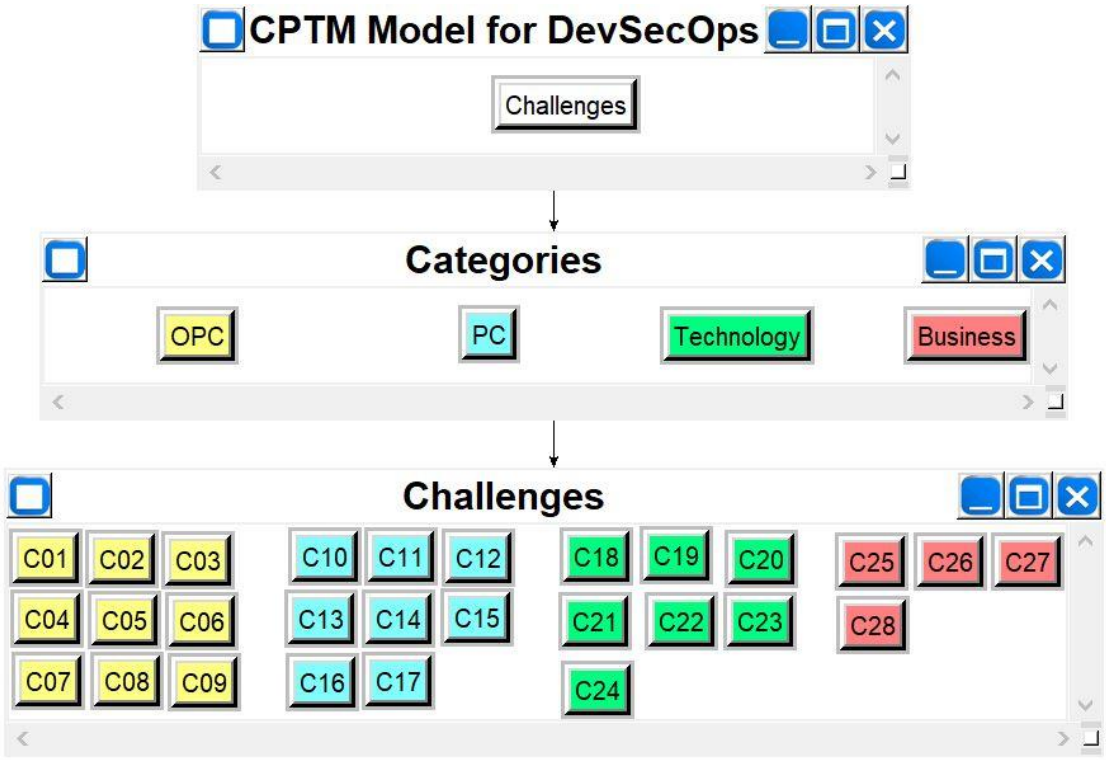


Figure 4. 6 - AHP structure for DevSecOps practices level by SuperDecisions

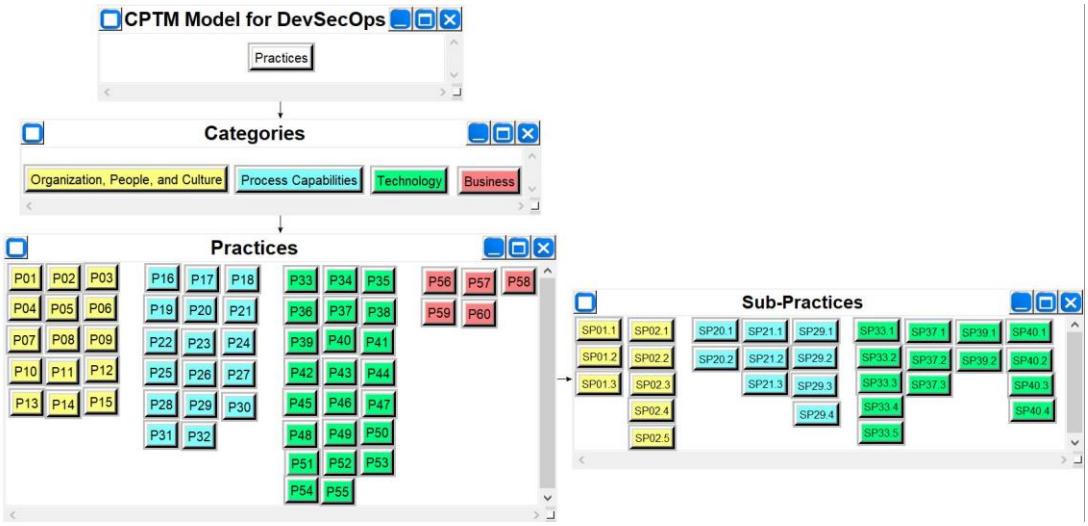


Figure 4. 7 - AHP structure for DevSecOps tools level by SuperDecisions

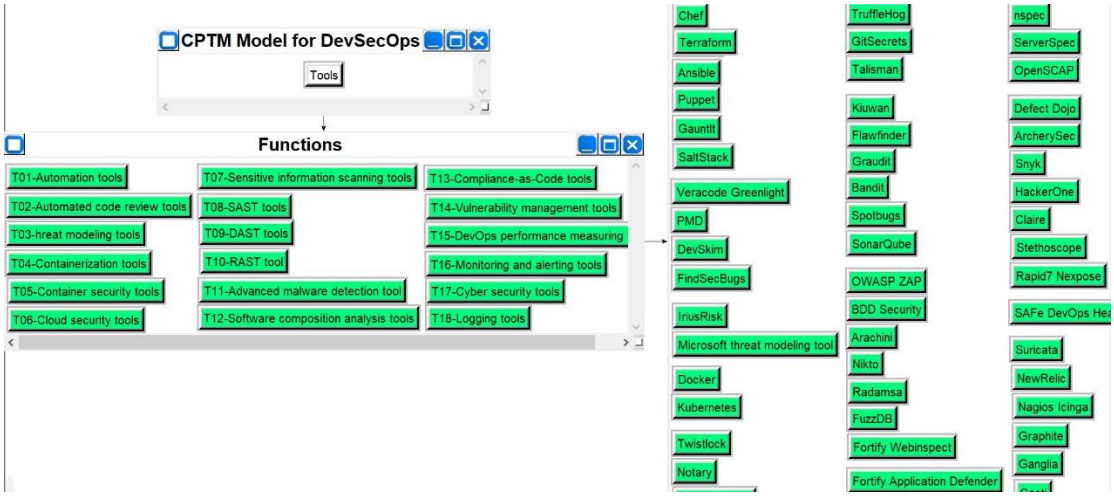
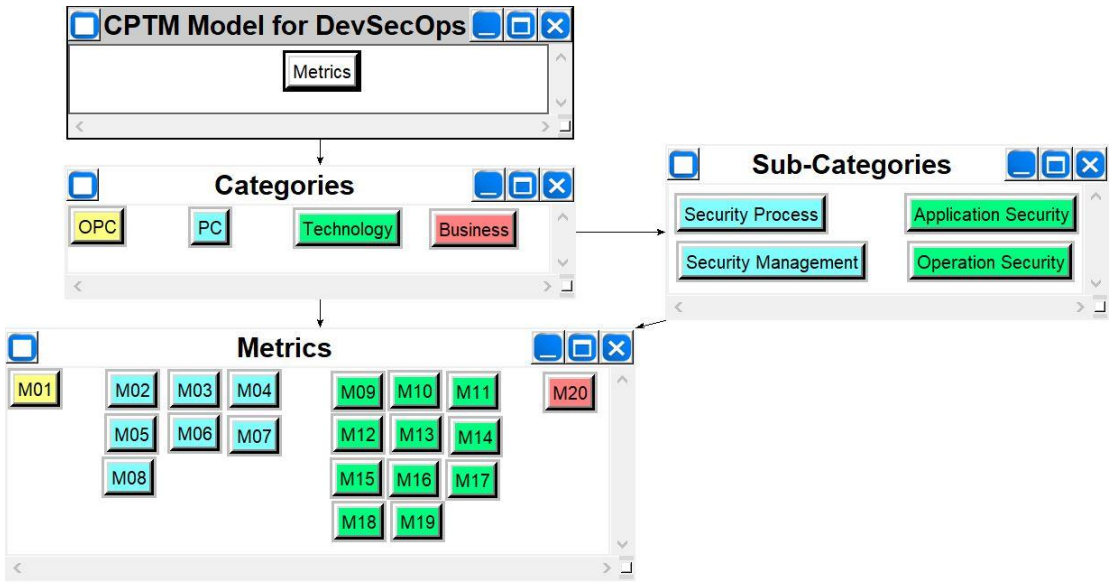


Figure 4. 8 - AHP structure for DevSecOps metrics level by SuperDecisions



4.3.2.5 Minimum Number of Comparisons

When the questionnaire included a large number of AHP pairwise comparisons/questions, the minimum number of comparisons was used to evaluate the importances, rather than completing all of them. For instance, the second round of the Delphi survey minimised the number of AHP comparisons by avoiding the repetition of evaluating DevSecOps challenges and by limiting the large volume of identified DevSecOps practices to be compared. Minimising the number of AHP comparisons eliminated redundant questions, reduced unnecessary workloads and survey durations for participants, and mitigated inconsistencies caused by redundant pairwise

comparisons (Mu & Pereyra-Rojas, 2017).

Table 4. 3 provides an example of comparing DevSecOps practices, the minimum number of comparisons is comprised of only the comparison judgments in the diagonal (shaded) above the unit diagonal (filled with 1s) of the comparison matrix, e.g., $\frac{P01}{P02}$, $\frac{P02}{P03}$, $\frac{P03}{P04}$, $\frac{P04}{P05}$ and $\frac{P05}{P06}$ (Mu & Pereyra-Rojas, 2017). Once the participants have made these comparisons, the researcher can calculate the remaining comparison judgments in the upper and lower parts of the matrix. Assuming that values of $\frac{P01}{P02} = \frac{1}{3}$ and $\frac{P02}{P03} = \frac{1}{2}$, then one can calculate the value of $\frac{P01}{P03} = \frac{P01}{P02} * \frac{P02}{P03} = \frac{1}{3} * \frac{1}{2} = \frac{1}{6}$. For $\frac{P03}{P01}$, it will be the reciprocal of the value $\frac{P01}{P03} = 6$. The exact process can be repeated for the remaining cells in the matrix.

Table 4. 3 - Minimising the number of AHP pairwise comparisons

AHP pairwise comparison	P01	P02	P03	P04	P05	P06
P01	1	P01/P02				
P02		1	P02/P03			
P03			1	P03/P04		
P04				1	P04/P05	
P05					1	P05/P06
P06						1

4.4 Reflection on the Literature

This section reflects related research to this empirical investigation, including previous Delphi and survey-based studies, as well as studies that have applied the AHP method in conjunction with DevSecOps or DevOps research. A reflection on the literature regarding the combination of Delphi and AHP in other research domains is also presented. The learning goals from this reflection are to provide insights and understanding into the relevance, procedure, and application of these two methods to enable a reliable research design.

4.4.1 Delphi or Survey Studies on DevSecOps

In the existing literature, no academic papers have presented Delphi studies on DevSecOps. However, some traditional questionnaire-based survey studies on this topic have been identified, all of which are follow-up studies aimed at validating and refining the findings of review studies.

Rahman and Williams (2016) identified sets of DevOps activities and security practices through a grey literature review (GLR) and surveyed nine practitioners to assess the practicality of the findings in industry. Rafi et al. (2020) identified eighteen security challenges in DevOps through an SLR and surveyed industrial and academic experts to prioritise and develop a taxonomy of these challenges. Akbar, Rafi, et al. (2022) surveyed practitioners to assess eighteen DevSecOps challenges and ten categories identified in their MLR, and the results showed that their findings were relevant to the industry. Mao et al. (2020) presented a GLR study to identify sets of DevSecOps challenges and practices. As a follow-up to the GLR, Zhou et al. (2023) surveyed DevSecOps practitioners in the Chinese SE industry to summarise the current state of DevSecOps in practice.

The key learning from these studies is that the complexity of the survey questionnaire tends to be inversely proportional to the number of participants. For example, Rahman and Williams (2016) used a complex questionnaire to evaluate a large number of DevSecOps practices. Their survey questionnaire included both open-ended questions and closed-ended questions (i.e., Likert scales and multiple-choice), which increased its complexity. Only nine participants completed their survey, which was a relatively small sample size for a standard survey study. In contrast, Zhou et al. (2023) designed a straightforward questionnaire comprising eleven multiple-choice questions and one fill-in question, enabling them to collect 239 usable responses within 45 days. It was also the case that China's industrial population was vast.

4.4.2 Delphi or Survey Studies on DevOps

In the literature, there have been no journal articles and conference papers that presented Delphi studies on DevOps, perhaps due to its long-term, complex nature. However, a doctoral thesis (Greene, 2020) presented a Delphi study to explore various aspects of the perceived importance of developing a governance model for DevOps teams.

The author conducted four rounds of a Delphi survey with eighteen DevOps experts from a large United States manufacturing organisation. That Delphi study employed multiple question formats and data analytical methods. In Round One, the participants answered eight open-ended questions, and the data from this round were coded and themed; in Round Two, the participants ranked the themes using a 3-point Likert Scale; in Round Three, the participants reached a consensus on the

identified themes; and in Round Four, the participants assessed the themes again by using a 5-point Likert Scale.

The lesson learned from the PhD research is that the Delphi method is a pragmatic approach that can be customised to varied question formats and adapted to several data analytical methods across different survey rounds as needed.

4.4.3 AHP on DevSecOps and DevOps

Although there are no existing academic papers that have adopted the AHP method on the DevSecOps topic, some high-quality papers were captured, which prioritised various aspects of DevOps using the AHP or fuzzy AHP method, or prioritised various aspects of DevSecOps/DevOps using other decision-making approaches, such as Ranking Organisation Method for Enrichment Evaluation (PROMETHEE), Technique for Order Preference by Similarity to an ideal Solution (TOPSIS), and their variants.

Khan and Shameem (2020) employed the AHP method to calculate the prioritisation weights for sixteen DevOps challenging factors and rank them. Rafi et al. (2020) used PROMETHEE-II to prioritise and develop the taxonomy of eighteen DevOps security challenges. Zohaib, Alsanad and Abdullah Alhogail (2024) used fuzzy AHP to prioritise 48 identified DevOps implementation guidelines. Akbar and his colleagues presented three studies: prioritising DevOps success factors (Akbar et al., 2020) and DevOps practices (Akbar, Rafi, et al., 2022) using fuzzy AHP, and assessing DevSecOps challenges (Akbar, Smolander, et al., 2022) using fuzzy TOPSIS.

These studies share a similar research methodology comprising three stages: an SLR or MLR study to identify sets of findings; an ordinary survey to verify the findings; and a pairwise-comparison-based survey (e.g., AHP, PROMETHEE, or TOPSIS) to prioritise the findings.

In contrast, Noorani et al. (2022) identified some DevOps factors through an SLR and categorised the findings into a SWOT (Strengths, Weaknesses, Opportunities, Threats) framework. They subsequently conducted an AHP-based survey rather than undertaking an ordinary survey first. The reason was that the authors had already built their framework before their survey stage, and they believed that AHP was suitable for assessing their findings (i.e., a multi-layered framework).

The same was true for this research: a conceptual framework (i.e., the DevSecOps CPTM Model

Version 1.0) had already been drafted before the survey stage, with a hierarchical structure and a large number of elements. Therefore, the Delphi-AHP-based survey was directly conducted.

4.4.4 Combination of Delphi and AHP Methods

To the best of knowledge, the combination of Delphi and AHP represents a bold and new attempt at an empirical investigation in the SE domain. However, the literature contains evidence of many successful applications of this hybrid-method approach across other domains.

Kim, Jang and Lee (2013) presented the application of Delphi-AHP method to select the priorities of waste electrical and electronic equipment; Di Zio and Maretti (2013) employed Delphi-AHP method to assess the acceptability of energy sources; Md Arof (2015) reviewed eight applications of a combined Delphi-AHP method in maritime transport research; Bouzon et al. (2016) used fuzzy Delphi-AHP method to identify and analyse reverse logistics barriers; Kim et al. (2022) presented a probabilistic tunnel collapse risk evaluation model by using Delphi-AHP method; Zhao, Md Ali and Ahmad (2023) combined Delphi method with AHP to develop the indicators for sustainable urban regeneration in historic urban areas.

By synthesising the reflection on the literature and the experience from this research, it can be summarised that the combination of Delphi and AHP is suitable for the research, which:

- Has limited expert availability or involves geographically dispersed experts, but still pursues credible results.
- Validates sets of findings or outcomes of review studies on a controversial subject with limited information.
- Address a complex problem or a layered system that can be hierarchically structured.
- Tolerates a long period of surveying time and research duration.

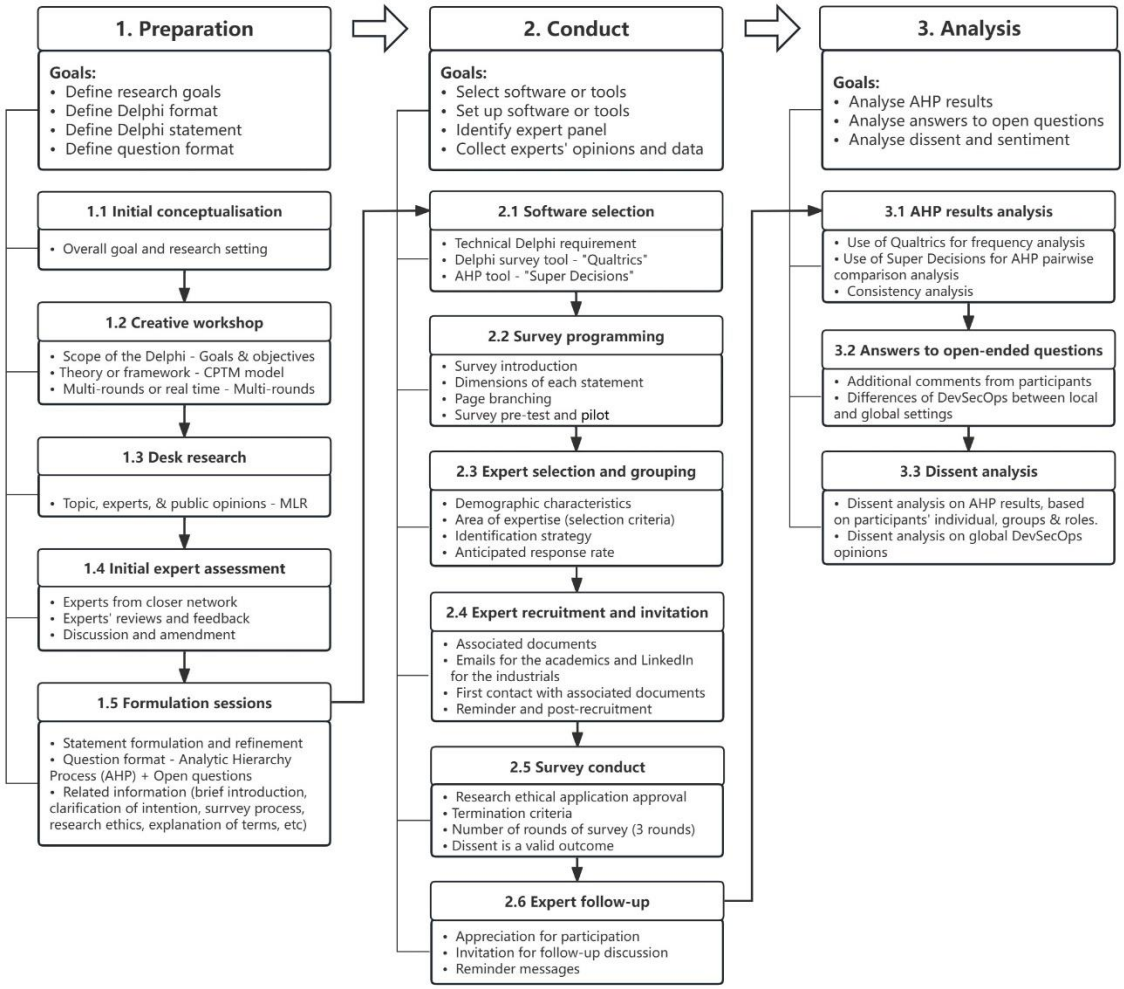
4.5 Research Design and Implementation

Although this empirical investigation employed a hybrid-method approach, its core remained a Delphi survey. According to Beiderbeck et al. (2021), a Delphi process consisted of three phases: Preparation, Conduct, and Analysis. Under each phase, a set of steps was included. Beiderbeck et

al. (2021) designed the Delphi process as a standard process for all disciplines, so it was adjusted by adding or removing specific steps based on the needs and real situations. In this case, the Delphi survey was combined with the AHP method to structure the complex problem hierarchically, prioritise the criteria, show the relationships between criteria and alternatives, and make the final decision based on prioritisation.

Figure 4. 9 depicts the Delphi process adapted from the guidelines by Beiderbeck et al. (2021), including its phases, goals, and steps. Appendix B.7, "Delphi Plan" (Page 324), provides a more detailed table created at the beginning of the survey stage to guide the Delphi implementation, including three main phases, the goals of each phase, the steps under each phase, and the estimated time for each step and goal.

Figure 4. 9 - Delphi study process



4.5.1 Preparation Phase

The goals of the Preparation phase were to:

- Define research goals: *To conduct a Delphi study to validate, refine, and improve the findings of the previous MLR study by soliciting DevSecOps and GSE experts' opinions on the DevSecOps CPTM Model (Version 1.0) and its adaptation in a global setting*.
- Define Delphi format: *Scope of the Delphi, Theory/Framework, and Multiple rounds of conduct*.
- Define Delphi statement: *supporting documents, e.g., Ethical Application, Information Sheet, and Explanation of Terms*.
- Define question formats: *Pairwise comparisons in Analytic Hierarchy Process (AHP), plus a few multiple-choice and Open-ended questions*.

4.5.1.1 Initial Conceptualisation

A Delphi study started with an initial conceptualisation to define the overall research goal (Beiderbeck et al., 2021). As presented in Chapter 3, an MLR study was conducted to provide a comprehensive review of DevSecOps by reporting new findings from academia and industry and building a conceptual framework for DevSecOps adoption based on these findings. A set of research objectives and questions of the MLR study was established: (They are also termed "Review objectives" and "Review questions", to distinguish them from the research objectives and questions of this empirical Delphi-AHP study.)

Research/Review Objectives:

- To explore the current state of DevSecOps in the existing white and grey literature.
- To explore the adoption of DevSecOps in GSE from the existing white and grey literature.
- To establish a conceptual framework of DevSecOps based on the existing literature.

Research/Review Questions:

- ***RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect, and their links) in the existing white and grey literature?***

Sub-question 1.1: What aspects of DevSecOps can be found in the existing white and grey literature?

Sub-question 1.2: What themes do these aspects contain?

Sub-question 1.3: How do the identified aspects and themes link to each other?

- ***RQ2: How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?***

To address RQ1, the MLR identifies five aspects of DevSecOps research (Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement), collects related themes of each aspect, and generates a DevSecOps CPTM Model (Version 1.0) by integrating the themes of the latter four aspects. To address RQ2, an unexplored area relating to the application of DevSecOps in GSE has been identified.

Subsequently, the Delphi-AHP study aimed to evaluate and validate the MLR findings and further refine and upgrade the DevSecOps CPTM Model from Version 1.0 to Version 2.0 by incorporating additional GSE aspects. Hence, a new set of research objectives and questions was established as follows:

Research Objectives:

- To validate and refine the conceptual framework through an empirical investigation.
- To investigate differences between DevSecOps in local and global settings and further upgrade the drafted conceptual framework into a global version that could guide practitioners adopting the DevSecOps approach to support software engineering practices in a GSE setting.
- To investigate whether a consensus or dissent exists on the DevSecOps approach between the SE industry and academia.

Research Questions:

- ***RQ3: How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?***

  Sub-question 3.1: What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?

Sub-question 3.2: Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?

- ***RQ4: What are the experts' opinions on DevSecOps in GSE contexts?***

  Sub-question 4.1: How is DevSecOps different between local and global settings?

  Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?

Both research questions continued the two review questions from the previous MLR. RQ3 was for the evaluation and prioritisation of the identified challenges, practices, tools, and metrics of DevSecOps. RQ4 was to verify and supplement the differences between local and global DevSecOps. The integration of RQ3 and RQ4 could refine and enhance the DevSecOps CPTM Model into a comprehensive global version (Version 2.0). Additionally, the dissent analysis (i.e., Sub-question 3.1) aimed to investigate whether there is consensus or dissent regarding experts' understanding of DevSecOps.

4.5.1.2 Creative Workshops

Creative workshops were carried out between two supervisors and the researcher/PhD student (Gavin Zhao) to define the format of the Delphi survey, including scope, theory/framework, and format (multi-rounds or real-time).

Delphi Scope:

To align with the research topics (DevOps, DevSecOps, and GSE) and the previous MLR study, the scope of this Delphi survey was defined as the experts from academia and industry with specific expertise, experience, and authority in Software Engineering (SE) and Information Security (IS).

Theory/framework:

The DevSecOps CPTM Model (Version 1.0) established in the previous MLR study served as the theory/framework for this Delphi-AHP study. The original version of the model still had some gaps that needed to be filled, e.g., the absence of global aspects and the gap between identified tools and practices. Hence, the Delphi-AHP study was undertaken to evaluate and validate the

MLR findings and further refine and upgrade the DevSecOps CPTM Model from Version 1.0 to Version 2.0 by incorporating additional GSE aspects.

Delphi Format:

Delphi has two formats: multi-round and real-time (Beiderbeck et al., 2021). A traditional multi-round Delphi was decided upon rather than a real-time one. This Delphi survey, combined with the AHP method, was complex and time-consuming, and the number of participants was relatively small (approximately 18). By contrast, the real-time format was considered appropriate for surveys involving a larger number of participants (hundreds) and addressing a relatively simple question within a shorter time (Aengenheyster et al., 2017; Beiderbeck et al., 2021; Donohoe, Stellefson, & Tennant, 2012).

4.5.1.3 Desk Research

Desk research was the process of gathering and analysing the existing information from secondary sources, rather than collecting new data firsthand (Beiderbeck et al., 2021). Before the Delphi-AHP study, a literature review (Chapter 2) and an MLR study (Chapter 3) on DevSecOps and its adoption in Global Software Engineering (GSE) contexts were conducted as desk research to identify the recent research and practical trends. Regarding the research methodology, the literature on the Delphi and AHP methods was also reviewed. Hence, a solid foundation for the Delphi-AHP study and the entire doctoral research was established.

4.5.1.4 Initial Expert Assessment

The final research proposal for this research (PGR9) and the MLR findings were initially assessed by two experts at AUT. In addition, the MLR study (Zhao, Clear, & Lal, 2024b) and its review protocol (Zhao, Clear, & Lal, 2024a) were peer-reviewed by three reviewers and published in the Journal of Systems and Software, a Q1-ranked journal. Hence, the research proposal and previous work were discussed with those reviewers and experts, and were revised based on their valuable feedback and suggestions.

4.5.1.5 Formulation Sessions

This step involved defining the Delphi statement and formulating the survey question format (Beiderbeck et al., 2021).

The Delphi statement was defined by some supporting documents, including:

- A Research Ethics Statement, which was approved by the Auckland University of Technology Ethics Committee (Appendix A on Page 294)
- An Information Sheet that provided a brief description of the Delphi study (Appendix B.2 on Page 306)
- A Consent Form that provided for participants' action (Appendix B.3 on Page 311).
- An Introductory Letter that provided a self-introduction and a brief description of the research to make initial contact with potential participants (Appendix B.4 on Page 312).
- An Explanation of Terms that defined all technical terms in the research (Appendix E.2 on Page 379).

The question format consisted of both closed-ended and open-ended questions. Although the questionnaire content varied across the three rounds of the survey, the majority of questions remained in a closed-ended format, i.e., the AHP pairwise-comparison questions.

To address RQ3 regarding the DevSecOps CPTM Model, the Analytic Hierarchy Process (AHP) method was employed to prioritise all elements within the model's associated categories through pairwise comparisons. One open-ended question asked participants to add comments to the findings.

To address RQ4 regarding DevSecOps in GSE, a multiple-choice question "how do you think DevSecOps differs in local and global settings" was raised, containing four options: "Extremely different", "Slightly different", "Not different", and "Uncertain". There was an additional open-ended question that allowed participants to leave comments if they believed that local and global DevSecOps are different.

4.5.2 Conduct Phase

The goals of the Conduct phase were to:

- Select tools: *SuperDecisions for AHP; Qualtrics for Delphi*.

- Set up the software or tools: *AHP pairwise comparisons on SuperDecisions; Delphi survey programming on Qualtrics*.
- Identify the expert panel: *18 experts as participants for the study*.
- Collect experts' opinions and data: *Three survey rounds*.

4.5.2.1 Software Selection

As explained previously in Section 4.3.2.4 (Page 75), SuperDecisions (SuperDecisions, 2023) was selected for structuring the AHP, making comparisons, generating a questionnaire, and analysing AHP results. Meanwhile, Qualtrics (Qualtrics, 2024) was selected as the online survey tool to release questionnaires and receive participants' responses.

Qualtrics is a platform for conducting online surveys. It provides a wide range of formats and approaches, from simple questionnaires to sophisticated social science research instruments with extensive scripting, randomisation, and interactive question presentation (Qualtrics, 2024). Qualtrics enables users to conduct a variety of online data collection and analysis activities, including market research, customer satisfaction surveys, product and concept testing, employee evaluations, and website feedback (Qualtrics, 2024).

With respect to security and privacy, Qualtrics complies with applicable data privacy laws, including the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). It has enterprise security features to ensure the safety, security, and trustworthiness of user and respondent data (Qualtrics, 2024).

4.5.2.2 Survey Programming

The AHP tool SuperDecisions was used to generate the pairwise comparison questionnaires. Subsequently, the generated questionnaires were transferred to the Qualtrics platform for the survey programming. Qualtrics (2024) supports various question types (e.g., multiple choice, text entry, slider, matrix table, form field, rank order, side-by-side, etc.). In this study, a bipolar matrix table was selected for matching the pairwise comparison matrices generated by SuperDecisions.

After survey programming, the preview survey was pre-tested internally and improved. A pilot round of the survey was conducted with three experts from a close network (e.g., at AUT) to test

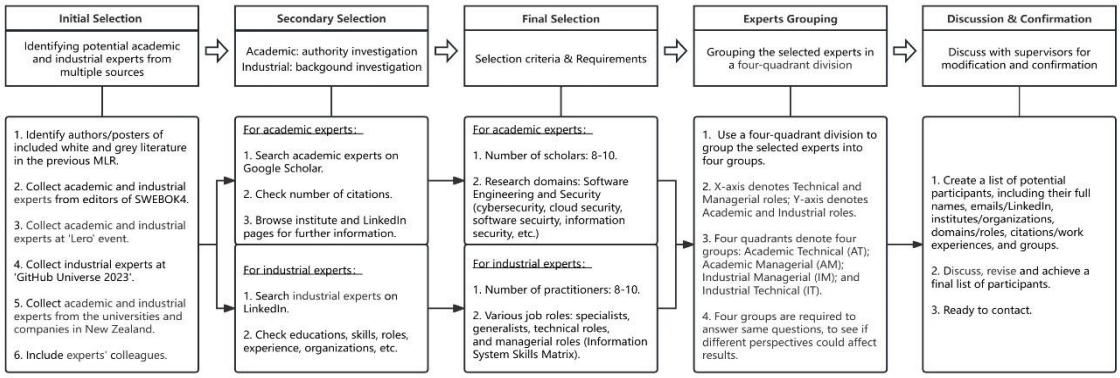
the usability. Eventually, the first round of the Delphi survey was released to participants.

4.5.2.3 Expert Selection and Grouping

To conduct the Delphi-AHP study based on the MLR findings, potential experts/participants were identified and selected from academia and industry to achieve the research aim. In this thesis, 'experts' refer to all individuals with expertise in DevSecOps before they accepted the survey invitation. 'Participants' refer to the experts who participated in the Delphi survey.

Figure 4. 10 depicts the expert selection process, which consists of five steps: Initial Selection, Secondary Selection, Final Selection, Experts Grouping, and Discussion & Confirmation.

Figure 4. 10 - Expert selection process



Step (1): Initial Selection:

In the initial selection, a large number of potential experts were identified from multiple sources:

- the authors of the selected papers for the MLR study, providing access to a global set of researchers and practitioners;
- the editors of the book "Guide to the Software Engineering Body of Knowledge" aka "SWEBOK Guide V4.0" (Washizaki, 2024);
- the participants at an event 'Frontiers of Technology: Cybersecurity – Achievements and Challenges', at Lero, University of Limerick;
- the industrial presenters in GitHub Universe 2023 (Chakrabarty et al., 2023);
- New Zealand-based universities and companies;

- and the colleagues or acquaintances of the identified experts.

First, hundreds of authors' and posters' names were extracted, including duplicates, from the academic papers and grey literature sources used in the MLR study. For the academic papers, a limit of six authors per paper was applied. After extraction, the frequency of each name was counted, and the results were printed and sorted in descending order. The "Namesake" problem was resolved through manual inspection. As a result, 35 academic authors and 18 industrial authors/posters were identified. The number of identified industrial experts was relatively low because some GL articles either did not include the authors/posters or lacked reliable information. As listed in Appendix B.5, "Initial Potential Experts Selection from MLR" (Page 313), the identified authors were further grouped into three tiers. Academic authors were identified and grouped by frequency (>= 2); industrial authors were grouped by work experience (years).

The high refusal rate and potential dropout rate are inevitable problems with the Delphi study. Hence, it was necessary to identify and invite as many potential experts as possible. For example, nine academic and five industrial experts were identified by using the book "Guide to the Software Engineering Body of Knowledge" (Washizaki, 2024). The editors of the chapters or sub-chapters related to the research topic, such as "Software Security" and "Software Testing", were identified as potential participants. Thirty industrial experts were identified among the presenters at GitHub Universe 2023 (Chakrabarty et al., 2023). Three academic experts were identified from the conferees at "Frontiers of Technology: Cybersecurity – Achievements and Challenges" at Lero, University of Limerick. Five scholars from New Zealand universities and three practitioners from NZ-based companies were identified. Some identified experts could not participate but referred to their colleagues or acquaintances as suitable potential participants. **Table 4. 4** summarises the number and sources of experts identified during the initial selection.

Table 4. 4 - Potential experts identified from different sources

Expert sources	Number of identified experts
Authors of selected papers in MLR	35 academic and 18 industrial experts
Editors of "SWEBOK4"	9 academic and 3 industrial experts
Participants at 'Lero event'	3 academic and 0 industrial experts
Presenters in 'GitHub Universe 2023'	0 academic and 30 industrial experts
Scholars in NZ universities and practitioners from NZ companies	5 academic and 5 industrial experts
Colleagues/acquaintances of experts	3 academic and 4 industrial experts

Step (2): Secondary Selection:

The secondary selection was divided into two routes: one to evaluate the authority of the selected academic authors, and the other to investigate the background of the selected industrial posters. For academic experts, names were searched for on Google Scholar, and citation counts were used as the primary indicator of authority. In addition to authority, the relevance of their studies to this research topic was considered; higher relevance may require lower authority.

Based on the results of the initial selection, authors who met one of the following criteria were included:

- Higher relevance plus over 300 citations.
- Moderate relevance plus over 500 citations.
- Lower relevance plus over 1000 citations.

For the industrial experts, LinkedIn was used to identify their biographies, including the skills, positions/roles, organisations, and work experience (see Appendix B.5, "Initial Potential Experts Selection from MLR" on Page 313).

Step (3): Final Selection:

In the final selection, a set of expert selection criteria and requirements was defined, which included the number of experts, professional experience, working fields, and domains of expertise (Weiser et al., 2022), as shown in **Table 4. 5**.

Table 4. 5 - Expert selection criteria

Criteria	Requirements
Number of experts	16-20 experts.
Professional experience	Minimum 5 years.
Field of work	• Academia: 8-10 experts. • Industry: 8-10 experts. • Non-profit organisations (NPOs): No formal number required. • Overlapped fields: No formal number required.
Domain of expertise	Academia: Software Engineering and Security Industry: various job roles in the Information System Skills Matrix (specialist, generalists, technical roles, and managerial roles) (Weiser et al., 2022)

These criteria were applied to initial and secondary selections to group the selected experts. 16 to 20 experts were expected to be identified, half from academia and half from industry. The panel size for Delphi surveys is typically set between 15 and 30 (Lilja, Laakso, & Palomäki, 2011). However, it can vary with the problem's scope and available resources (Powell, 2003). Given the high refusal and potential dropout rates, it was necessary to identify as many experts as possible at this stage of selection.

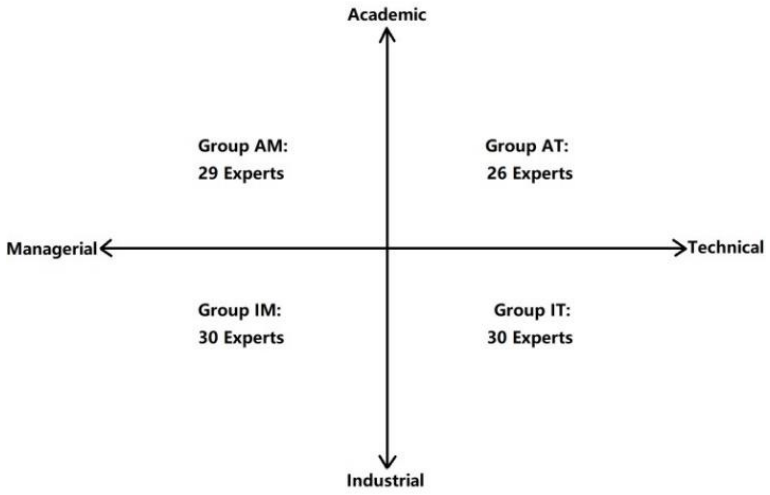
For the final academic expert selection, based on their research focuses and interests listed on Google Scholar, the research domains of selected scholars were classified into two subjects: Software Engineering and Security to match the research topic – DevSecOps. According to Jiang (2021), the cybersecurity domain encompasses security architecture, security engineering, network security, physical security, cloud security, software/application security, data protection, security governance, policies, and laws and regulations. In total, 55 academic experts were identified, with 31 specialising in Software Engineering, 17 in Security, and 7 in both fields.

For the final industrial expert selection, all 18 experts were included, as they were already scarce. To add more people, five editors of the book "Guide to the Software Engineering Body of Knowledge" (Washizaki, 2024), thirty presenters from GitHub Universe 2023, two New Zealand practitioners, and four colleagues of identified experts were identified as potential industrial participants. An Information System Skills Matrix presented by Monin and Dewe (1994) divides job roles into four types, i.e., specialists, generalists, technical roles, and managerial roles. The identified industrial experts encompassed a wide range of job roles, highlighting the comprehensiveness of this research, as detailed in Appendix B.6, "Final Potential Experts Selection" (Page 316).

Step (4): Experts Grouping:

To achieve a diverse range of voices and a broad base of knowledge in DevSecOps, a heterogeneous expert pool was established, comprising potential participants from both academic and industrial backgrounds, with technical or managerial expertise. A four-quadrant division was designed to classify the expert pool into four small groups, as shown in **Figure 4. 11**.

Figure 4. 11 - Four-quadrant division for expert grouping



X-axis denotes Technical and Managerial roles; Y-axis denotes Academic and Industrial roles. Technical and Managerial roles were adapted from the Information System Skills Matrix (Monin & Dewe, 1994), but without distinguishing between Specialist and Generalist roles. The current industry is shifting from a focus on "specialist to generalist" individuals to one that emphasises "T-skilled" individuals (Lal & Clear, 2021). Additionally, redundant groups will reduce the sample size within each group, thereby affecting data reliability. The academic experts were also classified into Technical and Managerial roles based on their research interests and topics stated in their biographies and publications.

As shown in **Figure 4. 11** and **Table 4. 6**, four quadrants denote four groups of potential experts: Group Academic Technical (AT) – 26 experts; Group Academic Managerial (AM) – 29 experts; Group Industrial Managerial (IM) – 30 experts; and Group Industrial Technical (IT) – 30 experts. **Table 4. 7** defines the aforementioned roles and groups of the identified potential experts.

Table 4. 6 - Potential expert groups

Groups	Experts pool
Group AT	Len Bass, Hausi A. Müller, Bram Adams, Marin Litoiu, Pooyan Jamshidi, Chris Parnin, Abdu Gumaei, Gabriel Tamura, André van Hoorn, Umberto Villano, Muhammad Azeem Akbar, Akond Rahman, Lotfi ben Othmane, Nicolas Ferry, Ali Ouni, Xin Peng, Eda Marchetti, Seiji Munetoh, Francis Bordeleau, Said Daoudagh, David Nandigam, Laurens Sion, Marios Fokaefs, Paolo Prinetto, Ian Welch, Mee Loong (Bobby) Yang
Group AM	Laurie Williams, Wouter Joosen, M. Ali Babar, Ricardo Colomo-Palacios, Stephen MacDonell, Daniela S. Cruzes, Norha M. Villegas, Riccardo Scandariato, Massimiliano Rak, Martin Gilje Jaatun, Sabrina Marczak, Julian Bass, Valentina Casola, Viktoria Stray, Alessandra De Benedictis, Jesus Luna, Alain April, Nobukazu Yoshioka, Maribel Sanchez-Segura, Haifeng Shen, Hasan Yasar, Mansooreh Zahedi, Erkuden Rios, Katja

	Tuma, John Noll, Tiziana Margaria, Sanjay Mathrani, Kelly Blincoe, Daniel Alencar da Costa
Group IM	Kirstie Magowan, Anastasios Arampatzis, Pete Cheslock, Gilad David Maayan, Bojana Dobran, Isaac Eldridge, Lucian Constantin, Christy Maerz, Ilai Bavati, Marius Rimkus, Ayush Singh, Mike Hanley, Robert Daugherty, Rob Bos, Caleb Queern, James Williams, Andrew Hoog, Asha Chakrabarty, Javier Cardoso, Erik Arcos, Justin Hutchings, Zain Malik, Courtney Claessens, Erin Havens, V. S. Mani, Baren Nel, Cheryll Singh, Philip Coster, Kinza Sarwar, Olivia Tang
Group IT	Sam Bocetta, James Darwin, Tj Blogumas, Mike Spisak, Kev Zettler, Ethan Miller, Mark Preston, Steve Tockey, Rich Hilliard, Steve Schwarm, Niroshan Rajadurai, René van Osnabrugge, Charlie Rice, Jérôme Djebari, Chad Bentz, Dan Shanahan, Abhishek Dutta, Mohammad Ismail, Igwe Kalu, Jeanyhwh Desulme, Nick Liffen, Mathew Payne, Shadi Samadi, Greg Mohler, Matthew Chieco, Charlton Trezevant, Joseph Katsioloudes, Tim Tegeler, Song Sun, Aziel Shaw

Table 4. 7 - Experts' roles and groups

Experts' Role	Experts' Group
Academic: Experts whose occupations are connected with education, especially studying in schools, colleges, universities, or other research institutions.	**Academic Technical:** Experts who work in academic institutions, such as professors and lecturers, and their research interests and publications connected with the practical use of machinery, methods, and technologies, e.g., programming, networking, clouds, cybersecurity practices, etc.
Industrial: Experts who work in the SE and cybersecurity industries.	**Academic Managerial:** Experts who work in academic institutions, such as professors and lecturers, and their research interests and publications connected with management, e.g., software process, software ecosystems, agile, CoSE, GSE, governance, policies, laws, and regulations, etc.
Managerial: Experts whose occupations relate to management or managers.	**Industrial Managerial:** Experts who work in SE and cybersecurity industries as managers or other positions relating to management, e.g., directors, general managers, product managers, CEO, CTO, etc.
Technical: Experts whose occupations are connected with the practical use of machinery, methods, and technologies.	**Industrial Technical:** Experts who work in the SE and cybersecurity industrial sector, and their work is connected with the practical use of machinery, methods, and technologies, e.g., engineers, architects, developers, testers, tech consultants, etc.

A few of the identified experts belonged to more than one expert group, such as serving as a university scholar while simultaneously working in industry. They were grouped based on their preference when they accepted the survey invitation. Experts in different groups were required to complete the same questionnaire to investigate whether different roles and perspectives affected the results.

After three rounds of expert selection and grouping, a list of potential participants was compiled

(55 academic experts and 60 industrial experts), including full names, emails/LinkedIn profiles, institutes/organisations, domains/roles, citations/work experiences, and groups. This list was discussed, revised, and confirmed with supervisors to start the recruitment process.

4.5.2.4 Expert Recruitment and Invitation

After finalising the list of potential experts, the researcher initiated the recruitment process by inviting academic and industrial experts to participate in the Delphi panel via email and LinkedIn messages.

A total of 115 invitations were sent to experts, and 32 responses were received, including both acceptances and refusals. The refusal rate was 72%. Among the 32 experts who replied, 18 accepted the invitation, 11 declined, and 3 experts withdrew after initially accepting.

Supporting Documents:

A set of supporting documents was developed and sent to the potential participants, including

- A Research Ethics Statement, which was approved by the Auckland University of Technology Ethics Committee (Appendix A on Page 294)
- An Information Sheet that provided a brief description of the Delphi study (Appendix B.2 on Page 306)
- A Consent Form that provided for participants' action (Appendix B.3 on Page 311).
- An Introductory Letter that provided a self-introduction and a brief description of the research to make initial contact with potential participants (Appendix B.4 on Page 312).
- An Explanation of Terms that defined all technical terms in the research (Appendix E.2 on Page 379).

Demographic Information:

In some Delphi studies, participants are required to answer demographic questions, such as gender, age, country, organisation, expertise, and work experience (Beiderbeck et al., 2021). This research omitted the step because demographic information was obtained from Google Scholar or LinkedIn during the initial and secondary selection stages.

Invitation via Email:

The academic experts were sent invitations via email with the associated documents. Their email addresses were provided on their publications and institution webpages. However, the email addresses of most industrial experts were not publicly available; therefore, LinkedIn was used to contact them.

Invitation via LinkedIn:

Kozłowski et al. (2021) proposed a virtual network sampling method using LinkedIn to obtain an appropriate number of respondents for global survey research, thereby enabling researchers to minimise both costs and risks. This approach leverages several advantages of LinkedIn network sampling, including the ability to target specific respondent groups, reduced recruitment costs, personalised outreach, and the absence of geographic limitations. LinkedIn has a network structure in which users are connected to specific groups and also have broader connections. Hence, it resembles the snowballing, enabling researchers to identify additional respondents by exploring the network of connections among people.

On the other hand, the LinkedIn network sampling method has two drawbacks: the exclusion of individuals in the study population who do not use LinkedIn or check it frequently, and a high refusal rate when initially contacted. In this study, the researcher was unable to establish sufficient connections and receive timely responses, primarily due to these two limitations. The result remained unsatisfactory, although the researcher had expanded their network through snowball sampling and upgraded to “LinkedIn Sales”, a subscription-based service that offered additional credit for InMail messages. Furthermore, according to LinkedIn’s policy, a user cannot send a follow-up or second InMail message to another user until the recipient has accepted and responded to the initial message.

Participants:

Delphi is an expert-based method; therefore, the number of participants may be much smaller than what is traditionally considered sufficient to ensure survey reliability (Loo, 2002). Ultimately, the number of participants does not guarantee reliability; however, the representativeness and expertise of participants do. In general, the sample size for a Delphi survey is mainly set between

15 and 30 (Lilja, Laakso, & Palomäki, 2011). It can range from 4 to hundreds, depending on the scope of the problem and the available resources (Powell, 2003).

The literature indicates that the number of iterations (real-time or multiple rounds) and the complexity of the questionnaire format are inversely proportional to the number of participants. For example, Warth, von der Gracht and Darkow (2013) conducted a real-time Delphi survey with 140 participants to assess controversial projections of electric drive vehicles for 2030; Ho et al. (2018) surveyed 25 participants in two Delphi rounds to rate the offshore wind farm (OWF) sitting criteria; Greene (2020) surveyed 18 participants in four rounds to explore various aspects of the perceived importance of developing a governance model for DevOps teams.

According to Md Arof (2015), when a Delphi survey is combined with the AHP method, the number of participants can be equal, slightly larger, or slightly smaller than when utilising Delphi individually. A key advantage of the combined Delphi-AHP method is its suitability for research topics with limited expert participation (Md Arof, 2015). For example, Kim, Jang and Lee (2013) surveyed 10 experts using Delphi-AHP method to select the priorities of waste electrical and electronic equipment; Di Zio and Maretti (2013) surveyed 12 experts to assess the acceptability of energy sources; Bouzon et al. (2016) surveyed 16 experts to identify and analyse reverse logistics barriers; Kim et al. (2022) surveyed 21 experts to develop a probabilistic tunnel collapse risk evaluation model by using Delphi-AHP method; Zhao, Md Ali and Ahmad (2023) presented Delphi-AHP method to develop the indicators for sustainable urban regeneration in historic urban areas, the number of participants was 25 in the first round and decreased to 20 and 16 in the second and third round, respectively.

Based on the above examples, 16-20 were justified as an acceptable sample size for this research, and 4-5 participants per group were considered sufficient to ensure representativeness. In fact, experience from this research indicates that AHP is more suitable for small to medium-sized groups (e.g., 8-20 participants) than for tiny groups (fewer than 5 participants) or relatively large groups (over 30 participants). Using the AHP method with very few participants may lead to high inconsistencies in comparisons. By contrast, using AHP with large groups may yield overly aggregated comparison outcomes (i.e., converging to the “Equally important” scale). In which case, the AHP method will lose its merit.

Before conducting the first round of the Delphi survey, 18 participants accepted invitations; all met the criteria for expert selection. The number of participants varied across rounds, as participants could withdraw at any time before the survey's completion. **Table 4. 8** provides limited information about the 18 participants, as confidentiality agreements have been signed and the Delphi survey is anonymous.

Table 4. 8 - List of participants

Group	Part. ID	Occupation / Role	Academic	Industry			Country
				Portfolio	Program	Team	
AT	Pa1	Researcher	1				Belgium
	Pa2	Assistant Professor	1				Canada
	Pa3	Senior Lecturer	1				NZ
	Pa4	Associate Professor	1				USA
AM	Pa5	Professor	1				Ireland
	Pa6	Associate Professor	1				Ireland
	Pa7	Professor / Director	1	1	1		UK
	Pa8	Researcher / Technical Manager / Senior Security Architect	1	1	1		Germany
	Pa9	Researcher/Director	1	1	1		USA
IT	Pa10	Principal Consultant			1	1	USA
	Pa11	Principal Software Engineer		1	1	1	Germany
	Pa12	Senior DevOps Engineer			1	1	China
	Pa13	DevOps Engineer			1	1	USA
IM	Pa14	Head Marketing and Communications		1	1		India
	Pa15	General Manager (Operations side)		1	1	1	NZ
	Pa16	Senior Security Consultant		1	1		NZ
	Pa17	Senior Agile Coach		1	1		NZ, China
	Pa18	Consultant (Adviser) / Researcher	1	1	1		NZ

To ensure a sample that was representative of the nature of the DevSecOps ecosystem and the reliability of the findings, participants were recruited worldwide, with a distribution across New Zealand, Europe, Asia, and North America. Participants hold various roles and work in different domains, including academic, industrial, managerial, and technical. Participants Pa1 – Pa9 are academic experts; Pa10 – Pa18 are industrial experts; Pa5 – Pa9 and Pa14 – Pa18 are managerial experts; Pa1 – Pa4 and Pa10 – Pa 13 are technical experts.

According to **Figure 4. 11** (Page 93), participants were divided into four groups: Group Academic Managerial (AM), Group Industrial Managerial (IM), Group Academic Technical (AT), and

Group Industrial Technical (IT). All groups would answer the same questionnaire.

Industrial participants were from large-scale or globally distributed software development organisations and were further differentiated by three organisational levels: Portfolio, Program, and Team, from high to low (Beecham et al., 2021). The portfolio level is the highest level of an organisation and is responsible for managing the organisation's strategic objectives, programs, and projects. The program level, which is below the portfolio level, contains many teams that deliver solutions. The team level is foundational and responsible for providing value to customers (Beecham et al., 2021). By doing so, it could be investigated whether industrial participants hold dissenting opinions due to differences in organisational levels and perspectives.

4.5.2.5 Survey Conduct

The first round of the Delphi survey was conducted once the number of participants reached the minimum requirement of 16.

Research Ethics:

A research ethical application has been approved by the Auckland University of Technology Ethics Committee as "23/225 - Representing global DevSecOps to usefully support software engineering practice", which is declared when inviting and recruiting experts (see Appendix A on Page 294).

Delphi Survey Rounds:

Delphi was an iterative process comprising three survey rounds, as shown in **Table 4. 9**. Each round consisted of 20-30 questions and lasted 20-30 minutes. Round One was conducted to rate the importance of the identified DevSecOps challenges using an AHP format (pairwise comparisons). Based on Round One results, the list of DevSecOps challenges was revised. Round Two was conducted to assess the importance of the revised challenges and the identified DevSecOps practices. Based on Round Two results, the list of practices was revised. Round Three was conducted to evaluate the importance of the revised practices and the identified DevSecOps metrics. Based on the results of three Delphi survey iterations, the DevSecOps CPTM Model (Version 1.0) was revised. Since the second survey round, a minimum number of AHP comparisons has been used instead of completing all pairwise comparisons. By doing so,

redundant questions were removed to reduce unnecessary workload and survey durations for participants, and the inconsistencies among redundant pairwise comparisons were avoided (Mu & Pereyra-Rojas, 2017).

Table 4. 9 - Three rounds of Delphi survey

Round of survey	Estimated duration	Goal and activities
Round One	20 minutes	Rate the importance of identified DevSecOps challenges
Round Two	20 minutes	Reassess the revised challenges and rate the importance of identified DevSecOps practices (minimum number of comparisons)
Round Three	30 minutes	Reassess the revised practices and rate the importance of identified DevSecOps metrics (minimum number of comparisons)

The importance of the identified DevSecOps tools was not assessed through a survey round, because the tools' themes overlapped with some practices in the "Technology" category. Second, making pairwise comparisons is the fundamental principle of the AHP method; however, tools of different types are incommensurable. For instance, a security testing tool and a communication tool are not comparable in perceived importance, as they serve different purposes. Third, it is unrealistic to assume that participants have experience with all tools across the DevSecOps lifecycle, even though they are DevSecOps and DevOps experts. Hence, the use of a survey to evaluate tools was abandoned.

Stopping Criteria:

According to Beiderbeck et al. (2021), stopping criteria should be defined in terms of time, participants, and consensus. The three dimensions (time, participants, and consensus) are interdependent and constrain one another. Enlisting participants and reaching consensus takes time. Conversely, the time constraint limits the number of participants and survey rounds, which may further affect consensus.

At the beginning of the study, a timeline was developed, including the estimated duration for each phase and step of the Delphi process (see Appendix B.7, "Delphi Plan" on Page 324). The researcher continually updated the schedule as the research progressed and wrote progress reports to assess progress and refine the plan. Participant recruitment proved to be the most time-consuming and challenging step, and it was underestimated. In this case, the first survey round began with the confirmed participants while simultaneously inviting additional experts to

participate. Earlier in the Delphi study, the participants dimension had higher priority than the time dimension. That is, it was acceptable to spend time on recruiting high-quality participants and giving them sufficient time to consider the invitation and respond to the survey. In the middle and later stages, time requirements were prioritised over participant needs, allowing the research to be completed on schedule. It is unrealistic to wait indefinitely for participants' responses and to expect to achieve perfect consensus. Hence, the period for one survey iteration was approximately three months, including data collection, processing, analysis, and reporting. It may vary depending on the specific context, for example, on common and national holidays worldwide.

Although the majority of studies use Delphi to build consensus, and consensus plays an important role in Delphi studies, it is not Delphi's aim (Von der Gracht, 2012). Dissent is a valid outcome and could even be the intended outcome of using this method. For example, Warth, von der Gracht and Darkow (2013) conducted a dissent-based Delphi study on multi-stakeholder scenario development for the future of electric drive vehicles, and the results revealed a considerably high degree of dissent. Ho et al. (2018) conducted a Delphi survey to prioritise offshore wind farm (OWF) sitting criteria, using response stability as the stopping criterion rather than consensus. They focused on understanding dissent through a comprehensive discussion.

Reaching consensus was not considered a mandatory stopping criterion for this Delphi study, as neither consensus nor its absence was required or guaranteed. The subjectivity and dissent arising from different participants' roles, expertise, and experiences were regarded as a strength in knowledge production rather than a problem. Nonetheless, this research calculated the Coefficient of variation (CV) to evaluate and analyse dissent. According to Dajani, Sincoff and Talley (1979), the survey could be terminated when the CV was less than 0.5, whereas Rho (2006) expanded the acceptable range to be less than 0.8. Later, Section 4.5.3.3 discusses the CV in detail.

As the AHP approach was employed in this research, the consistency ratio (CR) was used to evaluate consistency. According to Saaty (1982), the "reasonable" value of CR is lower than 0.2, and the "acceptable" value is lower than 0.1. Hence, achieving acceptable CR values across all comparison matrices, i.e., below 0.1, was considered a key stopping criterion for the Delphi study. After completing a survey round, data analysis and consistency analysis were conducted together, and the results were shared with participants to help them complete the following rounds more

accurately.

Similar to the study by Ho et al. (2018), this research treated response stability as another stopping criterion. As noted above, two Delphi iterations were conducted to assess each element of the DevSecOps CPTM Model (i.e., Challenges, Practices, and Metrics). When two iterations produced relatively stable ratings and rankings for the assessed elements, it was time to stop assessing them and move on to the next set. Dialectically speaking, the survey round could also be terminated if the results reached consensus rather than stability, indicating that the majority of participants overturned the previous round's results. In other words, reaching stability and reaching consensus were two optional stopping criteria. Achieving the acceptable consistency ratio (CR) was mandatory.

4.5.2.6 Expert Follow-Up

Expert follow-up was an indispensable step in the Delphi study. The results of each completed round were sent to participants to help them prepare for the next round, and an overview of the final research findings was shared to enrich the discussion. Participants would be officially acknowledged for their contributions to any research outputs if they agreed to be acknowledged during the participant invitation process. Throughout the Delphi process, reminder or prompting messages were sent to participants to track progress.

It was inevitable that a few participants withdrew during the conduct of survey rounds, as the Information Sheet declared: "Kindly rest assured that your participation is voluntary, and you may withdraw at any time before the completion of the Delphi survey". This challenge was addressed by inviting additional participants. It should be noted that newly joined participants were required to complete all previous survey rounds. For instance, a participant who newly joined during Round Two had to complete Round One before proceeding to Round Two. This is due to the Delphi method's iterative feedback process. Thus, as the Delphi survey progressed, a slight decrease in participant numbers was perfectly acceptable rather than an increase. Otherwise, the consensus reached in earlier rounds could be impacted.

4.5.3 Analysis Phase

The goals of the Analysis phase were to:

- Analyse AHP results.
- Analyse answers to open-ended questions.
- Analyse dissents.

4.5.3.1 AHP Results Analysis

Calculating Mean and Standard Deviation:

In each survey round, a frequency analysis was conducted once data collection was complete. This was done by using the Qualtrics (2024) survey platform. The arithmetic mean and standard deviation were mainly used (Beiderbeck et al., 2021).

Mean, median, and mode are three measures of central tendency (Von der Gracht, 2012). For relatively complex problems, the mean and median are more suitable choices than the mode, as the mode is not appropriate for scales with many values.

Standard deviation is a measure of dispersion for the mean, and the Interquartile range (IQR) is a measure of dispersion for the median. In Delphi studies, both sets of measures are commonly used, though it is argued that median and IQR are more robust than mean and standard deviation because they exclude data outliers (Ho et al., 2018; Von der Gracht, 2012).

Nonetheless, mean and standard deviation were used as measures in this study, because extreme values are common in AHP comparisons, which should not be eliminated as outliers. Without extreme values, the aggregation of participants' judgments would be taken arbitrarily close to very moderate outcomes, as it converges to the "Equally important" scale. In which case, ranking gaps would get weakened, and the AHP method would lose its merit.

Conversion between Qualtrics and SuperDecisions:

As mentioned previously in Section 4.5.2.2, the AHP tool SuperDecisions (SuperDecisions, 2023) was used to generate the pairwise comparison questionnaire, which was then transferred to the Qualtrics platform (Qualtrics, 2024) for survey programming. In this step, the process was reversed, i.e., Qualtrics was used to collect data (responses), and the scores (relative priorities)

were imported into SuperDecisions for AHP pairwise comparison analysis.

It is worth noting that the Qualtrics score should be converted to the SuperDecisions score, according to **Table 4. 10**, since the Qualtrics platform does not support the AHP format. (AHP Scale: 1 - Equally important; 3 - Moderately more important; 5 - Strongly more important; 7 - Very strongly more important; 9 - Extremely more important; 2, 4, 6, 8 - Intermediate values; fractions - reciprocal comparisons).

Table 4. 10 - Conversion between Qualtrics score, SuperDecisions score, and AHP score

Qualtrics		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	
SuperDecisions	**C1**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**C2**
AHP	**C1**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**1/2**	**1/3**	**1/4**	**1/5**	**1/6**	**1/7**	**1/8**	**1/9**	**C2**

SuperDecisions generated the AHP results of each pairwise comparison, including the normalised value, idealised value, and consistency ratio. This tool featured five modes of result reporting: Matrix, Graphical, Verbal, Questionnaire, and Direct (SuperDecisions, 2023).

Aggregation of Judgments:

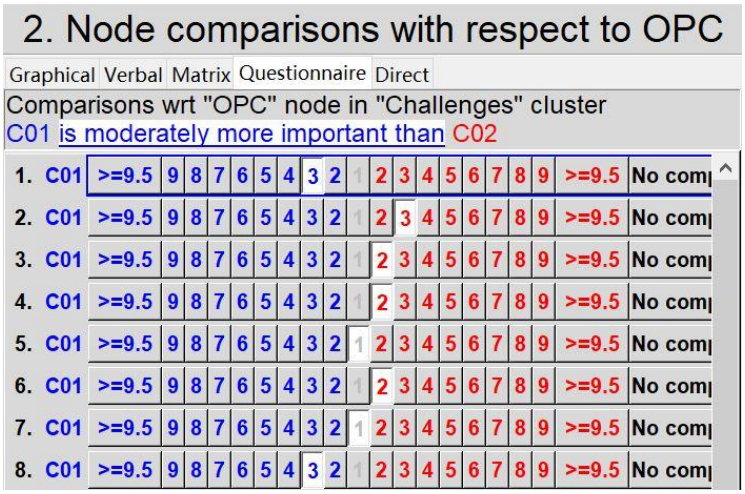
Multiple participants were involved in the Delphi survey. When performing AHP pairwise comparisons, participants would have different opinions. Hence, the aggregation of participants' judgments was obtained by calculating the mean. In general, the arithmetic mean is suitable for integers, whereas the geometric mean fits fractions (i.e., reciprocal values). AHP pairwise comparisons always yield a dataset containing both integers and fractions, i.e., conflicting ratings. Hence, both arithmetic and geometric means can be selected; the latter is more commonly used, for instance, in (Akbar et al., 2020; Akbar, Rafi, et al., 2022; Zohaib, Alsanad, & Abdullah Alhogail, 2024).

In this research, as presented previously in **Table 4. 10**, the arithmetic means of Qualtrics scores were directly calculated, ranging from 1 to 17 without fractions, rather than calculating the geometric means of AHP scores after two conversions. Afterwards, the arithmetic means of Qualtrics scores (ranging from 1 to 17) were converted to SuperDecisions scores (red and blue scores ranging from 1 to 9). SuperDecisions scores were then imported to the SuperDecisions tool for further calculations. This was the easiest and most accurate method, as it involved no fractional

or irrational numbers during the initial processing of the raw data.

In fact, as shown in **Figure 4. 12**, AHP fractional scores were not actually used when performing calculations with the tool combination of Qualtrics and SuperDecisions (using its Questionnaire mode).

Figure 4. 12 - SuperDecisions node comparisons in Questionnaire mode



Consistency Assessment:

The consistency ratio (CR) of each comparison matrix was also calculated using the AHP tool SuperDecisions. This tool uses the term “Inconsistency Ratio” instead of “Consistency Ratio”. The consistency could be affected by many factors, and the most important one was participants’ AHP knowledge. If a participant responded to the questionnaire based on the AHP principle consciously, inconsistency could be controlled.

The acceptable value of CR was defined between 0 and 0.1. This study combined the AHP method with multiple Delphi survey rounds, enabling participants and the researcher to revise and repeat the pairwise comparisons if an unacceptable CR occurred. After completing a survey round, data analysis and consistency analysis were conducted, and results were shared with participants to help them complete the next round.

In this research, achieving an acceptable CR value for all comparison matrices was considered one of the stopping criteria for the Delphi survey; the AHP consistency would therefore be iteratively improved if it was not within an acceptable range.

4.5.3.2 Answers to Open-ended Questions

At the end of each questionnaire, one or two open-ended questions were raised to collect participants' additional comments on the existing findings and their opinions on the differences between local and global DevSecOps. Similar to the data synthesis step in the previous MLR study, the Thematic Analysis (TA) method was adopted to code and theme the collected qualitative data.

In the previous MLR study, TA was initially conducted in an inductive approach (coding and theming were directed by the content of data), and then switched to a deductive approach (coding and theming were directed by existing concepts) (Braun & Clarke, 2020).

With the Delphi study, the TA process was performed in reverse. It initially followed a deductive approach, attempting to map the newly collected data onto the DevSecOps CPTM Model (Version 1.0), where there were overlaps between codes and themes. Subsequently, coding and theming were directed by the content of new data in an inductive approach (Braun & Clarke, 2021). In the next Delphi iteration, these emerging themes were included for comparison with the original themes to add or replace the least important existing ones. In this way, the DevSecOps CPTM Model was refined and improved iteratively.

4.5.3.3 Dissent Analysis

A dissent analysis was conducted to examine participants' agreement and disagreement with AHP comparison results and global DevSecOps opinions, using both quantitative and qualitative methods.

For AHP comparison results, dissents were analysed at three levels of the participants:

- Individual level: to analyse the consensus or dissent among each participant regardless of participant's position and role.
- Groups level: to analyse the consensus or dissent between four participants' groups (i.e., Group AT, Group AM, Group IT, and Group IM).
- Roles level: to analyse the consensus or dissent between four participants' roles (i.e., Academic, Industrial, Managerial, and Technical).

Two datasets for Groups AM and AT were merged into the Academic role; Groups IM and IT were merged into the Industrial role; Groups AT and IT were merged into the Technical role; and Groups AM and IM were merged into the Managerial role. Each group consisted of four to five participants, and each role covered two groups involving eight to ten participants, as there were overlaps between them.

In this research, the Coefficient of variation (CV) was calculated to evaluate the degree of dissent among participants based on participants' individual level. CV is defined as the ratio of the standard deviation to the arithmetic mean, and a higher value indicates greater dissimilarity among participants (Everitt, 2006). This research used CV rather than standard deviation because it is a dimensionless number, meaning it is independent of the unit in which the measurement was taken. In contrast, the standard deviation depends on its corresponding arithmetic mean, so it should not be used for comparing datasets with different units (Everitt, 2006).

In a combined AHP-Delphi study presented by Kim, Jang and Lee (2013), CV was analysed for dissent evaluation, and the acceptable value was set at 0.5 or lower. According to Dajani, Sincoff and Talley (1979), the survey could be terminated when the CV value was less than 0.5. In contrast, in another combined AHP-Delphi study by Kim et al. (2022), the acceptable CV value was defined as less than 0.8, according to Rho (2006). As discussed above in Section 4.5.2.5, although reaching consensus was not a stopping criterion for the Delphi study, both acceptable CV ranges (0-0.5 and 0-0.8) were used for the dissent analysis.

Given the AHP method's key feature of compromise and consensus building for groups (Packeer Mohamed et al., 2022), analysing dissent by participants' roles and groups was more reasonable than analysing it at the individual level. This was also more satisfactory for the content of *Sub-question 3.2: "Do experts have dissents on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*.

Unlike the results of AHP pairwise comparisons, which were complicated and extremely large in quantity, the dissent analysis for global DevSecOps opinions was more straightforward because it used multiple-choice questions. It was more intuitive to identify the proportions of participants across different options and to conduct dissent analysis by the individual, group, and role levels.

4.6 Chapter Summary

In summary, this chapter explains the research design in terms of the research paradigm, methodology, and methods; describes the overall research process of this study in detail; and discusses the validity and appropriateness of the combination of the Delphi survey and the AHP method. Next, Chapter 5 presents the results of the Round One Delphi survey. In Round One, the participants assessed the importance of 28 identified DevSecOps challenges.

5 Chapter 5: Round One Results – DevSecOps Challenges

This chapter presents the results of the first round of the Delphi survey, conducted between May and October 2024, involving 18 participants. In Round One, the importance of 28 identified DevSecOps challenges was compared using the AHP method (pairwise comparisons) across the four categories: Business, Organisation, People & Culture (OPC), Process Capabilities (PC), and Technology.

The priorities and rankings of these challenges and categories were calculated and reported. In the meantime, consistency ratios were calculated, and all were within an acceptable range. In addition to the AHP pairwise comparisons, seven new DevSecOps challenges were identified from participants' comments and incorporated to improve the DevSecOps CPTM Model (Version 1.0). Furthermore, participants' opinions on the differences between local and global DevSecOps challenges were collected and reported. Finally, a dissent analysis was conducted to examine participants' agreement and disagreement with the AHP comparison results and global DevSecOps opinions, by using both quantitative and qualitative methods.

In Section 5.1, the context and background for conducting the Round One survey are provided, including the survey objectives, questionnaire content, and participants' information. AHP comparison results are reported and analysed in Section 5.2, alongside seven additional DevSecOps challenges identified from participants' comments. In Section 5.3, participants' opinions on the difference between local and global DevSecOps challenges are reported and analysed. Section 5.4 provides information on the dissent analysis for the Round One survey. Section 5.5 concludes the Round One survey and provides information on the Round Two Delphi survey.

5.1 Context and Background of Round One

This section provides the context and background for the first round of the Delphi survey, including its objectives, questionnaire content, and participants' information.

5.1.1 Survey Objectives in Round One

The Delphi survey was an iterative process comprising three rounds. The first round of the Delphi survey was conducted for multiple objectives. The primary purpose was to rate and rank the importance of 28 identified DevSecOps challenges within their categories, by using a set of AHP-based closed-ended questions (i.e., pairwise comparisons), to answer *RQ3: "How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?"*

Some additional DevSecOps challenges were expected to be collected from participants, to answer *Sub-question 3.1: "What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?"* In the following Round Two, these new challenges would be compared with the initially identified ones. Based on the Round Two results, some of the lowest-ranked original challenges might be replaced to refine and improve the DevSecOps CPTM Model (Version 1.0).

Next, to address *Sub-question 3.2: "Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical and managerial)?"*, the participants were divided in four small groups, namely, Group Academic Managerial (AM), Group Industrial Managerial (IM), Group Academic Technical (AT), and Group Industrial Technical (IT). All these groups answered the same questionnaire, enabling an investigation into whether participants held dissenting opinions due to differences in role and organisational level. A dissent analysis was conducted accordingly for this survey round.

Finally, participants' opinions on the difference between local and global DevSecOps challenges were collected, to address *RQ4: "What are the experts' opinions on DevSecOps in GSE contexts?"* and its two related sub-questions: *"Sub-question 4.1: How is DevSecOps different between local and global settings?"* and *"Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?"*

5.1.2 Questionnaire Contents in Round One

To achieve these objectives and address corresponding research questions, the Round One survey questionnaire consisted of 20 questions, employing multiple formats, including AHP pairwise comparisons (Questions 1 – 17), multiple-choice (Question 19), and open-ended questions (Questions 18 and 20). **Table 5. 1** lists the 20 questions, including their objectives, format types,

and descriptions. A sample of the Round One survey is provided in Appendix B.8, "Sample of Delphi Survey – Round One" (Page 326), or it can be assessed at the Qualtrics platform (https://aut.au1.qualtrics.com/jfe/form/SV_aXjvAyldHeQpUmG).

Table 5. 1 - List of questions in Round One

Question ID	Objective	Format	Description
Q1	To rate the importance of the four categories of DevSecOps challenges	6 AHP pairwise comparisons	Business : OPC Business : PC Business : Technology OPC : PC OPC : Technology PC : Technology
Q2	To rate the importance of nine DevSecOps challenges in the OPC (Organisation, People & Culture) category	36 AHP pairwise comparisons	C01 : C02 – C09
Q3			C02 : C03 – C09
Q4			C03 : C04 – C09
Q5			C04 : C05 – C09
Q6			C05 : C06 – C09
Q7			C06 : C07 – C09 C07 : C08 – C09 C08 : C09
Q8	To rate the importance of eight DevSecOps challenges in the PC (Process Capabilities) category	28 AHP pairwise comparisons	C10 : C11 – C17
Q9			C11 : C12 – C17
Q10			C12 : C13 – C7
Q11			C13 : C14 – C17
Q12			C14 : C15 – C17 C15 : C16 – C17 C16 : C17
Q13	To rate the importance of the seven DevSecOps challenges in the Technology category	21 AHP pairwise comparisons	C18 : C19 – C24
Q14			C19 : C20 – C24
Q15			C20 : C21 – C24
Q16			C21 : C22 – C24 C22 : C23 – C24 C23 : C24
Q17	To rate the importance of the four DevSecOps challenges in the Business category	6 AHP pairwise comparisons	C25 : C26 – C28 C26 : C27 – C28 C27 : C28
Q18	To collect additional DevSecOps challenges	1 open-ended question	Add more challenges
Q19	To investigate how DevSecOps challenges differ in local and global settings	1 multiple-choice question	4 options: Extremely different; Slightly different; Not different; Uncertain
Q20	To collect differences between local and global DevSecOps challenges	1 open-ended question	List the differences if opted for "Extremely/Slightly Different" in Q19

5.1.3 Participants in Round One

In Round One, 18 participants were surveyed. The number of participants met the minimum requirement of 16, as set in Section 4.5.2.4, in Chapter 4 (Page 95). While the Delphi survey ought to be anonymous, **Table 5. 2** provides brief information about the participants in the Round One survey. These 18 were the initial participants, as listed in **Table 4. 8**, in Section 4.5.2.4, Chapter 4 (Page 98). Participants could vary across rounds because the Delphi survey was voluntary, and some might withdraw at any time before completing the entire survey

Table 5. 2 - List of participants in Round One

Group	Part. ID	Occupation / Role	Academic	Industry			Country
				Portfolio	Program	Team	
AT	Pa1	Researcher	1				Belgium
	Pa2	Assistant Professor	1				Canada
	Pa3	Senior Lecturer	1				NZ
	Pa4	Associate Professor	1				USA
AM	Pa5	Professor	1				Ireland
	Pa6	Associate Professor	1				Ireland
	Pa7	Professor / Director	1	1	1		UK
	Pa8	Researcher / Technical Manager / Senior Security Architect	1	1	1		Germany
	Pa9	Researcher / Director	1	1	1		USA
IT	Pa10	Principal Consultant			1	1	USA
	Pa11	Principal Software Engineer		1	1	1	Germany
	Pa12	Senior DevOps Engineer			1	1	China
	Pa13	DevOps Engineer			1	1	USA
IM	Pa14	Head Marketing and Communications		1	1		India
	Pa15	General Manager (Operations side)		1	1	1	NZ
	Pa16	Senior Security Consultant		1	1		NZ
	Pa17	Senior Agile Coach		1	1		NZ, China
	Pa18	Consultant (Adviser) / Researcher	1	1	1		NZ

To ensure the representativeness of participants and the reliability of the survey, participants were recruited from around the world and distributed across New Zealand, Europe, Asia, and North America. These participants have a variety of roles and work in different fields. They could be academic, industrial, managerial, technical, or have multiple identities. As shown in **Table 5. 2**:

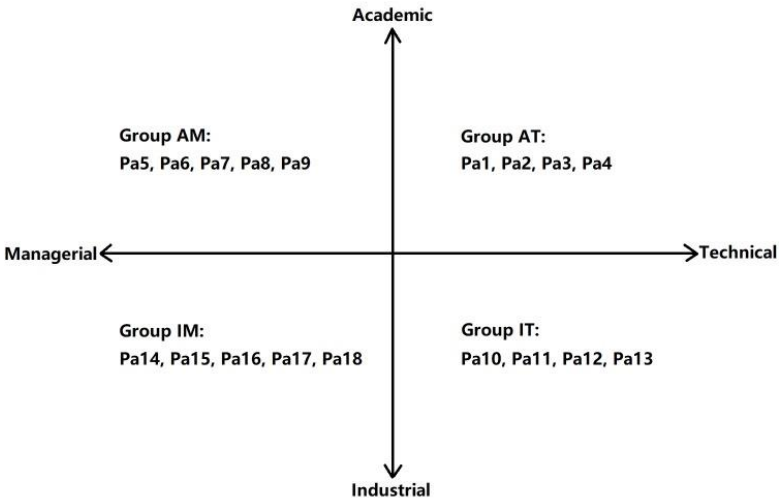
- Participants Pa1 – Pa9 are academic experts.

- Pa10 – Pa18 are industrial experts.
- Pa5 – Pa9 and Pa14 – Pa18 are managerial experts.
- and Pa1 – Pa4 and Pa10 – Pa13 are technical experts.

According to the four-quadrant division for expert grouping presented in Section 4.5.2.3, Chapter 4 (Page 89), the participants were divided into four groups, as shown in **Figure 5. 1**:

- Participants Pa1, Pa2, Pa3, and Pa4 are placed in the Group Academic Technical (AT).
- Pa5, Pa6, Pa7, Pa8, and Pa9 are in the Group Academic Managerial (AM);
- Pa10, Pa11, Pa12, and Pa13 are in the Group Industrial Technical (IT);
- Pa14, Pa15, Pa16, Pa17, and Pa18 are in the Group Industrial Managerial (IM).

Figure 5. 1 - Four-quadrant division for participants grouping in Round One



Regardless of the group placement, all participants were to answer the same questionnaire. Industrial participants were from large-scale or globally distributed organisations, and were further differentiated into three organisational levels: Portfolio level, Program level, and Team level (Beecham et al., 2021). By doing so, it enabled an investigation into whether participants held dissenting opinions due to differences in roles and organisational levels.

5.2 RQ3 – Evaluation of DevSecOps Challenges

This section presents the results of the AHP pairwise comparisons (Questions 1 – 17) and one open-ended question (Question 18). The data were collected, calculated, analysed, discussed, and reported using multiple tools (e.g., Qualtrics, SuperDecisions, and MS Excel spreadsheets) to partially address RQ3 and its associated sub-questions, with a focus on the DevSecOps challenges dimension.

5.2.1 AHP Results of DevSecOps Challenges

In Round One, as shown in **Table 5. 1**, there were a total of 97 AHP pairwise comparisons that were investigated via 17 survey questions and contained in five comparison matrices. Hence, six comparisons between four categories of challenges; 36 comparisons between nine challenges in the "Organisation, People and Culture (OPC)" category; 28 comparisons between eight challenges in the "Process Capabilities (PC)" category; 21 comparisons between seven challenges in the "Technology" category; and six comparisons between four challenges in the "Business" category.

The survey responses were collected using the Qualtrics online survey platform (Qualtrics, 2024). The initial response sheets contain participant information, such as IP addresses; therefore, they cannot be disclosed due to confidentiality constraints. The processed dataset of responses is available at zenodo.org (https://doi.org/10.5281/zenodo.16932278).

There were 18 participants involved in Round One. Because the key feature of the AHP method is building compromise and consensus among participants (Packeer Mohamed et al., 2022), data should be analysed at the group level rather than the individual level.

Thus, the extracted data were further transformed into mean values to evaluate the AHP pairwise comparisons and obtain a set of group judgements. The Consistency Ratio (CR) was calculated to assess the consistency of the AHP comparisons. The Coefficient of Variation (CV) was calculated to assess the degree of consensus or dissent among participants. Section 4.5.3 in Chapter 4 (Page 103) has introduced the data analysis process in detail.

AHP comparison matrices:

Table 5. 3 presents a standard 9-point AHP Scale, which was used to quantify priorities (Saaty, 2013).

Table 5. 3 - 9-point AHP comparison scale

Scale	Numerical score	Reciprocal
Equally important	1	1
Moderately more important	3	1/3
Strongly more important	5	1/5
Very strongly more important	7	1/7
Extremely more important	9	1/9
Intermediate values	2, 4, 6, 8	1/2, 1/4, 1/6, 1/8

Table 5. 4 reports the AHP comparison matrix for the four categories: Organisation, People and Culture (OPC), Process Capabilities (PC), Technology, and Business.

Table 5. 4 - AHP comparison matrix for categories of challenges

Categories of Challenges	Business	OPC	PC	Technology
Business	1	1/3	1	2
OPC	3	1	4	5
PC	1	1/4	1	4
Technology	1/2	1/5	1/4	1

Table 5. 5, **Table 5. 6**, **Table 5. 7**, and **Table 5. 8** provide four AHP comparison matrices for the identified DevSecOps challenges within each category. These matrices listed the relative priorities of all criteria and sub-criteria, and fractions were the relative priorities of reciprocal comparisons, according to **Equation (3)**: $\mathbf{A} = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ \frac{1}{a_{12}} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{1n}} & \frac{1}{a_{2n}} & \cdots & 1 \end{pmatrix}$ (Page 73), which has been presented in Section 4.3.2.2 in Chapter 4 (Page 72). For example, Business : OPC = 1/3, when OPC : Business = 3.

Table 5. 5 - AHP comparison matrix for challenges in "Organisation, People & Culture" category

Challenges in Organisation, People & Culture	C01	C02	C03	C04	C05	C06	C07	C08	C09
C01	1	3	1/3	1/2	1/2	1	1/2	1	3
C02	1/3	1	1/2	1/3	1/2	1/2	1	1/2	4
C03	3	2	1	1	2	1	2	2	5
C04	2	3	1	1	3	3	4	2	6
C05	2	2	1/2	1/3	1	1	1/2	2	5
C06	1	2	1	1/3	1	1	2	1	4
C07	2	1	1/2	1/4	2	1/2	1	1	4
C08	1	2	1/2	1/2	1/2	1	1	1	4
C09	1/3	1/4	1/5	1/6	1/5	1/4	1/4	1/4	1

Table 5. 6 - AHP comparison matrix for challenges in "Process Capabilities" category

Challenges in Process Capabilities	C10	C11	C12	C13	C14	C15	C16	C17
C10	1	2	1/3	1/3	1	1/3	1	1/4
C11	1/2	1	1/3	1/4	1/3	1/4	1	1/5
C12	3	3	1	3	3	1/2	2	1/3
C13	3	4	1/3	1	2	1/2	2	1/4
C14	1	3	1/3	1/2	1	1/2	2	1/3
C15	3	4	2	2	2	1	3	1/2
C16	1	1	2	1/2	1/2	1/3	1	1/3
C17	4	5	3	4	3	2	3	1

Table 5. 7 - AHP comparison matrix for challenges in "Technology" category

Challenges in Technology	C18	C19	C20	C21	C22	C23	C24
C18	1	1/2	1/3	1	2	1/2	1
C19	2	1	1/4	2	2	1/2	1
C20	3	4	1	2	3	1/2	2
C21	1	1/2	1/2	1	2	1/2	1/2
C22	1/2	1/2	1/3	1/2	1	1/2	1/3
C23	2	2	2	2	2	1	1
C24	1	1	1/2	2	3	1	1

Table 5. 8 - AHP comparison matrix for challenges in "Business" category

Challenges in Business	C25	C26	C27	C28
C25	1	1/3	1/3	1/2
C26	3	1	2	3
C27	3	1/2	1	2
C28	2	1/3	1/2	1

Priorities, rankings, and consistency ratios:

97 AHP pairwise comparisons required a massive number of computations that could not be manually done; hence, the AHP tool SuperDecisions (SuperDecisions, 2023) was used to derive the priorities/weights and the consistency ratios. The calculation method and equations have been presented in Sections 4.3.2.2 (Page 72) and 4.3.2.3 (Page 74) in Chapter 4.

According to **Equation (5)**: $\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^{T}\mathbf{1} = 1 \end{cases}$ (Page 73), the priority/weight vector **w** was derived from any pairwise comparison matrix **A**. According to **Equation (6)**: $CR = \frac{CI}{RI}$ (Page 74), the value of the consistency ratio (CR) was obtained by comparing the consistency index (CI) of the matrix versus the consistency index of a random-like matrix (RI). The value of CI was obtained by using **Equation (7)**: $CI = \frac{\lambda_{max}-n}{n-1}$ (Page 74). The value of RI depends on the matrix size. Saaty (2013) calculated the RI values for the matrices of different sizes in **Table 5. 9**. In this case, the sizes of the five matrices were 4, 9, 8, 7, and 4 (previously reported in **Table 5. 4**, **Table 5. 5**, **Table 5. 6**, **Table 5. 7**, and **Table 5. 8**). Hence, their corresponding RI values were 0.89, 1.45, 1.40, 1.35, and 0.89.

Table 5. 9 - RI values for the matrices of different sizes

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49	1.52	1.54	1.56	1.58	1.59

Table 5. 10 presents the AHP results for four categories, **Table 5. 11**, **Table 5. 12**, **Table 5. 13**, and **Table 5. 14** report the AHP results for the identified DevSecOps challenges within each category. In these tables:

- "Normalised Value" represents the priority/weight within its own matrix/category (SuperDecisions, 2023). It may easily be confused with another AHP terminology, "Local priority/weight", which is explicitly used for alternatives, rather than criteria and sub-criteria (Mu & Pereyra-Rojas, 2017). Here, categories are criteria, and challenges are sub-criteria.
- "Idealised Value" represents the relative proportion compared with the highest-priority object, which has an idealised value of "1" by default (SuperDecisions, 2023).

- "Ranking" refers to the ranking of each object within its own matrix/category, ordered by normalised/idealised values from high to low (SuperDecisions, 2023).
- "Consistency Ratio" represents the consistency among all pairwise comparisons in a matrix; its acceptable value should be between 0 and 0.1, and a lower value means more consistency (Saaty, 1982). In Round One, all consistency ratios were below 0.1, indicating the AHP results were consistent (Mu & Pereyra-Rojas, 2017).

As shown in **Table 5. 10**, the most important category of DevSecOps challenges was "Organisation, People and Culture," with a normalised value of approximately 0.55, significantly higher than the other three categories. (All results were rounded to two decimal places for reporting.) The "Process Capabilities" category ranked second with a priority of 0.20, followed by "Business" with a priority of 0.17. The least important category was "Technology", with the lowest priority of 0.08. This result reveals the different focuses of DevSecOps challenges in reality and in the literature. The MLR findings presented in Chapter 3 show that a majority of the identified challenges are technology-related (Zhao, Clear, & Lal, 2024b). This difference implies that technological challenges are the easiest to identify and overcome among these DevSecOps challenges.

Table 5. 10 - AHP results for categories of challenges

Categories of Challenges	Normalised Value	Idealised Value	Ranking
Business	0.17293954718544899	0.31545340844252778	3
Organisation, People and Culture	0.54822532442839877	1.0	1 (highest)
Process Capabilities	0.20006106203563501	0.36492488238157644	2
Technology	0.078774066350517175	0.14368921470866036	4
consistency ratio = 0.04663			

Table 5. 11 lists the priorities and rankings of nine DevSecOps challenges in the "Organisation, People & Culture" (OPC) category. The top three important OPC-related challenges were: "C04 – Lack of security awareness and responsibility"; "C03 – Neglecting security"; and "C06 – Recruiting challenges". C04 and C03 both point out the difficulty of building a security-awareness culture. C04 is more focused on individual employees, while C03 emphasises the corporate culture and organisational consciousness. C06 reflects a real-world situation where DevSecOps

organisations struggle to recruit suitable talent.

Table 5. 11 - AHP results for challenges in "Organisation, People & Culture" category

Challenges in Organisation, People and Culture	Normalised Value	Idealised Value	Ranking
C01 - Cultural resistance and organisational opposition	0.088579069370106789	0.38565563449669021	7
C02 - Challenges of collaboration, communication and coordination	0.064891977073468152	0.2825267500547784	8
C03 - Neglecting security	0.17281650611212448	0.75240866482465463	2
C04 - Lack of security awareness and responsibility	0.22968436461639974	1.0	1
C05 - Lack of security knowledge and skills, need for training	0.11095377365292257	0.48307064278506112	4
C06-Recruiting challenges	0.11416175567088627	0.49703755787447701	3
C07 - Inconsistent security polices design	0.10067299594143227	0.43831018323588622	5
C08 - Challenges of governance and leadership	0.092745510629789349	0.40379549032292816	6
C09 - Lacking confidence	0.025494046932870273	0.11099600521545457	9
consistency ratio = 0.04642			

Table 5. 12 lists the priorities and rankings of eight DevSecOps challenges in the "Process Capabilities" (PC) category. The three most important challenges were: "C17 – Inadequate privileged credentials and access controls causing cyber-attack"; "C15 – Ignoring processes and security essentials leading to technical and security debt"; and "C12 – Compliance challenges".

Table 5. 12 - AHP results for challenges in "Process Capabilities" category

Challenges in Process Capabilities	Normalised Value	Idealised Value	Ranking
C10 - Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	0.060700947267667192	0.20616300948926211	6
C11 - Using unsuitable metrics	0.04105990078931674	0.13945470535622151	8
C12 - Compliance challenges	0.16193158486625681	0.54997993227653874	3
C13 - Neglecting change control in security	0.1142938319648401	0.38818439290621332	4
C14 - Lack of standards	0.081480788031473683	0.27673908286886018	5
C15 - Ignoring processes and security essentials leading to technical and security debt	0.18673224404672531	0.63421220152625357	2
C16 - Poor visibility of security track record	0.05936889677495108	0.20163887023392157	7
C17 - Inadequate privileged credentials and access controls causing cyber attacks	0.29443180625876908	1.0	1
consistency ratio = 0.04369			

Table 5. 13 reports the priorities and rankings of seven DevSecOps challenges in the "Technology" category. The three most important technological challenges were "C20 – Challenges of legacy system refactoring"; "C23 – Availability and reliability of infrastructure, tools, automation, and network bandwidth"; and "C24 – Continuous deployment chaos".

Table 5. 13 - AHP results for challenges in "Technology" category

Challenges in Technology	Normalised Value	Idealised Value	Ranking
C18 - Lack of mature tools for automation and security	0.09759623597928975	0.39257094283837307	5
C19 - Complexity in managing different tools	0.12931604763720622	0.52016066230097924	4
C20 - Challenges of legacy system refactoring	0.24860789561664393	1.0	1
C21 - Use of cloud and serverless computing brings security complications	0.093201141754257213	0.37489212288725687	6
C22 - Containers and other tools come with their risks	0.062882471581701224	0.25293835268477022	7
C23 - Availability and reliability of infrastructure, tools, automation, and network bandwidth	0.22047614690792383	0.88684289918088699	2
C24 - Continuous deployment chaos	0.14792006052297782	0.59499341384986482	3
consistency ratio = 0.04943			

Table 5. 14 presents the AHP pairwise results for the four DevSecOps business challenges. "C26 – Conflicts between security and business" was the most critical business challenge, with a priority of 0.45, significantly higher than others. It implies a prevailing problem: security requirements and business goals appear to conflict in practice. "C27 – Customer readiness for frequent releases" ranked second, with a priority of 0.28, followed by "C28 – Training users for using advanced tools" with 0.16. They both pose the challenges of assimilating customers into DevSecOps. The least important business challenge was "C25 – Challenges of cost control", with the lowest priority of 0.11. That is probably because C25 is a very general business challenge, not specific to DevSecOps.

Table 5. 14 - AHP results for challenges in "Business" category

Challenges in Business	Normalised Value	Idealised Value	Ranking
C25 - Challenges of cost control	0.1059244470719073	0.23667469807827529	4
C26 - Conflicts between security and business	0.44755289826914652	1.0	1
C27 - Customer readiness for frequent releases	0.2829011223553482	0.63210655868710053	2
C28 - Training users for using advanced tools	0.16362153230359791	0.36559149306458122	3
consistency ratio = 0.02660			

Overall priorities and rankings:

To derive the overall priorities/weights of the identified challenge, the normalised value of each challenge (in **Table 5. 11**, **Table 5. 12**, **Table 5. 13**, and **Table 5. 14**) was multiplied by the normalised value of its corresponding category (in **Table 5. 10**), according to **Equation (5)**:

$$\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^{T}\mathbf{1} = 1 \end{cases}$$ (Page 73).

The overall priorities of the 28 identified DevSecOps challenges are listed in **Table 5. 15**, ordered by ranking, with higher priorities indicating greater importance. The results were shared with the participants to help them complete the second round of the survey. A spreadsheet was also provided to present these results, allowing participants to sort by columns such as challenge ID or challenges' category. The least important challenges might be removed if they still have very low priorities after the completion of Round Two, based on participants' consensus on their low rankings.

Table 5. 15 - Overall priorities and rankings of DevSecOps challenges

Overall Ranking	Overall Priority	DevSecOps Challenge	Challenges Category
1	0.125918785	C04 - Lack of security awareness and responsibility	Organisation, People & Culture
2	0.094742385	C03 - Neglecting security	Organisation, People & Culture
3	0.077399596	C26 - Conflicts between security and business	Business
4	0.062586366	C06 - Recruiting challenges	Organisation, People & Culture
5	0.060827669	C05 - Lack of security knowledge and skills, need for training	Organisation, People & Culture
6	0.05890434	C17 - Inadequate privileged credentials and access controls causing cyber attacks	Process Capabilities
7	0.055191486	C07 - Inconsistent security polices design	Organisation, People & Culture
8	0.050845438	C08 - Challenges of governance and leadership	Organisation, People & Culture
9	0.048924792	C27 - Customer readiness for frequent releases	Business
10	0.048561289	C01 - Cultural resistance and organisational opposition	Organisation, People & Culture
11	0.037357851	C15 - Ignoring processes and security essentials leading to technical and security debt	Process Capabilities
12	0.035575425	C02 - Challenges of collaboration, communication and coordination	Organisation, People & Culture
13	0.032396205	C12 - Compliance challenges	Process Capabilities
14	0.028296634	C28 - Training users for using advanced tools	Business
15	0.022865745	C13 - Neglecting change control in security	Process Capabilities
16	0.019583855	C20 - Challenges of legacy system refactoring	Technology

17	0.018318526	C25 - Challenges of cost control	Business
18	0.017367803	C23 - Availability and reliability of infrastructure, tools, automation, and network bandwidth	Technology
19	0.016301133	C14 - Lack of standards	Process Capabilities
20	0.013976482	C09 - Lacking confidence	Organisation, People & Culture
21	0.012143896	C10 - Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	Process Capabilities
22	0.011877405	C16 - Poor visibility of security track record	Process Capabilities
23	0.011652265	C24 - Continuous deployment chaos	Technology
24	0.010186751	C19 - Complexity in managing different tools	Technology
25	0.008214487	C11 - Using unsuitable metrics	Process Capabilities
26	0.007688052	C18 - Lack of mature tools for automation and security	Technology
27	0.007341833	C21 - Use of cloud and serverless computing brings security complications	Technology
28	0.004953508	C22 - Containers and other tools come with their own risks	Technology

In Round One, four of the top five most important challenges were in the "Organisation, People and Culture" category, depending on the highest category priority (in **Table 5. 10** on Page 118). One business challenge (C26), by contrast, ranked third despite not having a high category priority, reflecting its high importance. The lowest-priority "Technology" category resulted in some technological challenges clustered at the bottom of the table.

5.2.2 Open-ended Question Results – Additional Challenges

In addition to the 17 AHP pairwise-comparison questions, Round One included an open-ended question (Question 18), which allowed participants to provide comments on additional DevSecOps challenges. Ten participants provided comments, while the remaining eight participants skipped this question.

Similar to the data synthesis step in the MLR study, the Thematic Analysis (TA) method was adopted to code and theme the qualitative data, but performed in reverse order. It initially followed a deductive approach, attempting to map the newly identified codes to the existing themes in the DevSecOps CPTM Model (Version 1.0). Subsequently, coding and theming were directed by the content of new data in an inductive approach (Braun & Clarke, 2021).

As a result, as shown in **Table 5. 16**, seven new themes/challenges were derived and further grouped into four categories. In Round Two, these emerging challenges would be compared with

the original ones to replace the least important initial ones. By doing so, the DevSecOps CPTM Model (Version 1.0) would be refined via iterative improvement.

Table 5. 16 - Additional new challenges

Comment/Code	Handling	New Theme/Challenge	Category
"Product teams are going rogue and not following DevSecOps practices."	Included	NC01 – Product teams are not following DevSecOps practices	Organisation, People & Culture
"Improper or inadequate risk assessment and management."	Included	NC02 – Challenges of risk assessment and management	Process Capabilities
"Lack of reference model for DevSecOps process."	Included	NC03 – Lacking reference of DevSecOps model	Process Capabilities
"Domain of application is also matters like cyber physical systems"	Included	NC04 – Domain of application	Technology
"Code security / code developers working on their dependences / logic, excluding external tools."	Included	NC05 – Developers and security teams exclude external tools.	Technology
"The ever-stricter security requirements for businesses that are increasingly challenging are underrepresented."	Included	NC06 – Challenges of security/DevSecOps requirements for businesses.	Business
"Compatibility mismatches between security requirements of external partners."	Included		
"Balance between risk, cost, and impact to business."	Included	NC07 – Balance between risk, cost, and impact to business	Business
"In general, the popularity of DevSecOps is less than DevOps in the IT industry, so that IT industry workers lack some of the knowledge and the awareness of importance in this area."	Overlapped with existing challenges.	N/A	Organisation, People & Culture
"Scaling of security measures across multiple teams, especially in large organisations or those with microservices architecture. It could be a vital aspect of DevSecOps as it ensures that security measures are integrated into the s/w dev and deployment instead of being considered an afterthought."	Considered a practice rather than a challenge, so keep it for Round 2.	N/A	Technology

5.3 RQ4 – DevSecOps Challenges in GSE Contexts

This section presents participants' opinions on the differences between local and global DevSecOps challenges by using a multiple-choice question (Question 19) and one open-ended

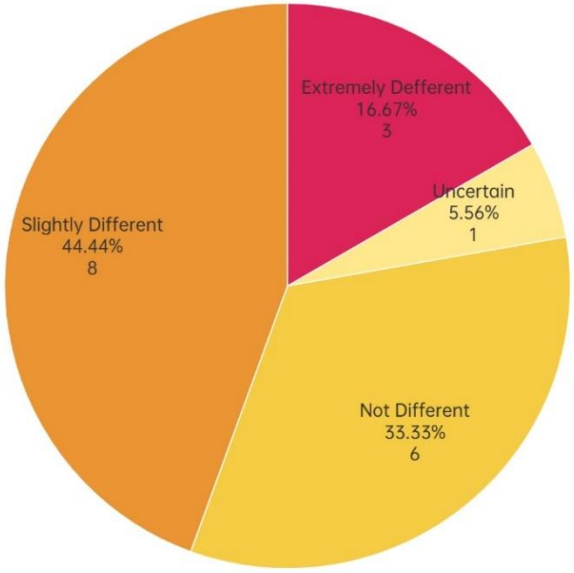
question (Question 20), to address RQ4 and its associated sub-questions.

5.3.1 Choice Question Results – "How different?"

To address sub-question 4.1, a multiple-choice question (Question 19) was raised to survey participants' opinions on "How do DevSecOps challenges differ in local and global settings?", including four options: "Extremely Different", "Slightly Different", "Not Different", and "Uncertain".

Figure 5. 2 presents a pie chart showing the proportions of these options based on participants' responses. 44.44% of participants selected "Slightly Different," which ranked first. "Not Different" ranked second, with 33.33%, followed by "Extremely Different" at 16.67%. Only one participant (5.56%) opted for "Uncertain".

Figure 5. 2 - Opinions on "How do DevSecOps challenges differ in local and global settings?"



This result corresponds to the MLR finding: "there is a notable absence of the global dimension of DevSecOps in both of the white and grey literature" (Zhao, Clear, & Lal, 2024b). Four assumptions have been presented in Section 3.3.2, Chapter 3 (Page 60).

The first assumption is that there are no distinguishing characteristics of DevSecOps, whether it is adopted in a local or global setting. Assumption two may be that security is typically a centralised and control-oriented function in organisations, so global aspects are not prominent. The third assumption is that there is a research gap between DevSecOps and GSE, and the fourth

one is due to an improper search string.

To a certain extent, this result corroborates the first and second assumptions. Most experts believe that DevSecOps challenges are only slightly different across local and global settings, which may lead to a lack of attention to global DevSecOps research, thereby supporting the third assumption.

5.3.2 Open-ended Question Results – "What are the differences?"

To address sub-question 4.2, at the end of the questionnaire survey, another open-ended question was included to collect participants' opinions on the differences between local and global DevSecOps challenges, if they selected "Extremely Different" or "Slightly Different" in Question 19. Of the eighteen participants, eleven responded to this question, while the remaining seven skipped it, having previously selected "Not Different" or "Uncertain" in Question 19.

As shown in **Table 5. 17**, the Thematic Analysis (TA) method was again employed to code and theme participants' comments. Eventually, five global DevSecOps challenges were identified, which were used to update the DevSecOps CPTM Model (Version 1.0), resulting in a global version (Version 2.0).

Table 5. 17 - Differences between local and global DevSecOps challenges

Comment/Code	Theme/Global Challenge	Category
"It amplifies challenges in team collaboration and communication."	CG01 – Amplified challenges of collaboration, communication, and coordination due to remote work	Organisation, People & Culture
"More remote work, less direct communication, more ease of misunderstanding, increased communication need in writing."		
"Synchronisation/Coordination between remote teams."		
"Issues of data residency and, for example, different nation's regulations about personal data can complicate things more than if everything is done in a single country/region."	CG02 – Challenges of data residency and management due to more inconsistent policies, regulations, and laws worldwide.	Organisation, People & Culture
"Compliance with external regulations worldwide."		
"Complex regulatory requirements"		
"Due to the law policies and private data limitation, many data are prohibited in some areas or countries, so that it is difficult to manage data globally in DevSecOps. Otherwise, if managing data locally, it will not face such challenges."		
"Everyone views DevSecOps differently. Such phenomenal magnified in global settings compared to what we see in local settings."	CG03 – Magnified different understanding of DevSecOps culture in global settings.	Organisation, People & Culture

"The dev and deployment can vary significantly between local and global settings."		
"Disagreements on best practices."		
"Risks/Threats spanning multiple regions. Independent development can propagate risks in a non-uniform way."	CG04 – Magnified challenges of risks and threats in global settings.	Process Capabilities
"Business focus and level of importance is different."	CG05 – Different business focus and levels of importance in global settings.	Business

This result indicates that global DevSecOps challenges primarily focus on the Organisation, People, and Culture perspectives. In addition, these global DevSecOps challenges can be seen as an amplification of the fundamental DevSecOps challenges. For example:

- "CG01 – Amplified challenges of collaboration, communication and coordination due to remote work" is an amplification of "C02 – Challenges of collaboration, communication and coordination".
- "CG02 – Challenges of data residency and management due to more inconsistent policies, regulations, and laws worldwide" is an amplification of "C07 – Inconsistent security polices design".
- "CG03 – Magnified different understanding of DevSecOps culture in global settings" is an amplification of "C01 – Cultural resistance and organisational opposition".
- "CG04 – Magnified challenges of risks and threats in global settings" is an amplification of "NC02 – Challenges of risk assessment and management".
- "CG05 – Different business focus and levels of importance" is an amplification of "C26 – Conflicts between security and business".

By summarising the results of Question 19 and Question 20, a conclusion could be drawn to partly answer RQ4, that is: DevSecOps challenges slightly differ between local and global contexts; the differences mainly focus on the aspect of Organisation, People and Culture; and specific existing challenges will be magnified if it is transferred from a local to a global setting.

5.4 Dissent Analysis for Round One

In this section, a dissent analysis is presented for Round One to examine the degree of consensus or dissent among participants regarding the AHP comparison results and the global DevSecOps challenges, using both quantitative and qualitative methods.

5.4.1 Dissent on AHP Results in Round One

To address *Sub-question 3.2: "Will the experts have dissent on the prioritisation due to their different roles (e.g. academic, industrial, technical, and managerial)?"*, a dissent analysis was conducted to discuss participants' agreements and disagreements on AHP comparison results, based on three levels: participants' individual level, participants' groups level, and participants' roles level:

- Individual level: to analyse the consensus or dissent among each participant regardless of the participant's position and role.
- Groups level: to analyse the consensus or dissent between the four participants' groups (i.e., Group AT, Group AM, Group IT, and Group IM).
- Roles level: to analyse the consensus or dissent between the four participants' roles (i.e., Academic, Industrial, Managerial, and Technical).

Based on the four-quadrant division for participants grouping on **Figure 5. 1** (Page 113), the datasets were merged as follows:

- Two datasets for Groups AM and AT were merged into a single dataset for the Academic role.
- Two datasets for Groups IM and IT were merged into a single dataset for the Industrial role.
- Two datasets for Groups AT and IT were merged into a single dataset for the Technical role.
- Two datasets for Groups AT and IT were merged into a single dataset for the Managerial role.

Packeer Mohamed et al. (2022) note that the AHP method is adaptable to both groups and individuals, and it fosters compromise and consensus-building. Based on experience and learnings from Round One, it can be observed that AHP is most suitable for small to medium-sized groups (e.g., 8 to 20 people), rather than for individuals, extremely small groups (fewer than

five people), or relatively large groups (over 30 people). Using AHP with fewer than five participants could lead to inconsistencies in the comparisons. Using AHP with large groups can lead to overly-compromised comparison results (i.e., converging to the "Equally important" scale), which may narrow the ranking gaps and render the AHP method less meritorious.

Therefore, the dissent analysis was performed based on participants' roles rather than the small groups, as this was better suited to addressing *Sub-question 3.2: "Do experts have dissent on the prioritisation due to their different roles, e.g., academic, industrial, technical, and managerial?"*.

In addition, Coefficients of variation (CVs) were calculated to evaluate the degree of dissent at the individual level, thereby analysing consensus or dissent for each participant regardless of position or role, to determine whether participants with similar roles would have more consensual opinions or whether individual variation is the primary factor.

Dissent analysis based on individuals:

Coefficients of variations (CVs) of all 97 AHP comparison results were calculated to evaluate the degree of dissent among participants based on the individual level, to verify whether participants with similar roles would have more consensual opinions, or if individual variations are the primary factor. CV is defined as the ratio of the standard deviation to the arithmetic mean, and a higher value indicates greater dissimilarity among participants (Everitt, 2006).

For this research, the acceptable value of CV was defined as below 0.5 (Dajani, Sincoff, & Talley, 1979). Rather than reporting all 97 CVs individually in Round One, **Table 5. 18** shows the average value for the overall CVs (i.e., all 18 participants in Round One) and for the CVs across four groups and four roles. Results were compared and discussed to investigate whether participants in the same group or with the same roles had more consensual results. Most of the CV values were below 0.5, i.e., within the acceptable range defined by Dajani, Sincoff and Talley (1979), while all were below 0.8, i.e., within a wider acceptable range defined by Rho (2006).

Table 5. 18 - Coefficient of variations in Round One

Overall CVs (average value)	**CVs in 4 Groups (average value)**			
0.47	Group AT: 0.39	Group AM: 0.39	Group IT: 0.35	Group IM: 0.55
	CVs in 4 Roles (average value)			
	Technical: 0.42	Managerial: 0.48	Academic: 0.39	Industrial: 0.52

The average value of overall CVs was 0.47, three of the four participants' groups had lower values than 0.47, respectively: Group Academic Technical (AT) – 0.39; Group Academic Managerial (AM) – 0.39; Group Industrial Technical (IT) – 0.35, indicating that the participants in these three groups held more consensual opinions on the prioritisation of DevSecOps challenges. However, there was an exception: Group Industrial Managerial (IM) had a CV of 0.55, which was higher than the overall value of 0.47. In other words, the degree of dissent among the five participants (Pa14, Pa15, Pa16, Pa17, and Pa18) in this group was slightly higher, despite their similar roles

If CVs were compared based on participants' roles rather than groups, academic experts and technical experts held more consensual opinions, with CVs of 0.39 and 0.42, respectively, which were lower than the overall value of 0.47. In contrast, industrial and managerial experts expressed more dissenting opinions, with CVs of 0.52 and 0.48, respectively, exceeding the overall value of 0.47, primarily due to dissent among the five participants (Pa14, Pa15, Pa16, Pa17, and Pa18).

A conjecture for this result is that the five participants are all industrial experts who manage at the portfolio level within their organisations, as shown in **Table 5. 2** (Page 112), and they may possess unique insights and specialised opinions relevant to their own organisations. During the survey, these participants were so confident in their judgments that they were more likely to assign higher scores in AHP pairwise comparisons (e.g., opting for "8" or "9," which represents "Extremely more important"). By contrast, participants in other groups judged AHP comparisons more mildly and cautiously (e.g., opting for "2" or "3," which represents "Moderately more important"), making agreement more likely with them. Arnold et al. (2000) conducted relevant experiments, and the results showed that highly experienced practitioners (partners, managers, and directors) exhibited greater levels of dissent and bias due to overconfidence than did moderately experienced and insolvency practitioners (seniors and staff).

Dissent analysis based on groups:

There were noticeable disagreements among participants' groups about the priority and ranking of categories. **Table 5. 19** presents the AHP comparison results for the four categories in four participants' groups, compared with the overall results.

Table 5. 19 - Round One AHP results for categories based on groups

Categories of Challenges	Priority and Ranking				
	Overall	**Group AT**	**Group AM**	**Group IT**	**Group IM**
Organisation, People & Culture	**0.55, 1st**	0.54, 1st	0.49, 1st	0.59, 1st	0.34, 2nd (↓1)
Process Capabilities	**0.20, 2nd**	0.30, 2nd	0.19, 3rd (↓1)	0.25, 2nd	0.13, 3rd (↓1)
Business	**0.17, 3rd**	0.10, 3rd	0.14, 4th (↓1)	0.07, 4th (↓1)	0.48, 1st (↑2)
Technology	**0.08, 4th**	0.06, 4th	0.22, 2nd (↑2)	0.09, 3rd (↑1)	0.05, 4th

- For the "Organisation, People and Culture" category, which was the least controversial, three groups (Groups AT, AM, and IT) ranked it first, consistent with the overall result, while Group IM ranked it second.
- For the "Process Capabilities" category, two groups (Groups AT and IT) ranked it second, the same as the overall result; the other two groups (Groups AM and IM) ranked it third.
- For the "Technology" category, two groups (Groups AT and IM) rated it as the least important category, mirroring the overall result; Groups AM and IT ranked it second and third, respectively.
- For the "Business" category, which was the most controversial, only Group AT ranked it third, consistent with the overall result. Group IM rated "Business" as the most important category, whereas Groups AM and IT ranked it the lowest.

These different priorities across categories led to increasingly distinct priorities and rankings of the corresponding challenges, as listed in **Table 5. 20**.

Table 5. 20 - Round One overall priorities and rankings of DevSecOps challenges based on groups (ordered by overall ranking)

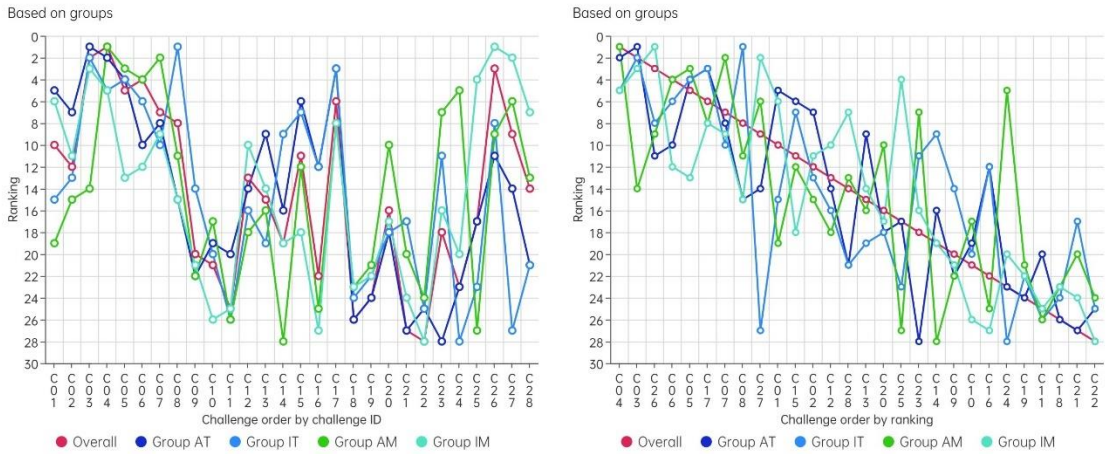
DevSecOps Challenge (Category: OPC, PC, B, T)	Overall Priority, Ranking within Category, Overall Ranking				
	Overall	**Group AT**	**Group AM**	**Group IT**	**Group IM**
C04 - Lack of security awareness and responsibility (OPC)	**0.126, 1, 1**	0.124, 2 (↓1), 2 (↓1)	0.119, 1 (-), 1 (-)	0.072, 4 (↓3), 5 (↓4)	0.061, 2 (↓1), 5 (↓4)
C03 - Neglecting security (OPC)	**0.095, 2, 2**	0.134, 1 (↑1), 1 (↑1)	0.029, 6 (↓4), 14 (↓12)	0.130, 2 (-), 2 (-)	0.083, 1 (↑1), 3 (↓1)
C26 - Conflicts between security and business (Business)	**0.078, 1, 3**	0.036, 1 (-), 11 (↓8)	0.043, 2 (↓1), 9 (↓6)	0.043, 1 (-), 8 (↓5)	0.194, 1 (-), 1 (↑2)
C06 - Recruiting challenges (OPC)	**0.063, 3, 4**	0.038, 7 (↓4), 10 (↓6)	0.062, 4 (↓1), 4 (-)	0.067, 5 (↓2), 6 (↓2)	0.029, 6 (↓3), 12 (↓8)
C05 - Lack of security knowledge and	**0.061,**	0.066,	0.063,	0.080,	0.024,

skills, need for training (OPC)	**4, 5**	3 (↑1), 4 (↑1)	3 (↑1), 3 (↑2)	3 (↑1), 4 (↑1)	7 (↓3), 13 (↓8)
C17 - Inadequate privileged credentials and access controls causing cyber-attacks (PC)	**0.059, 1, 6**	0.084, 1 (-), 3 (↑3)	0.047, 1 (-), 8 (↓2)	0.086, 1 (-), 3 (↑3)	0.039, 1 (-), 8 (↓2)
C07 - Inconsistent security polices design (OPC)	**0.055, 5, 7**	0.041, 6 (↓1), 8 (↓1)	0.066, 2 (↑3), 2 (↑5)	0.033, 6 (↓1), 10 (↓3)	0.035, 4 (↑1), 9 (↓2)
C08 - Challenges of governance and leadership (OPC)	**0.051, 6, 8**	0.030, 8 (↓2), 15 (↓7)	0.041, 5 (↑1), 11 (↓3)	0.143, 1 (↑5), 1 (↑7)	0.019, 8 (↓2), 15 (↓7)
C27 - Customer readiness for frequent releases (Business)	**0.049, 2, 9**	0.032, 2 (-), 14 (↓5)	0.051, 1 (↑1), 6 (↑3)	0.005, 4 (↓2), 27 (↓18)	0.174, 2 (-), 2 (↑7)
C01 - Cultural resistance and organisational opposition (OPC)	**0.048, 7, 10**	0.055, 4 (↑3), 5 (↑5)	0.022, 8 (↓1), 19 (↓9)	0.018, 9 (↓2), 15 (↓5)	0.054, 3 (↑4), 6 (↑4)
C15 - Ignoring processes and security essentials leading to technical and security debt (PC)	**0.037, 2, 11**	0.047, 2 (-), 6 (↑5)	0.039, 2 (-), 12 (↓1)	0.048, 2 (-), 7 (↑4)	0.011, 4 (↓2), 18 (↓7)
C02 - Challenges of collaboration, communication, and coordination (OPC)	**0.036, 8, 12**	0.041, 5 (↑3), 7 (↑5)	0.029, 7 (↑1), 15 (↓3)	0.027, 7 (↑1), 13 (↓1)	0.029, 5 (↑3), 11 (↑1)
C12 - Compliance requirements (PC)	**0.032, 3, 13**	0.033, 5 (↓2), 14 (↓1)	0.023, 5 (↓2), 18 (↓5)	0.016, 5 (↓1), 16 (↓3)	0.034, 2 (↑1), 10 (↑3)
C28 - Training users for using advanced tools (B)	**0.028, 3, 14**	0.013, 4 (↓1), 21 (↓7)	0.034, 3 (-), 13 (↑1)	0.012, 2 (↑1), 21 (↓7)	0.045, 4 (↓1), 7 (↑7)
C13 - Neglecting change control in security (PC)	**0.023, 4, 15**	0.041, 3 (↑1), 9 (↑6)	0.024, 3 (↑1), 16 (↓1)	0.014, 6 (↓2), 19 (↓4)	0.019, 3 (↑1), 14 (↑1)
C20 - Challenges of legacy system refactoring (T)	**0.020, 1, 16**	0.022, 1, 18 (↓2)	0.042, 3 (↓2), 10 (↑6)	0.015, 3 (↓2), 18 (↓2)	0.011, 2 (↓1), 17 (↓1)
C25 - Challenges of cost control (B)	**0.018, 4, 17**	0.024, 3 (↑1), 17 (-)	0.009, 4 (-), 27 (↓10)	0.011, 3 (↑1), 23 (↓6)	0.071, 3 (↑1), 4 (↑13)
C23 - Availability and reliability of infrastructure, tools, automation, and network bandwidth (T)	**0.017, 2, 18**	0.003, 7 (↓5), 28 (↓10)	0.050, 2 (-), 7 (↑11)	0.029, 1 (↑1), 11 (↑7)	0.016, 1 (↑1), 16 (↑2)
C14 - Lack of standards (PC)	**0.016, 5, 19**	0.027, 6 (↓1), 16 (↑3)	0.008, 7 (↓2), 28 (↓9)	0.036, 3 (↑2), 9 (↑10)	0.011, 5 (-), 19 (-)
C09 - Lacking confidence (OPC)	**0.014, 9, 20**	0.011, 9 (-), 22 (↓2)	0.018, 9 (-), 22 (↓2)	0.019, 8 (↑1), 14 (↑6)	0.006, 9 (-), 21 (↓1)
C10 - Difficulties in integrating security into DevOps without losing speed and affecting current process and performance (PC)	**0.012, 6, 21**	0.018, 7 (↓1), 19 (↑2)	0.024, 4 (↑2), 17 (↑4)	0.014, 7 (↓1), 20 (↑1)	0.004, 7 (↓1), 26 (↓5)
C16 - Poor visibility of security track record (PC)	**0.012, 7, 22**	0.035, 4 (↑3), 12 (↑10)	0.014, 6 (↑1), 25 (↓3)	0.028, 4 (↑3), 12 (↑10)	0.004, 8 (↓1), 27 (↓5)
C24 - Continuous deployment chaos (T)	**0.012, 3, 23**	0.009, 2 (↑1), 23 (-)	0.059, 1 (↑2), 5 (↑17)	0.004, 7 (↓4), 28 (↓5)	0.007, 3 (-), 20 (↑3)
C19 - Complexity in managing different tools (T)	**0.011, 4, 24**	0.008, 3 (↑1), 24 (-)	0.020, 5 (↓1), 21 (↑3)	0.011, 4 (-) , 22 (↑2)	0.006, 4 (-), 22 (↑2)

C11 - Using unsuitable metrics (PC)	**0.008, 8, 25**	0.017, 8 (-), 20 (↑5)	0.012, 6 (↑2), 26 (↓1)	0.006, 8 (-), 26 (↓1)	0.004, 6 (↑2), 25 (-)
C18 - Lack of mature tools for automation and security (T)	**0.008, 5, 26**	0.005, 5 (-), 26 (-)	0.017, 6 (↓1), 23 (↑3)	0.009, 5 (-), 24 (↑2)	0.005, 5 (-), 23 (↑3)
C21 - Use of cloud and serverless computing brings security complications (T)	**0.007, 6, 27**	0.004, 6 (-), 27 (-)	0.020, 4 (↑2), 20 (↑7)	0.016, 2, 17 (↑10)	0.005, 6 (-), 24 (↑3)
C22 - Containers and other tools come with their own risks (T)	**0.005, 7, 28**	0.005, 4 (↑3), 25 (↑3)	0.016, 7 (-), 24 (↑4)	0.009, 6 (↑1), 25 (↑3)	0.002, 7 (-), 28 (-)

Suppose the overall rankings were compared directly with those obtained from four participants' groups, as shown in **Figure 5. 3**. In this case, there were significant discrepancies in the rankings of DevSecOps challenges at the group level. The red line on the graph represents the overall rankings of 28 DevSecOps challenges; the other four coloured lines represent the rankings given by four participants' groups, respectively: Group Academic Technical (AT), Group Industrial Technical (IT), Group Academic Managerial (AM), and Group Industrial Managerial (IM).

Figure 5. 3 - Round One dissents on rankings of DevSecOps challenges based on groups



There are two distinct viewpoints illustrated in **Figure 5. 3**. On the left-hand side, 28 challenges are ordered by their identifiers from C01 to C28, showing the differences in rankings within each category. For example, it can be seen that the challenges in the "Organisation, People and Culture" (C01 – C09) and "Business" categories (C25 – C28) had more different rankings than the challenges in the "Process Capacities" (C10 – C17) and "Technology" categories (C18 – C24). Whereas the line graph on the right-hand side of **Figure 5. 3** orders the 28 DevSecOps challenges

by rankings from first to 28th place, showing the differences in rankings more visually. The further a point is from the red line, the greater the difference between the two rankings.

Dissent analysis based on roles:

To address *Sub-question 3.2: "Will the experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*, the dissent analysis was further conducted by role rather than by group.

Table 5. 21 reports AHP results for the four categories, based on four role types: Academic, Industrial, Technical, and Managerial. The two datasets of Groups AM and AT were merged into the Academic role; Groups IM and IT were merged into the Industrial role; Groups AT and IT were merged into the Technical role; and Groups AM and IM were merged into the Managerial role. By doing so, the compromise and consensus were built more solidly, relying on a larger number of participants (Packeer Mohamed et al., 2022). Taking this knowledge into the remaining rounds of the Delphi survey (Rounds Two and Three), the dissent analysis for AHP results would be conducted only based on participants' roles, rather than small groups.

Table 5. 21 - Round One AHP results for categories based on roles

Categories of Challenges	Priority and Ranking				
	Overall	**Academic (Group AT+AM)**	**Industrial (Group IT+IM)**	**Managerial (Group AM+IM)**	**Technical (Group AT+IT)**
Organisation, People & Culture	**0.55, 1st**	0.52, 1st	0.53, 1st	0.48, 1st	0.58, 1st
Process Capabilities	**0.20, 2nd**	0.26, 2nd	0.17, 3rd (↓1)	0.15, 3rd (↓1)	0.26, 2nd
Business	**0.17, 3rd**	0.13, 3rd	0.24, 2nd (↑1)	0.29, 2nd (↑1)	0.09, 3rd
Technology	**0.08, 4th**	0.09, 4th	0.06, 4th	0.08, 4th	0.06, 4th

As shown in **Table 5. 21**, participants across all four roles reached consensus that "Organisation, People and Culture" was the most important category and "Technology" was the least important. For the "Process Capabilities" and the "Business" categories, academic and technical roles ranked "Process Capabilities" second and "Business" third, consistent with the overall result. However, industrial and managerial roles did the opposite, rating "Business" highly over "Process Capabilities". This suggests that industrial professionals, particularly those in managerial roles and higher positions, tend to prioritise business challenges. **Table 5. 22** provides the overall

priorities and rankings of DevSecOps challenges, based on participants' roles.

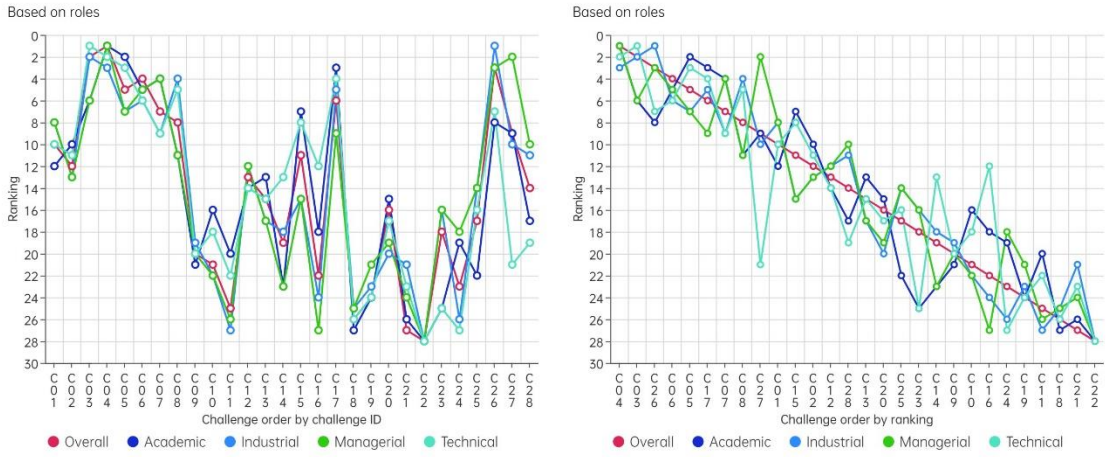
Table 5. 22 - Round One overall priorities and rankings of DevSecOps challenges based on roles (order by overall ranking)

DevSecOps Challenge (Category: OPC, PC, B, T)	Overall Priority, Ranking within Category, Overall Ranking				
	Overall	**Academic**	**Industrial**	**Managerial**	**Technical**
C04 - Lack of security awareness and responsibility (OPC)	**0.126, 1, 1**	0.150, 1 (-), 1 (-)	0.091, 2 (↓1), 3 (↓2)	0.122, 1 (-), 1 (-)	0.093, 2 (↓1), 2 (↓1)
C03 - Neglecting security (OPC)	**0.095, 2, 2**	0.053, 5 (↓3), 6 (↓4)	0.129, 1 (↑1), 2 (-)	0.057, 4 (↓2), 6 (↓4)	0.151, 1 (↑1), 1 (↑1)
C26 - Conflicts between security and business (Business)	**0.078, 1, 3**	0.044, 1 (-), 8 (↓5)	0.132, 1 (-), 1 (↑2)	0.111, 2 (↓1), 3 (-)	0.047, 1 (-), 7 (↓4)
C06 - Recruiting challenges (OPC)	**0.063, 3, 4**	0.057, 4 (↓1), 5 (↓1)	0.057, 4 (↓1), 6 (↓2)	0.060, 3 (-), 5 (↓1)	0.054, 5 (↓2), 6 (↓2)
C05 - Lack of security knowledge and skills, need for training (OPC)	**0.061, 4, 5**	0.074, 2 (↑2), 2 (↑3)	0.050, 5 (↓1), 7 (↓2)	0.048, 5 (↓1), 7 (↓2)	0.085, 3 (↑1), 3 (↑2)
C17 - Inadequate privileged credentials and access controls causing cyber-attacks (PC)	**0.059, 1, 6**	0.068, 1 (-), 3 (↑3)	0.057, 1 (-), 5 (↑1)	0.041, 1 (-), 9 (↓3)	0.082, 1 (-), 4 (↑2)
C07 - Inconsistent security polices design (OPC)	**0.055, 5, 7**	0.062, 3 (↑2), 4 (↑3)	0.047, 7 (↓2), 9 (↓2)	0.069, 2 (↑3), 4 (↑3)	0.037, 6 (↓1), 9 (↓2)
C08 - Challenges of governance and leadership (OPC)	**0.051, 6, 8**	0.036, 7 (↓1), 11 (↓3)	0.061, 3 (↑3), 4 (↑4)	0.034, 7 (↓1), 11 (↓3)	0.081, 4 (↑2), 5 (↑3)
C27 - Customer readiness for frequent releases (Business)	**0.049, 2, 9**	0.044, 2 (-), 9 (-)	0.046, 2 (-), 10 (↓1)	0.117, 1 (↑1), 2 (↑7)	0.013, 4 (↓2), 21 (↓12)
C01 - Cultural resistance and organisational opposition (OPC)	**0.048, 7, 10**	0.035, 8 (↓1), 12 (↓2)	0.049, 6 (↑1), 8 (↑2)	0.043, 6 (↑1), 8 (↑2)	0.036, 7 (-), 10 (-)
C15 - Ignoring processes and security essentials leading to technical and security debt (PC)	**0.037, 2, 11**	0.049, 2 (-), 7 (↑4)	0.024, 3 (↓1), 15 (↓4)	0.022, 3 (↓1), 15 (↓4)	0.044, 2 (-), 8 (↑3)
C02 - Challenges of collaboration, communication, and coordination (OPC)	**0.036, 8, 12**	0.042, 6 (↑2), 10 (↑2)	0.031, 8 (-), 13 (↓1)	0.030, 8 (-), 13 (↓1)	0.034, 8 (-), 11 (↑1)
C12 - Compliance requirements (PC)	**0.032, 3, 13**	0.033, 4 (↓1), 14 (↓1)	0.031, 2 (↑1), 12 (↑1)	0.032, 2 (↑1), 12 (↑1)	0.024, 5 (↓2), 14 (↓1)
C28 - Training users for using advanced tools (B)	**0.028, 3, 14**	0.024, 3 (-), 17 (↓3)	0.036, 3 (-), 11 (↑3)	0.041, 3 (-), 10 (↑4)	0.016, 3 (-), 19 (↓5)
C13 - Neglecting change control in security (PC)	**0.023, 4, 15**	0.034, 3 (↑1), 13 (↑2)	0.019, 4 (-), 17 (↓2)	0.022, 4 (-), 17 (↓2)	0.022, 6 (↓2), 15 (-)
C20 - Challenges of legacy system refactoring (T)	**0.020, 1, 16**	0.025, 1 (-), 15 (↑1)	0.012, 2 (↓1), 20 (↓4)	0.016, 3 (↓2), 19 (↓3)	0.018, 1 (-), 17 (↓1)
C25 - Challenges of cost control (B)	**0.018, 4, 17**	0.014, 4 (-), 22 (↓5)	0.030, 4 (-), 14 (↑3)	0.024, 4 (-), 14 (↑3)	0.019, 2 (↑2), 16 (↑1)
C23 - Availability and reliability of	**0.017,**	0.009,	0.019,	0.022,	0.007,

infrastructure, tools, automation, and network bandwidth (T)	**2, 18**	4 (↓2), 25 (↓7)	1 (↑1), 16 (↑2)	1 (↑1), 16 (↑2)	4 (↓2), 25 (↓7)
C14 - Lack of standards (PC)	**0.016, 5, 19**	0.013, 8 (↓3), 23 (↓4)	0.017, 5 (-), 18 (↑1)	0.009, 6 (↓1), 23 (↓4)	0.031, 4 (↑1), 13 (↑6)
C09 - Lacking confidence (OPC)	**0.014, 9, 20**	0.015, 9 (-), 21 (↓1)	0.012, 9 (-), 19 (↑1)	0.012, 9 (-), 20 (-)	0.014, 9 (-), 20 (-)
C10 - Difficulties in integrating security into DevOps without losing speed and affecting current process and performance (PC)	**0.012, 6, 21**	0.025, 5 (↑1), 16 (↑5)	0.008, 6 (-), 22 (↓1)	0.009, 5 (↑1), 22 (↓1)	0.016, 7 (↓1), 18 (↑3)
C16 - Poor visibility of security track record (PC)	**0.012, 7, 22**	0.023, 6 (↑1), 18 (↑4)	0.007, 7 (-), 24 (↓2)	0.006, 8 (↓1), 27 (↓5)	0.032, 3 (↑4), 12 (↑10)
C24 - Continuous deployment chaos (T)	**0.012, 3, 23**	0.020, 2 (↑1), 19 (↑4)	0.005, 6 (↓3), 26 (↓3)	0.016, 2 (↑1), 18 (↑5)	0.006, 6 (↓3), 27 (↓4)
C19 - Complexity in managing different tools (T)	**0.011, 4, 24**	0.012, 3 (↑1), 24 (-)	0.008, 4 (-), 23 (↑1)	0.010, 4 (-), 21 (↑3)	0.009, 3 (↑1), 24 (-)
C11 - Using unsuitable metrics (PC)	**0.008, 8, 25**	0.017, 7 (↑1), 20 (↑5)	0.005, 8 (-), 27 (↓2)	0.006, 7 (↑1), 26 (↓1)	0.009, 8 (-), 22 (↑3)
C18 - Lack of mature tools for automation and security (T)	**0.008, 5, 26**	0.008, 6 (↓1), 27 (↓1)	0.006, 5 (-), 25 (↑1)	0.008, 6 (↓1), 25 (↑1)	0.007, 5 (-), 26 (-)
C21 - Use of cloud and serverless computing brings security complications (T)	**0.007, 6, 27**	0.008, 5 (↑1), 26 (↑1)	0.008, 3 (↑3), 21 (↑6)	0.008, 5 (↑1), 24 (↑3)	0.009, 2 (↑4), 23 (↑4)
C22 - Containers and other tools come with their own risks (T)	**0.005, 7, 28**	0.007, 7 (-), 28 (-)	0.004, 7 (-), 28 (-)	0.004, 7 (-), 28 (-)	0.005, 7 (-), 28 (-)

Figure 5. 4 shows differences in rankings of DevSecOps challenges across the four participants' roles. The red line represents the overall rankings of 28 DevSecOps challenges; the other four coloured lines represent the rankings given by four participants' roles: Academic, Industrial, Managerial, and Technical.

Figure 5. 4 - Round One dissents on rankings of DevSecOps challenges based on roles



Compared to the ranking dissents based on groups in **Figure 5. 3** (Page 132), the dissents in **Figure 5. 4** have been significantly narrowed by the compromises within one type of role. Most rankings were not that different, but there were exceptions. For example, the right-hand graph in **Figure 5. 4** shows that the rankings of C27 (within the "Business" category) and C16 (within the "Process Capabilities" category) exhibited the most pronounced differences between roles, particularly between managerial and technical participants. **Table 5. 21** (Page 133) also reports that managerial roles and technical roles had conflicting rankings for the "Process Capabilities" and "Business" categories.

By summing up the AHP results in **Table 5. 18**, **Table 5. 19**, **Table 5. 20**, **Table 5. 21**, and **Table 5. 22**, and comparing the dissents in **Figure 5. 3** and **Figure 5. 4**, a conclusion could be drawn: in Round One, participants reached a rough consensus on the ranking of DevSecOps challenges and categories when analysing the data based on participants' roles; whereas significant dissent existed when analysing the data based on participants' groups or individuals.

In the following Round Two, Round One results would be shared with the participants to help them re-evaluate these challenges. As the Delphi iterations progressed, the dissents were reanalysed and reported in Chapter 6 to assess whether and how the participants' opinions had changed.

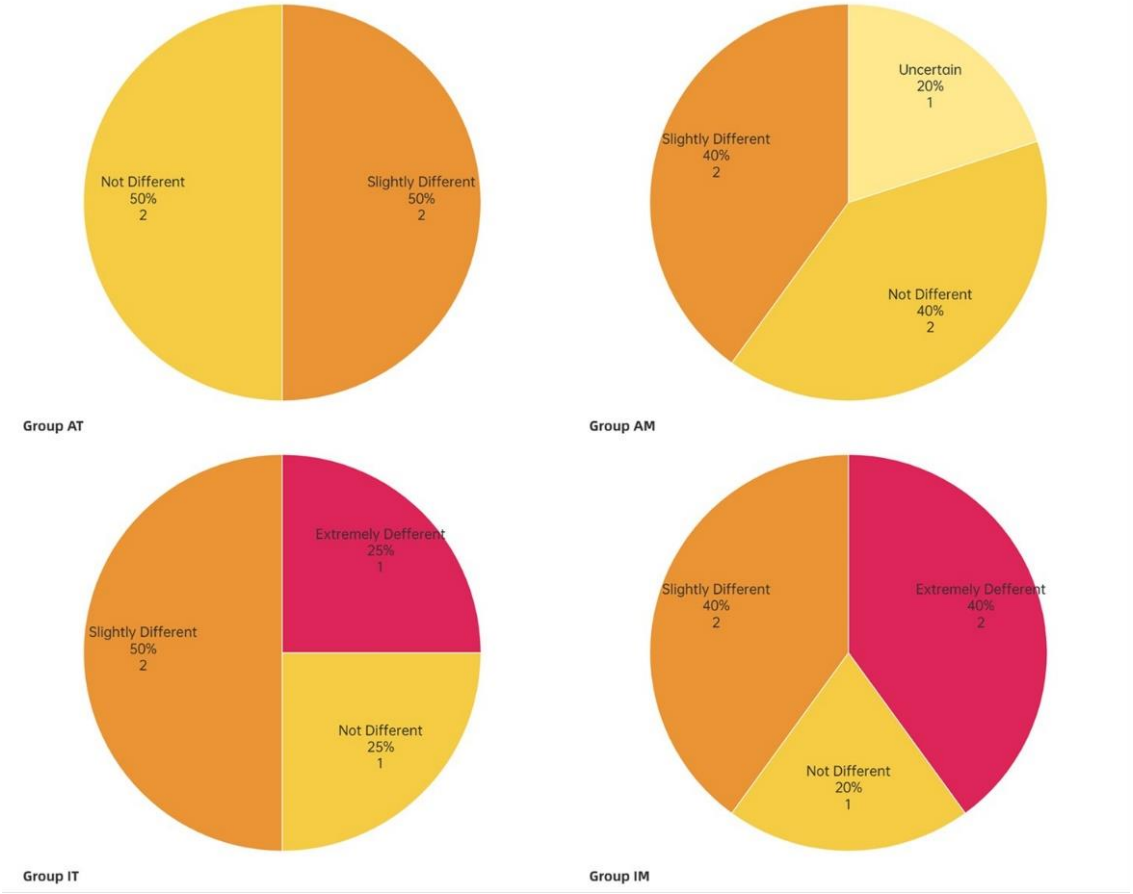
5.4.2 Dissent on Global DevSecOps Challenges in Round One

As shown in **Figure 5. 2** in Section 5.3.1 (Page 124), the opinions of "Slightly Different" and

"Not Different" ranked first and second, respectively, with 44.44% and 33.33%, together accounting for 77.77%.

The four pie charts in **Figure 5. 5** compare the opinions of the four groups. The two opinions of "Slightly Different" and "Not Different" were evenly distributed in each group. Three participants opted for "Extremely Different", and they were all industrial experts – two industrial managerial participants (Pa16 and Pa18) from New Zealand and one industrial technical participant (Pa12) from China.

Figure 5. 5 - Opinions on "How do DevSecOps challenges differ in local and global settings?" in four groups



As an island nation, New Zealand is inclined to be somewhat insular; the geographic distance may pose greater challenges to its limited business resources when it adopts DevSecOps in a global context. For instance, as mentioned by participant Pa18, who is from the SE industry of New Zealand, "it is often difficult to replicate the sophistication of global organisations in the NZ context due to constraints in budgets". Whereas, more global DevSecOps challenges in China,

such as "strict regulations about data residency and access" and "viewing DevSecOps differently," mentioned by two Chinese participants, Pa13 and Pa17, may be caused by cultural distances, which are more related to security policies, laws and rules, organisational structures, corporate culture, and governance. In addition, one participant from the Group Academic Managerial selected "Uncertain".

As shown in **Table 5. 17** in Section 5.3.2 (Page 125), five challenges of global DevSecOps are identified from 12 comments. Many overlaps and similarities exist in these comments and codes, suggesting a consensus among participants on the differences between local and global DevSecOps. However, these global challenges introduce no new themes and merely amplify the existing fundamental challenges. Thus, the opinions of "Slightly Different" and "Not Different" both make sense.

To sum up, regarding the differences between local and global DevSecOps challenges, participants reached a consensus: "DevSecOps challenges are barely or slightly different between local and global contexts, and certain existing challenges will be magnified in global settings".

5.5 Chapter Summary

This chapter presents the results of the Round One Delphi survey, involving 18 participants. In Round One, the importance of 28 identified DevSecOps challenges was compared using AHP pairwise comparisons across four categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology. The priorities and rankings of these challenges and categories were calculated and reported by using multiple tools, including Qualtrics, SuperDecisions, and MS Excel spreadsheets. Consistency ratios were evaluated to ensure the AHP results fell within an acceptable range.

In addition to AHP pairwise comparisons, seven additional DevSecOps challenges were collected from participants' comments. Furthermore, participants' opinions on the differences between local and global DevSecOps challenges were collected and reported. Most participants believe that DevSecOps differs slightly between local and global contexts; the differences mainly focus on the "Organisation, People, and Culture" aspects, and specific existing challenges will be

magnified when transferred from a local to a global setting.

A dissent analysis was conducted to examine participants' agreement and disagreement with the AHP comparison results and the global DevSecOps opinions, using both quantitative and qualitative methods. Due to the merit of the AHP method, which is compromise building, it could be concluded that: in Round One, participants reached a rough consensus on the ranking of DevSecOps challenges and categories based on participants' roles, but there were some inescapable dissenting opinions if we analysed data based on participants' groups (4 or 5 people) or individuals. In Round Two, the results from Round One would be shared with participants to help them re-evaluate these challenges. Regarding the difference between local and global DevSecOps challenges, the participants reached consensus in Round One, focusing on "Slightly Different" and "Not Different".

To sum up, the above findings have partly answered RQ3, RQ4, and associated sub-questions, i.e., have addressed the dimension of challenges. However, the evaluation of DevSecOps challenges was not completed during Round One, so a deep discussion on these findings has not been presented. The first round of the Delphi survey often had limitations, notably that the quality of the results and the efficiency of responses might be affected by participants' confusion with the questionnaire format and unfamiliarity with the AHP approach. This problem could be continually mitigated as the Delphi process went on.

Next, Chapter 6 reports the results of Round Two, in which participants initially reassessed the importance of 35 revised DevSecOps challenges and subsequently assessed the importance of 60 identified DevSecOps practices. Participants were required to review Round One results, which had been sent back to them before conducting Round Two, to verify and improve the findings iteratively.

6 Chapter 6: Round Two Results – Revised DevSecOps Challenges and DevSecOps Practices

This chapter presents the results of the second round of the Delphi survey, conducted between October and December 2024, involving 14 participants. In Round Two, the importance of 35 DevSecOps challenges (revised) and 60 DevSecOps practices was compared by using the AHP method (pairwise comparisons) across the four categories: Business, Organisation, People & Culture (OPC), Process Capabilities (PC), and Technology.

First, the priorities and rankings of 35 revised DevSecOps challenges (28 initially identified challenges plus seven new ones added during Round One) and four categories were calculated and reported to confirm the findings of Round One.

Second, the priorities and rankings of 60 identified DevSecOps practices were calculated and reported. Consistency ratios for all AHP comparisons were also calculated and reported to assess whether the AHP results were acceptable.

Third, based on participants' comments, three additional practices were identified and incorporated to improve the DevSecOps CPTM Model (Version 1.0).

Fourth, participants' opinions on the differences between local and global DevSecOps practices were collected and reported.

Finally, a dissent analysis was conducted to examine participants' agreement and disagreement with the AHP comparison results and global DevSecOps opinions, by using both quantitative and qualitative methods.

In Section 6.1, the context and background for conducting the Round Two survey are provided, including the survey objectives, questionnaire content, and participants' information. The AHP comparison results for the revised DevSecOps challenges are reported and analysed in Section 6.2. Next, the AHP comparison results for DevSecOps practices are presented in Section 6.3, along with three additional practices identified from participants' comments. In Section 6.4, participants' opinions on the difference between local and global DevSecOps practices are

reported and analysed. Section 6.5 provides information on the dissent analysis for the Round Two survey. Section 6.6 concludes the Round Two survey and provides information on the Round Three Delphi survey.

6.1 Context and Background of Round Two

This section provides the context and background for the second round of the Delphi survey, including its objectives, questionnaire content, and participants' information.

6.1.1 Survey Objectives in Round Two

The second round of the Delphi survey was conducted for multiple objectives.

First, to verify the Round One results, at the beginning of Round Two, the importance of 35 revised DevSecOps challenges was reassessed by using AHP-based closed-ended questions (i.e., pairwise comparisons). Before Round Two started, the results of the Round One survey were shared with the participants to help them reassess these challenges.

Second, the primary purpose of Round Two was to rate and rank the importance of 60 identified DevSecOps practices through AHP pairwise comparisons, to answer *RQ3: "How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?"*

Third, some additional DevSecOps practices were expected to be collected from participants, to address *Sub-question 3.1: "What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?"* In the following Round Three, these new practices would be compared with the initially identified ones. Based on the Round Three results, some of the lowest-ranked original practices might be replaced. Hence, it helped to improve the DevSecOps CPTM Model (Version 1.0).

Forth, to address *Sub-question 3.2: "Do experts have dissents on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*, the participants were divided in four groups, namely, Group Academic Managerial (AM), Group Industrial Managerial (IM), Group Academic Technical (AT), and Group Industrial Technical (IT). All these groups answered the same questionnaire, enabling an investigation into whether participants held

dissenting opinions due to differences in role and organisational level. A dissent analysis was conducted accordingly for this survey round.

Finally, participants' opinions on the difference between local and global DevSecOps practices were collected, to address *RQ4: "What are the experts' opinions on DevSecOps in GSE contexts?"* and its two related sub-questions: *"Sub-question 4.1: How is DevSecOps different between local and global settings?"* and *"Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?"*

6.1.2 Questionnaire Contents in Round Two

To achieve the above survey objectives and address the corresponding research questions, the Round Two survey questionnaire consisted of 26 questions, employing multiple formats: AHP pairwise comparisons (Questions 1 – 23), multiple-choice (Question 25), and open-ended questions (Questions 24 and 26).

Table 6. 1 lists these questions, including objectives, format types, and descriptions. Due to the repetition of evaluating challenges and the large volume of identified practices to be evaluated in Round Two, the minimum number of comparisons was used to assess importance rather than completing all pairwise comparisons. Once participants had completed these comparisons, the researcher calculated the remaining comparison judgments. Minimising the number of AHP comparisons eliminated redundant questions, reduced unnecessary workloads and survey durations for participants, and mitigated inconsistencies caused by redundant pairwise comparisons (Mu & Pereyra-Rojas, 2017). A sample of the Round Two survey is provided in Appendix B.9, "Sample of Delphi Survey – Round Two" (Page 334), or it can be assessed at the Qualtrics platform (https://aut.au1.qualtrics.com/jfe/form/SV_9B6KY9x4IVJIsmO).

Table 6. 1 - List of questions in Round Two

Question ID	Objective	Format	Description
Q1	To rate the importance of the four categories of DevSecOps challenges	6 AHP pairwise comparisons	Business : OPC Business : PC Business : Technology OPC : PC OPC : Technology PC : Technology

Q2	To rate the importance of 9 initially identified DevSecOps challenges and a newly collected challenge in the OPC (Organisation, People & Culture) category	9 AHP pairwise comparisons	C01 : C02; C02 : C03; C03 : C04; C04 : C05; C05 : C06; C06 : C07; C07 : C08; C08 : C09; C09 : NC01
Q3	To rate the importance of 8 initially identified DevSecOps challenges and two newly collected challenges in the PC (Process Capabilities) category	9 AHP pairwise comparisons	C10 : C11; C11 : C12; C12 : C13; C13 : C14; C14 : C15; C15 : C16; C16 : C17; C17 : NC02; NC02 : NC03
Q4	To rate the importance of the seven identified initially, DevSecOps challenges, and two newly collected challenges in the Technology category	8 AHP pairwise comparisons	C18 : C19; C19 : C20; C20 : C21; C21 : C22; C22 : C23; C23 : C24; C24 : NC04; NC04 : NC05
Q5	To rate the importance of the four originally identified DevSecOps challenges and two newly collected challenges in the Business category	5 AHP pairwise comparisons	C25 : C26; C26 : C27; C27 : C28; C28 : NC06; NC06 : NC07
Q6	To rate the importance of the four categories of DevSecOps practices	6 AHP pairwise comparisons	Business : OPC Business : PC Business : Technology OPC : PC OPC : Technology PC : Technology
Q7	To rate the importance of 15 DevSecOps practices in the OPC (Organisation, People & Culture) category	7 AHP pairwise comparisons	P01 : P02; P02 : P03; P03 : P04; P04 : P05; P05 : P06; P06 : P07; P07 : P08
Q8		7 AHP pairwise comparisons	P08 : P09; P09 : P10; P10 : P11; P11 : P12; P12 : P13; P13 : P14; P14 : P15
Q9	To rate the importance of the three sub-practices of P01	2 AHP pairwise comparisons	SP01.1 : SP01.2; SP01.2 : SP01.3
Q10	To rate the importance of the five sub-practices of P02	4 AHP pairwise comparisons	SP02.1 : SP02.2; SP02.2 : SP02.3; SP02.3 : SP02.4; SP02.4 : SP02.5
Q11	To rate the importance of 17 DevSecOps practices in the PC (Process Capabilities) category	8 AHP pairwise comparisons	P16 : P17; P17 : P18; P18 : P19; P19 : P20; P20 : P21; P21 : P22; P22 : P23; P23 : P24
Q12		8 AHP pairwise comparisons	P24 : P25; P25 : P26; P26 : P27; P27 : P28; P28 : P29; P29 : P30; P30 : P31; P31 : P32
Q13	To rate the importance of the two sub-practices of P20	1 AHP pairwise comparison	SP20.1 : SP20.2
Q14	To rate the importance of the three sub-practices of P21	2 AHP pairwise comparisons	SP21.1 : SP21.2; SP21.2 : SP21.3
Q15	To rate the importance of the four sub-practices of P29	3 AHP pairwise comparisons	SP29.1 : SP29.2; SP29.2 : SP29.3; SP29.3 : SP29.4
Q16	To rate the importance of 23 DevSecOps practices in the Technology category	8 AHP pairwise comparisons	P33 : P34; P34 : P35; P35 : P36; P36 : P37; P37 : P38; P38 : P39; P39 : P40; P40 : P41

Q17		7 AHP pairwise comparisons	P41 : P42; P42 : P43; P43 : P44; P44 : P45; P45 : P46; P46 : P47; P47 : P48
Q18		7 AHP pairwise comparisons	P48 : P49; P49 : P50; P50 : P51; P51 : P52; P52 : P53; P53 : P54; P54 : P55
Q19	To rate the importance of the five sub-practices of P33	4 AHP pairwise comparisons	SP33.1 : SP33.2; SP33.2 : SP33.3; SP33.3 : SP33.4; SP33.4 : SP33.5
Q20	To rate the importance of the three sub-practices of P37	2 AHP pairwise comparisons	SP37.1 : SP37.2; SP37.2 : SP37.3
Q21	To rate the importance of the two sub-practices of P39	1 AHP pairwise comparison	SP39.1 : SP39.2
Q22	To rate the importance of the four sub-practices of P40	3 AHP pairwise comparisons	SP40.1 : SP40.2; SP40.2 : SP40.3; SP40.3 : SP40.4
Q23	To rate the importance of five DevSecOps practices in the Business category	4 AHP pairwise comparisons	P56 : P57; P57 : P58; P58 : P59; P59 : P60
Q24	To collect additional DevSecOps practices	1 open-ended question	Add more practices
Q25	To investigate how DevSecOps practices differ in local and global settings	1 multiple-choice question	4 options: Extremely different; Slightly different; Not different; Uncertain
Q26	To collect differences between local and global DevSecOps practices	1 open-ended question	List the differences if opted for “Extremely/Slightly Different” in Q25

6.1.3 Participants in Round Two

In total, 14 participants have completed the Round Two Delphi survey. Compared to Round One (**Table 5. 2** in Section 5.1.3, Chapter 5, Page 112), four of the 18 participants skipped or dropped out of Round Two. While the Delphi survey ought to be anonymous, **Table 6. 2** provides brief information about the participants in the Round Two survey.

Table 6. 2 - List of participants in Round Two

Group	Part. ID	Occupation / Role	Academic	Industry			Country
				Portfolio	Program	Team	
AT	Pa1	Researcher	1				Belgium
	Pa2	Assistant Professor	1				Canada
	Pa3	Senior Lecturer	1				NZ
AM	Pa5	Professor	1				Ireland
	Pa8	Researcher / Technical Manager / Senior Security Architect	1	1	1		Germany
IT	Pa10	Principal Consultant			1	1	USA
	Pa11	Principal Software Engineer		1	1	1	Germany
	Pa12	Senior DevOps Engineer			1	1	China
	Pa13	DevOps Engineer			1	1	USA

IM	Pa14	Head Marketing and Communications		1	1		India
	Pa15	General Manager (Operations side)		1	1	1	NZ
	Pa16	Senior Security Consultant		1	1		NZ
	Pa17	Senior Agile Coach		1	1		NZ, China
	Pa18	Consultant (Adviser) / Researcher	1	1	1		NZ

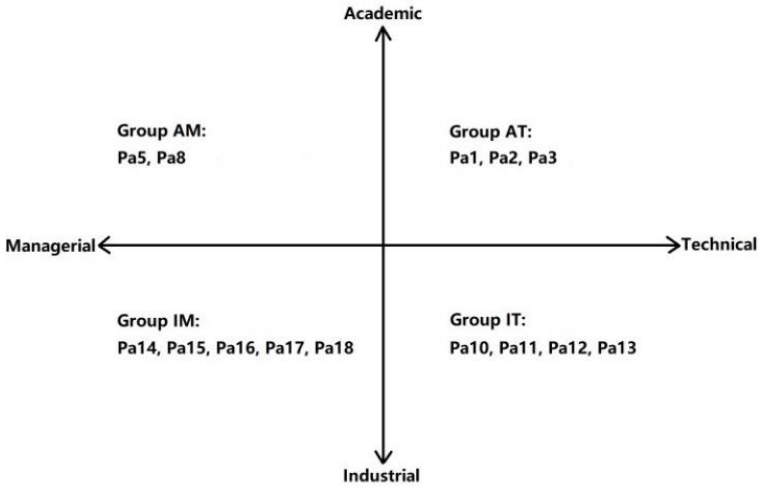
To ensure the representativeness of participants and the reliability of the survey, participants were recruited from around the world and distributed across New Zealand, Europe, Asia, and North America. These participants have diverse roles and work in different fields. They could be academic, industrial, managerial, technical, or have multiple identities.

- Participants Pa1, Pa2, Pa3, Pa5, and Pa8 are academic experts.
- Pa10 – Pa18 are industrial experts.
- Pa5, Pa8, and Pa14 – Pa18 are managerial experts.
- Pa1 – Pa3 and Pa10 – 13 are technical experts.

According to the four-quadrant division for experts grouping, which has been presented in Section 4.5.2.3, Chapter 4 (Page 89), the participants were divided into four groups, as shown in **Figure 6. 1**:

- Participants Pa1, Pa2, and Pa3 are in the Group Academic Technical (AT).
- Pa5 and Pa8 are in the Group Academic Managerial (AM).
- Pa10, Pa11, Pa12, Pa13 are in the Group Industrial Technical (IT).
- Pa14, Pa15, Pa16, Pa17, and Pa18 are in the Group Industrial Managerial (IM).

Figure 6. 1 - Four-quadrant division for participants grouping in Round Two



Regardless of the group placement, all participants were to answer the same questionnaire. Industrial participants were from large-scale or globally distributed organisations, and were further differentiated into three organisational levels: Portfolio level, Program level, and Team level (Beecham et al., 2021). By doing so, it enabled an investigation into whether participants held dissenting opinions due to differences in role and organisational level.

Compared to Round One (**Figure 5. 1** in Section 5.1.3, Chapter 5, Page 113), four academic participants (Pa4, Pa6, Pa7, and Pa9) did not complete the Round Two survey on schedule. Pa4 was in Group AT, and Pa6, Pa7, and Pa9 were in Group AM. Reasons for their withdrawals might include heavy workloads and losing touch due to job-hopping or the termination of their email address. The absence of the four academic participants in Round Two disrupted the balance between academic and industrial participants (5 versus 9). In contrast, the number of managerial and industrial participants remained balanced (7 versus 7).

6.2 RQ3 – Evaluation of Revised DevSecOps Challenges

This section presents the results of the first five AHP pairwise comparisons (Questions 1 – 5, refer to **Table 6. 1**) that were conducted to reassess the importance of 28 initially identified DevSecOps challenges plus 7 newly identified challenges from Round One, within their categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology. The priorities

and rankings of these 35 challenges were calculated, discussed, and reported by comparing them with the results of Round One.

All consistency ratios were within the acceptable range (between 0 and 0.1) (Saaty, 1982). The degree of dissent among participants, i.e., the value of Coefficient of Variation (CV), was within the acceptable range (between 0 and 0.5) (Dajani, Sincoff, & Talley, 1979) and lower than that in Round One. The results also demonstrated a relative stability in the rankings across both rounds. Hence, the evaluation of the challenges dimension of DevSecOps has been conducted thoroughly across the first two iterations of the Delphi survey.

6.2.1 AHP Results of Revised DevSecOps Challenges

In Round Two, due to the repetition of evaluating challenges, the minimum number of comparisons was used to assess importance rather than completing all AHP pairwise comparisons. By doing so, redundant questions were removed, thereby reducing the survey duration and inconsistencies between AHP comparisons (Mu & Pereyra-Rojas, 2017). Once the participants completed a minimum number of comparisons, the remaining comparison judgments were calculated by the researcher.

Refer to **Table 6. 1**, Questions 1 – 5 were employed to conduct AHP pairwise comparisons for 35 revised DevSecOps challenges and their categories. The data and responses were collected by using the Qualtrics online survey platform (Qualtrics, 2024). The original response sheets contain identifiable participant information, such as IP addresses; therefore, they cannot be publicly disclosed due to confidentiality constraints. The processed dataset of responses is available at zenodo.org (https://doi.org/10.5281/zenodo.16932278).

A total of 14 participants were involved in Round Two. The AHP method was employed to build compromise and consensus among participants, so the data were analysed at the group level rather than the individual level. The extracted data were averaged to evaluate the AHP pairwise comparisons and obtain a set of group judgements. The Consistency Ratio (CR) was calculated to assess the consistency of the AHP comparisons. The Coefficient of Variation (CV) was calculated to assess the degree of consensus or dissent among participants. Section 4.5.3 in Chapter 4 (Page 103) has introduced the data analysis process in detail.

AHP comparison matrices:

Table 6. 3 presents a standard 9-point AHP Scale, which was used to quantify priorities (Saaty, 2013).

Table 6. 3 - 9-point AHP comparison scale

Scale	Numerical score	Reciprocal
Equally important	1	1
Moderately more important	3	1/3
Strongly more important	5	1/5
Very strongly more important	7	1/7
Extremely more important	9	1/9
Intermediate values	2, 4, 6, 8	1/2, 1/4, 1/6, 1/8

Table 6. 4 provides the AHP comparison matrix for the four categories: Organisation, People and Culture (OPC), Process Capabilities (PC), Technology, and Business.

Table 6. 4 - AHP comparison matrix for categories of revised challenges

Categories of Challenges	Business	OPC	PC	Technology
Business	1/1	1/4	1/1	2/1
OPC	4/1	1/1	4/1	5/1
PC	1/1	1/4	1/1	4/1
Technology	1/2	1/5	1/4	1/1

Table 6. 5, **Table 6. 6**, **Table 6. 7**, and **Table 6. 8** provide four AHP comparison matrices for 35 revised DevSecOps challenges within each category, by using the minimum number of comparisons (Mu & Pereyra-Rojas, 2017). For example, in **Table 6. 5**, the minimum number of comparisons is comprised of only the comparison judgments in the diagonal (shaded) above the unit diagonal (filled with 1s) of the comparison matrix, e.g., $\frac{C01}{C02}$, $\frac{C02}{C03}$, $\frac{C03}{C04}$, $\frac{C04}{C05}$, $\frac{C05}{C06}$, etc. After the participants completed these comparisons, the researcher calculated the remaining comparison judgments in the upper and lower parts of the matrix. As shown in **Table 6. 5**, we knew the values $\frac{C01}{C02} = \frac{3}{1}$ and $\frac{C02}{C03} = \frac{1}{2}$, then we calculated the value $\frac{C01}{C03} = \frac{C01}{C02} * \frac{C02}{C03} = \frac{3}{1} * \frac{1}{2} = \frac{3}{2}$, and the value of $\frac{C03}{C01}$ was equal to reciprocal of the value $\frac{C01}{C03} = \frac{2}{3}$. The exact process was repeated for the remaining cells in the matrix using Excel spreadsheets.

Table 6. 5 - AHP comparison matrix for revised challenges in "Organisation, People & Culture" category

Challenges in Organisation, People and Culture	C01	C02	C03	C04	C05	C06	C07	C08	C09	NC01
C01	1/1	3/1	3/2	3/2	3/1	9/1	3/1	1/1	4/1	4/3
C02	1/3	1/1	1/2	1/2	1/1	3/1	1/1	1/3	4/3	4/9
C03	2/3	2/1	1/1	1/1	2/1	6/1	2/1	2/3	8/3	8/9
C04	2/3	2/1	1/1	1/1	2/1	6/1	2/1	2/3	8/3	8/9
C05	1/3	1/1	1/2	1/2	1/1	3/1	1/1	1/3	4/3	4/9
C06	1/9	1/3	1/6	1/6	1/3	1/1	1/3	1/9	4/9	1/7
C07	1/3	1/1	1/2	1/2	1/1	3/1	1/1	1/3	4/3	4/9
C08	1/1	3/1	3/2	3/2	3/1	9/1	3/1	1/1	4/1	4/3
C09	1/4	3/4	3/8	3/8	3/4	9/4	3/4	1/4	1/1	1/3
NC01	3/4	9/4	9/8	9/8	9/4	27/4	9/4	3/4	3/1	1/1

Table 6. 6 - AHP comparison matrix for revised challenges in "Process Capabilities" category

Challenges in Process Capabilities	C10	C11	C12	C13	C14	C15	C16	C17	NC02	NC03
C10	1/1	2/1	2/3	2/3	2/1	1/2	3/2	3/4	3/8	3/4
C11	1/2	1/1	1/3	1/3	1/1	1/4	3/4	3/8	1/5	3/8
C12	3/2	3/1	1/1	1/1	3/1	3/4	9/4	9/8	4/7	9/8
C13	3/2	3/1	1/1	1/1	3/1	3/4	9/4	9/8	4/7	9/8
C14	1/2	1/1	1/3	1/3	1/1	1/4	3/4	3/8	1/5	3/8
C15	2/1	4/1	4/3	4/3	4/1	1/1	3/1	3/2	3/4	3/2
C16	2/3	4/3	4/9	4/9	4/3	1/3	1/1	1/2	1/4	1/2
C17	4/3	8/3	8/9	8/9	8/3	2/3	2/1	1/1	1/2	1/1
NC02	8/3	16/3	16/9	16/9	16/3	4/3	4/1	2/1	1/1	2/1
NC03	4/3	8/3	8/9	8/9	8/3	2/3	2/1	1/1	1/2	1/1

Table 6. 7 - AHP comparison matrix for revised challenges in "Technology" category

Challenges in Technology	C18	C19	C20	C21	C22	C23	C24	NC04	NC05
C18	1/1	1/1	1/4	1/1	2/1	1/2	1/1	1/2	1/4
C19	1/1	1/1	1/4	1/1	2/1	1/2	1/1	1/2	1/4
C20	4/1	4/1	1/1	4/1	8/1	2/1	4/1	2/1	1/1
C21	1/1	1/1	1/4	1/1	2/1	1/2	1/1	1/2	1/4
C22	1/2	1/2	1/8	1/2	1/1	1/4	1/2	1/4	1/8
C23	2/1	2/1	1/2	2/1	4/1	1/1	2/1	1/1	1/2
C24	1/1	1/1	1/4	1/1	2/1	1/2	1/1	1/2	1/4
NC04	2/1	2/1	1/2	2/1	4/1	1/1	2/1	1/1	1/2
NC05	4/1	4/1	1/1	4/1	8/1	2/1	4/1	2/1	1/1

Table 6. 8 - AHP comparison matrix for revised challenges in "Business" category

Challenges in Business	C25	C26	C27	C28	NC06	NC07
C25	1/1	1/4	3/4	3/4	1/5	0/1
C26	4/1	1/1	3/1	3/1	3/4	1/7
C27	4/3	1/3	1/1	1/1	1/4	0/1
C28	4/3	1/3	1/1	1/1	1/4	0/1
NC06	16/3	4/3	4/1	4/1	1/1	1/5
NC07	80/3	20/3	20/1	20/1	5/1	1/1

Priorities, rankings, and consistency ratios:

In Round Two, the AHP tool SuperDecisions (SuperDecisions, 2023) was used again to derive the priorities/weights and the consistency ratios. The calculation method and equations have been presented and discussed in Chapters 4 and 5, so further elaboration is not provided in this chapter.

Table 6. 9 reports the AHP results for the four categories of the revised DevSecOps challenges, and **Table 6. 10**, **Table 6. 11**, **Table 6. 12**, and **Table 6. 13** report the AHP results for the challenges within each category. In these tables:

- "Normalised Value" represents the priority/weight within its own matrix/category (SuperDecisions, 2023). It may easily be confused with another AHP terminology, "Local priority/weight", which is explicitly used for alternatives, rather than criteria and sub-criteria (Mu & Pereyra-Rojas, 2017). Here, categories are criteria, and challenges are sub-criteria.
- "Idealised Value" represents the relative proportion compared with the highest-priority object, which has an idealised value of "1" by default (SuperDecisions, 2023).
- "Ranking" refers to the ranking of each object within its own matrix/category, ordered by normalised/idealised values from high to low (SuperDecisions, 2023).
- "Consistency Ratio" represents the consistency among all pairwise comparisons in a matrix; its acceptable value should be between 0 and 0.1, and a lower value means more consistency (Saaty, 1982). In Round Two, for the assessment of challenges, all consistency ratios were below 0.1, indicating that the AHP results were consistent (Mu & Pereyra-Rojas, 2017).

In addition to reporting the AHP results of Round Two, these tables also present comparisons with the AHP results of Round One, to show how participants adjusted their ratings and rankings after

reviewing the Round One results.

As shown in **Table 6. 9**, in Round Two, the most important category of DevSecOps challenges was "Organisation, People and Culture" with a normalised value of approximately 0.57, significantly higher than the other three categories. (All results were rounded to two decimal places for reporting.) The "Process Capabilities" category ranked second with a priority of 0.19, followed by "Business" with a priority of 0.16. The least important category was "Technology", with the lowest priority of 0.08.

Table 6. 9 - AHP results for categories of challenges

Categories of Challenges	Normalised Value	Idealised Value	Ranking
Round 2 (consistency ratio = 0.04910)			
Business	0.15692862919558948	0.27447894721232152	3
Organisation, People and Culture	0.57173284431974403	1.0	1 (highest)
Process Capabilities	0.19424974417066523	0.3397561397785121	2
Technology	0.077088782314001195	0.13483357319750019	4
Round 1 (consistency ratio = 0.04663)			
Business	0.17293954718544899	0.31545340844252778	3
Organisation, People and Culture	0.54822532442839877	1.0	1 (highest)
Process Capabilities	0.20006106203563501	0.36492488238157644	2
Technology	0.078774066350517175	0.14368921470866036	4

Compared with the results of Round One, the rankings remained the same, and the priority values were also very close. Thus, it can be concluded that the assessment of the importance of the challenge categories has yielded extremely stable results after two iterations of the Delphi survey.

This result has also validated the findings from Round One, revealing the different focuses of DevSecOps challenges in reality and in the literature. The MLR findings presented in Chapter 3 state that a majority of the identified challenges are technology-related. However, the results of both survey rounds show that the "Technology" category is the least important. This difference implies that technological challenges are the easiest to identify and overcome among these DevSecOps challenges.

Table 6. 10 lists the priorities and rankings of ten DevSecOps challenges in the "Organisation, People & Culture" category. In Round Two, the top three important OPC-related challenges respectively were: "C08 – Challenges of governance and leadership" (ranked 6th in Round One);

"C01 – Cultural resistance and organisational opposition" (ranked 7th in Round One); and a newly collected challenge from Round One survey "NC01 – Product team does not follow DevSecOps practices".

Table 6. 10 - AHP results for challenges in "Organisation, People and Culture" category

Challenges in Organisation, People and Culture	Normalised Value	Idealised Value	Ranking
Round 2 (consistency ratio = 0.00587)			
C01-Cultural resistance and organisational opposition	0.18874103796162131	0.9999996211245844	2
C02-Challenges of collaboration, communication and coordination	0.059706304867227243	0.3163396942752687	7
C03-Neglecting security	0.11796944136295701	0.62503310324712658	4
C04-Lack of security awareness and responsibility	0.11796944136295701	0.62503310324712658	5
C05-Lack of security knowledge and skills, need for training	0.059706304867227243	0.3163396942752687	8
C06-Recruiting challenges	0.020374236230368398	0.10794805799051548	10
C07-Inconsistent security polices design	0.059706310931741217	0.31633972640665753	6
C08-Challenges of governance and leadership	0.1887411094709876	1.0	1
C09-Lacking confidence	0.048368220421175973	0.25626754317988637	9
NC01-Product team does not follow DevSecOps practices	0.13871759252373711	0.73496226080550897	3
Round 1 (consistency ratio = 0.04642)			
C01-Cultural resistance and organisational opposition	0.088579069370106789	0.38565563449669021	7
C02-Challenges of collaboration, communication and coordination	0.064891977073468152	0.2825267500547784	8
C03-Neglecting security	0.17281650611212448	0.75240866482465463	2
C04-Lack of security awareness and responsibility	0.22968436461639974	1.0	1
C05-Lack of security knowledge and skills, need for training	0.11095377365292257	0.48307064278506112	4
C06-Recruiting challenges	0.11416175567088627	0.49703755787447701	3
C07-Inconsistent security polices design	0.10067299594143227	0.43831018323588622	5
C08-Challenges of governance and leadership	0.092745510629789349	0.40379549032292816	6
C09-Lacking confidence	0.025494046932870273	0.11099600521545457	9

Compared with the results of Round One, the primary reason for this difference in ranking could be the change of participants, namely, the absence of four academic participants in Round Two. In this case, the size of industrial participants was approximately twice that of academic ones (9 versus 5). Thus, this result makes sense, as the top three challenges are more dependent on the corporate governance perspective than on the academic one, especially since NC01 highlights a real-world pain point for the industry.

Table 6. 11 lists the priorities and rankings of ten DevSecOps challenges in the "Process Capabilities" category. A newly identified challenge, "NC02 – Improper or inadequate risk assessment and management" replaced "C17 – Inadequate privileged credentials and access controls causing cyber-attack" as the most important challenge. The second and third places, respectively, were: "C15 – Ignoring processes and security essentials leading to technical and security debt"; and "C12 – Compliance challenges", precisely the same as Round One.

Table 6. 11 - AHP results for challenges in "Process Capabilities" category

Challenges in Process Capabilities	Normalised Value	Idealised Value	Ranking
Round 2 (consistency ratio = 0.00844)			
C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	0.078526695824307186	0.38842745797277767	7
C11-Using unsuitable metrics	0.03852251916098319	0.19054926525733945	10
C12-Compliance challenges	0.11663391531959087	0.57692246904622124	3
C13-Neglecting change control in security	0.11663391531959087	0.57692246904622124	4
C14-Lack of standards	0.038522526849559181	0.19054930328840644	9
C15-Ignoring processes and security essentials leading to technical and security debt	0.15897001182811188	0.78633535946105981	2
C16-Poor visibility of security track record	0.048005567651133674	0.23745657977243151	8
C17-Inadequate privileged credentials and access controls causing cyber attacks	0.10100959090679718	0.49963771192640255	5
NC02-Improper or inadequate risk assessment and management	0.20216566623312865	1.0	1
NC03-Lack of reference model for DevSecOps process	0.10100959090679718	0.49963771192640255	6
Round 1 (consistency ratio = 0.04369)			
C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	0.060700947267667192	0.20616300948926211	6
C11-Using unsuitable metrics	0.04105990078931674	0.13945470535622151	8
C12-Compliance challenges	0.16193158486625681	0.54997993227653874	3
C13-Neglecting change control in security	0.1142938319648401	0.38818439290621332	4
C14-Lack of standards	0.081480788031473683	0.27673908286886018	5
C15-Ignoring processes and security essentials leading to technical and security debt	0.18673224404672531	0.63421220152625357	2
C16-Poor visibility of security track record	0.05936889677495108	0.20163887023392157	7
C17-Inadequate privileged credentials and access controls causing cyber attacks	0.29443180625876908	1.0	1

Table 6. 12 presents the priorities and rankings of nine DevSecOps challenges in the "Technology"

category. The three most important technological challenges were "C20 – Challenges of legacy system refactoring"; "C23 – Availability and reliability of infrastructure, tools, automation, and network bandwidth"; and "NC05 – Code security/code developers working on their dependence/business logic, excluding external tools". Except for NC05, which was newly identified from Round One, C20 and C23 were also ranked as the most important challenges in Round One.

Table 6. 12 - AHP results for challenges in "Technology" category

Challenges in Technology	Normalised Value	Idealised Value	Ranking
Round 2 (consistency ratio = 0.00000)			
C18-Lack of mature tools for automation and security	0.060606060606060608	0.25	5
C19-Complexity in managing different tools	0.060606060606060608	0.25	6
C20-Challenges of legacy system refactoring	0.24242424242424243	1.0	1
C21-Use of cloud and serverless computing brings security complications	0.060606060606060608	0.25	7
C22-Containers and other tools come with their own risks	0.030303030303030304	0.125	9
C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth	0.12121212121212122	0.5	3
C24-Continuous deployment chaos	0.060606060606060608	0.25	8
NC04-Domain of application does matter like cyber physical systems	0.12121212121212122	0.5	4
NC05-Code security/code developers working on their dependence/business logic, excluding external tools	0.24242424242424243	1.0	2
Round 1 (consistency ratio = 0.04943)			
C18-Lack of mature tools for automation and security	0.09759623597928975	0.39257094283837307	5
C19-Complexity in managing different tools	0.12931604763720622	0.52016066230097924	4
C20-Challenges of legacy system refactoring	0.24860789561664393	1.0	1
C21-Use of cloud and serverless computing brings security complications	0.093201141754257213	0.37489212288725687	6
C22-Containers and other tools come with their own risks	0.062882471581701224	0.25293835268477022	7
C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth	0.22047614690792383	0.88684289918088699	2
C24-Continuous deployment chaos	0.14792006052297782	0.59499341384986482	3

Table 6. 13 presents the AHP pairwise results for six DevSecOps business challenges. Two new challenges, "NC07 – Balance between risk, cost, and impact to business" and "NC06 – The ever-stricter security requirements for business that are increasingly challenging are underrepresented", ranked first and second, respectively, surpassing all the initially identified ones. Hence, the initial

ones dropped by two places in the rankings, but the order remained the same as in Round One.

Table 6. 13 - AHP results for challenges in "Business" category

Challenges in Business	Normalised Value	Idealised Value	Ranking
Round 2 (consistency ratio = 0.02562)			
C25-Challenges of cost control	0.042780429121581837	0.07487365124336022	6
C26-Conflicts between security and business	0.13346420137422221	0.23358699929743984	3
C27-Customer readiness for frequent releases	0.045889528960734761	0.080315150120707515	4
C28-Training users for using advanced tools	0.045889528960734761	0.080315150120707515	5
NC06-The ever-stricter security requirements for business that are increasingly challenging are underrepresented	0.16060803434279949	0.28109371965597862	2
NC07-Balance between risk, cost and impact to business	0.57136827723992689	1.0	1
Round 1 (consistency ratio = 0.02660)			
C25-Challenges of cost control	0.1059244470719073	0.23667469807827529	4
C26-Conflicts between security and business	0.44755289826914652	1.0	1
C27-Customer readiness for frequent releases	0.2829011223553482	0.63210655868710053	2
C28-Training users for using advanced tools	0.16362153230359791	0.36559149306458122	3

Overall priorities and rankings:

To derive the overall priorities/weights of the identified challenge, the normalised value of each challenge (in **Table 6. 10**, **Table 6. 11**, **Table 6. 12** and **Table 6. 13**) was multiplied by the normalised value of its corresponding category (in **Table 6. 9**), according to **Equation (5)**: $\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^{T}\mathbf{1} = 1 \end{cases}$ (Page 73).

The overall priorities of the 35 revised DevSecOps challenges are provided in **Table 6. 14**, ordered by ranking, with higher priorities indicating greater importance.

Table 6. 14 - Overall priorities and rankings of revised DevSecOps challenges

Overall Ranking	Overall Priority	DevSecOps Challenge	Challenges Category
1	0.107909491	C08-Challenges of governance and leadership	Organisation, People & Culture
2	0.10790945	C01-Cultural resistance and organisational opposition	Organisation, People & Culture
3	0.089664041	NC07-Balance between risk, cost and impact to business.	Business
4	0.079309404	NC01-Product teams are going rogue and not following DevSecOps practices	Organisation, People & Culture
5	0.067447004	C03-Neglecting security	Organisation, People & Culture
6	0.067447004	C04-Lack of security awareness and responsibility	Organisation, People & Culture

7	0.039270629	NC02-Improper or inadequate risk assessment and management	Process Capabilities
8	0.034136059	C07-Inconsistent security polices design	Organisation, People & Culture
9	0.034136056	C02-Challenges of collaboration, communication and coordination	Organisation, People & Culture
10	0.034136056	C05-Lack of security knowledge and skills, need for training	Organisation, People & Culture
11	0.030879884	C15-Ignoring processes and security essentials leading to technical and security debt	Process Capabilities
12	0.0276537	C09-Lacking confidence	Organisation, People & Culture
13	0.025203999	NC06-Challenges of security/DevSecOps requirements for businesses	Business
14	0.022656108	C12-Compliance challenges	Process Capabilities
15	0.022656108	C13-Neglecting change control in security	Process Capabilities
16	0.020944354	C26-Conflicts between security and business	Business
17	0.019621087	C17-Inadequate privileged credentials and access controls causing cyber attacks	Process Capabilities
18	0.019621087	NC03-Lack of reference model for DevSecOps process	Process Capabilities
19	0.01868819	C20-Challenges of legacy system refactoring	Technology
20	0.01868819	NC05-Code security/code developers working on their dependences/ business logic, excluding external tools	Technology
21	0.015253791	C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	Process Capabilities
22	0.01164862	C06-Recruiting challenges	Organisation, People & Culture
23	0.009344095	C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth	Technology
24	0.009344095	NC04-Domain of application is also matters like cyber-physical systems	Technology
25	0.009325069	C16-Poor visibility of security track record	Process Capabilities
26	0.007482991	C14-Lack of standards	Process Capabilities
27	0.007482989	C11-Using unsuitable metrics	Process Capabilities
28	0.007201381	C27-Customer readiness for frequent releases	Business
29	0.007201381	C28-Training users for using advanced tools	Business
30	0.006713474	C25-Challenges of cost control	Business
31	0.004672047	C18-Lack of mature tools for automation and security	Technology
32	0.004672047	C19-Complexity in managing different tools	Technology
33	0.004672047	C21-Use of cloud and serverless computing brings security complications	Technology
34	0.004672047	C24-Continuous deployment chaos	Technology
35	0.002336024	C22-Containers and other tools come with their own risks	Technology

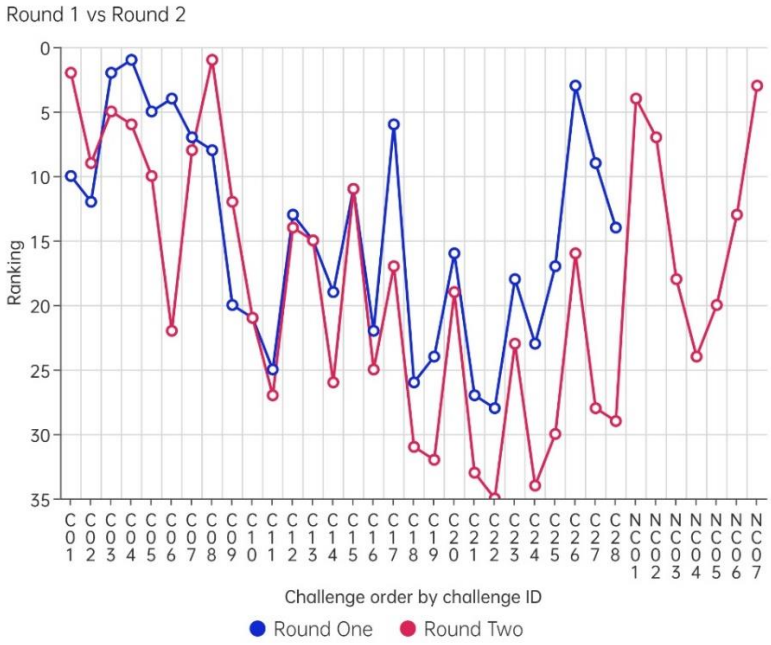
Similar to Round One findings, in Round Two, four of the top five most important challenges were in the “Organisation, People and Culture” category, due to the highest category priority (refer

to **Table 6. 9** on Page 151). One business challenge (NC07), by contrast, ranked third despite not having a high category priority, reflecting its high importance. The lowest-priority "Technology" category resulted in some technological challenges clustered at the bottom of the table.

6.2.2 Termination of the Survey for Challenges Evaluation

Figure 6. 2 depicts the comparison of the overall rankings of DevSecOps challenges between Round One and Round Two. The challenges are ordered by ID on the x-axis, and their rankings are on the y-axis. The blue line denotes the overall ranking in Round One, and the red line denotes the overall ranking in Round Two.

Figure 6. 2 - Comparison of rankings of Challenges between Round One and Round Two



The general trends of both lines are similar, although not identical. This indicates that the results have achieved relative stability in rankings over two iterations of the Delphi survey. In addition to stable rankings in both rounds, no additional DevSecOps challenges were collected from participants in Round Two.

The rankings and weights of the four categories of challenges in Round Two were approximately the same as they were in Round One. In three of the four categories, namely, "Process Capabilities", "Technology", and "Business", the rankings of challenges were extremely close in

both rounds. The significant change in rankings was concentrated in the "Organisation, People & Culture" category (i.e., Challenges C01 – C09). The two possible factors that might have led to this change are the absence of four academic participants and the inclusion of seven newly added challenges in Round Two.

As shown above in **Table 6. 9**, **Table 6. 10**, **Table 6. 11**, **Table 6. 12**, and **Table 6. 13**, all consistency ratios of the AHP comparisons were in the acceptable range (between 0 and 0.1). Hence, a clear conclusion can be drawn: the evaluation of the challenges dimension of DevSecOps has been completed, and it will stop at Round Two, since all stopping criteria have been met. The third round of the Delphi survey only focuses on the dimensions of practices and metrics.

6.3 RQ3 – Evaluation of DevSecOps Practices

This section presents the results of the AHP pairwise comparisons from Questions 6 to 23 (refer to **Table 6. 1** on Page 142), assessing the importance of the 60 identified DevSecOps practices and their associated sub-practices across four categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology. The priorities and rankings of these DevSecOps practices were calculated and reported. In addition to these AHP pairwise-comparison questions, one open-ended question (Question 24) was raised to allow participants to provide additional DevSecOps practices. The data were collected, calculated, analysed, discussed, and reported using multiple tools (e.g., Qualtrics, SuperDecisions, and MS Excel spreadsheets) to address RQ3 and associated sub-questions, specifically regarding the practices dimension of DevSecOps.

However, during the evaluation of DevSecOps practices, AHP results revealed two limitations: one was the high inconsistency in CR scores (between 0.1 and 0.2); the other was the abnormal ranking of practices caused by recency bias among participants, i.e., the rankings of later-ordered practices were significantly higher than those of earlier-ordered practices. These issues were not part of the first half of Round Two, i.e., the evaluation of revised challenges. Potentially having a large number of observations within a single matrix, possibly any more than 15, to be compared, could be the underlying cause of both limitations. Hence, for Round Three, the AHP structure and questionnaire were revised to include an AHP layer of sub-categories between the category and

practices. This revision aimed at reducing the volume of each AHP comparison matrix and mitigating both limitations.

6.3.1 AHP Results of DevSecOps Practices

The minimum number of AHP comparisons was used to assess the importance of 60 identified DevSecOps practices. Once participants had completed the minimum number of comparisons, the researcher calculated the remaining comparison judgments.

As shown in **Table 6. 1** (Page 142), Questions 6 – 23 were used to conduct AHP pairwise comparisons of the identified practices. Response data was collected via the Qualtrics online survey platform (Qualtrics, 2024). The extracted data were averaged to build a group compromise for the evaluation of AHP pairwise comparisons. The Consistency Ratio (CR) was calculated to assess the consistency of the AHP comparisons. The Coefficient of Variation (CV) was calculated to assess the degree of consensus or dissent among participants.

AHP comparison matrices:

Due to the page limit and the large volume of AHP comparison matrices for practices, a screenshot of all matrices is shown in **Figure 6. 3**. The full dataset is available at zenodo.org (https://doi.org/10.5281/zenodo.16932278).

Figure 6. 3 - AHP comparison matrices for practices

Categories of Practices	Business	OPC	PC	Technology
Business	1/1	1/4	1/3	1/1
OPC	4/1	1/1	4/1	5/1
PC	3/1	1/4	1/1	4/1
Technology	1/1	1/5	1/4	1/1

Practices in OPC	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15
P01	1/1	1/1	1/4	1/4	1/8	1/4	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P02	1/1	1/1	1/4	1/4	1/8	1/4	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P03	4/1	4/1	1/1	1/1	1/2	1/1	1/3	1/9	1/9	0/1	0/1	0/1	0/1	0/1	0/1
P04	4/1	4/1	1/1	1/1	1/2	1/1	1/3	1/9	1/9	0/1	0/1	0/1	0/1	0/1	0/1
P05	8/1	8/1	2/1	2/1	1/1	2/1	2/3	2/9	2/9	0/1	1/9	0/1	0/1	0/1	0/1
P06	4/1	4/1	1/1	1/1	1/2	1/1	1/3	1/9	1/9	0/1	0/1	0/1	0/1	0/1	0/1
P07	12/1	12/1	3/1	3/1	3/2	3/1	1/1	1/3	1/3	0/1	1/6	0/1	0/1	0/1	0/1
P08	36/1	36/1	9/1	9/1	9/2	9/1	3/1	1/1	1/1	1/4	1/2	0/1	0/1	0/1	0/1
P09	36/1	36/1	9/1	9/1	9/2	9/1	3/1	1/1	1/1	1/4	1/2	0/1	0/1	0/1	0/1
P10	144/1	144/1	36/1	36/1	18/1	36/1	12/1	4/1	4/1	1/1	2/1	2/5	1/5	0/1	0/1
P11	72/1	72/1	18/1	18/1	9/1	18/1	6/1	2/1	2/1	1/2	1/1	1/5	0/1	0/1	0/1
P12	360/1	360/1	90/1	90/1	45/1	90/1	30/1	10/1	10/1	5/2	5/1	1/1	1/2	1/8	0/1
P13	720/1	720/1	180/1	180/1	90/1	180/1	60/1	20/1	20/1	5/1	10/1	2/1	1/1	1/4	0/1
P14	2880/1	2880/1	720/1	720/1	360/1	720/1	240/1	80/1	80/1	20/1	40/1	8/1	4/1	1/1	1/5
P15	14400/1	14400/1	3600/1	3600/1	1800/1	3600/1	1200/1	400/1	400/1	100/1	200/1	40/1	20/1	5/1	1/1

Practices in PC	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32
P16	1/1	1/2	5/2	5/8	2/5	1/7	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P17	2/1	1/1	5/1	5/3	5/6	2/7	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P18	2/5	1/5	1/1	1/3	1/6	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P19	6/5	3/5	3/1	1/1	1/2	1/6	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P20	12/5	6/5	6/1	2/1	1/1	1/3	1/9	1/9	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P21	36/5	18/5	18/1	6/1	3/1	1/1	1/3	1/3	1/9	1/9	1/9	0/1	0/1	0/1	0/1	0/1	0/1
P22	108/5	54/5	54/1	18/1	9/1	3/1	1/1	1/1	1/3	1/3	1/3	0/1	0/1	0/1	0/1	0/1	0/1
P23	108/5	54/5	54/1	18/1	9/1	3/1	1/1	1/1	1/3	1/3	1/3	0/1	0/1	0/1	0/1	0/1	0/1
P24	324/5	162/5	162/1	54/1	27/1	9/1	3/1	3/1	1/1	1/1	1/1	1/4	0/1	0/1	0/1	0/1	0/1
P25	324/5	162/5	162/1	54/1	27/1	9/1	3/1	3/1	1/1	1/1	1/1	1/4	0/1	0/1	0/1	0/1	0/1
P26	324/5	162/5	162/1	54/1	27/1	9/1	3/1	3/1	1/1	1/1	1/1	1/4	0/1	0/1	0/1	0/1	0/1
P27	1296/5	648/5	648/1	216/1	108/1	36/1	12/1	12/1	4/1	4/1	4/1	1/1	1/5	0/1	1/7	0/1	0/1
P28	1296/1	648/1	3240/1	1080/1	540/1	180/1	60/1	60/1	20/1	20/1	20/1	5/1	1/1	1/3	2/3	2/9	0/1
P29	3888/1	1944/1	9720/1	3240/1	1620/1	540/1	180/1	180/1	60/1	60/1	60/1	15/1	3/1	1/1	2/1	2/3	2/9
P30	1944/1	972/1	4860/1	1620/1	810/1	270/1	90/1	90/1	30/1	30/1	30/1	15/2	3/2	1/2	1/1	1/3	1/9
P31	5832/1	2916/1	14580/1	4860/1	2430/1	810/1	270/1	270/1	90/1	90/1	90/1	45/2	9/2	3/2	3/1	1/1	1/3
P32	17496/1	8748/1	43740/1	14580/1	7290/1	2430/1	810/1	810/1	270/1	270/1	270/1	135/2	27/2	9/2	9/1	3/1	1/1

Practices in Technology	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48	P49	P50	P51	P52	P53	P54	P55
P33	1/1	1/1	1/2	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P34	1/1	1/1	1/2	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P35	2/1	2/1	1/1	1/4	0/1	1/8	0/1	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P36	8/1	8/1	4/1	1/1	1/3	2/3	2/9	4/9	2/9	2/9	1/9	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P37	24/1	24/1	12/1	3/1	1/1	2/1	1/3	4/3	2/3	2/3	1/3	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P38	12/1	12/1	6/1	3/2	1/2	1/1	1/3	2/3	1/3	1/3	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P39	36/1	36/1	18/1	9/2	3/2	3/1	1/1	2/1	1/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P40	18/1	18/1	9/1	9/4	3/2	3/2	1/2	1/1	1/2	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P41	36/1	36/1	18/1	9/2	3/2	3/1	1/1	2/1	1/1	2/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P42	36/1	36/1	18/1	9/2	3/2	3/1	1/1	2/1	1/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P43	72/1	72/1	36/1	9/1	3/1	6/1	2/1	4/1	2/1	2/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P44	144/1	144/1	72/1	18/1	6/1	12/1	4/1	8/1	4/1	4/1	2/1	2/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P45	288/1	288/1	144/1	36/1	12/1	24/1	8/1	16/1	8/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1	0/1
P46	576/1	576/1	288/1	72/1	24/1	48/1	16/1	32/1	16/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1	0/1
P47	1152/1	1152/1	576/1	144/1	48/1	96/1	32/1	64/1	32/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1	0/1	0/1
P48	2304/1	2304/1	1152/1	288/1	96/1	192/1	64/1	128/1	64/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	0/1	0/1	0/1	0/1	0/1
P49	4608/1	4608/1	2304/1	576/1	192/1	384/1	128/1	256/1	128/1	128/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8	0/1	0/1	0/1
P50	9216/1	9216/1	4608/1	1152/1	384/1	768/1	256/1	512/1	256/1	256/1	128/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8	0/1	0/1
P51	18432/1	18432/1	9216/1	2304/1	768/1	1536/1	512/1	1024/1	512/1	512/1	256/1	128/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8	0/1
P52	36864/1	36864/1	18432/1	4608/1	1536/1	3072/1	1024/1	2048/1	1024/1	1024/1	512/1	256/1	128/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4	1/8
P53	73728/1	73728/1	36864/1	9216/1	3072/1	6144/1	2048/1	4096/1	2048/1	2048/1	1024/1	512/1	256/1	128/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2	1/4
P54	147456/1	147456/1	73728/1	18432/1	6144/1	12288/1	4096/1	8192/1	4096/1	4096/1	2048/1	1024/1	512/1	256/1	128/1	64/1	32/1	16/1	8/1	4/1	2/1	1/1	1/2
P55	294912/1	294912/1	147456/1	36864/1	12288/1	24576/1	8192/1	16384/1	8192/1	8192/1	4096/1	2048/1	1024/1	512/1	256/1	128/1	65536/1	32/1	16/1	8/1	4/1	2/1	1/1

Practices in Business	P56	P57	P58	P59	P60
P56	1/1	2/1	2/3	8/3	8/9
P57	1/2	1/1	1/3	4/3	4/9
P58	3/2	3/1	1/1	4/1	4/3
P59	3/8	3/4	1/4	1/1	1/3
P60	9/8	9/4	3/4	3/1	1/1

Priorities, rankings, and consistency ratios:

The AHP software SuperDecisions (SuperDecisions, 2023) was used to derive the priorities/weights and the consistency ratios. **Table 6. 15** presents the AHP results for four categories of DevSecOps practices, and **Table 6. 16**, **Table 6. 17**, **Table 6. 18**, and **Table 6. 19** present the AHP results for practices within each category.

As shown at the bottom of **Table 6. 16**, **Table 6. 17**, and **Table 6. 18**, the values of consistency ratio (CR) were greater than 0.1 but lower than 0.2, indicating that AHP results were not consistent. According to Saaty (1982), the "reasonable" value of CR is 0.2 or lower, and the "acceptable" value is 0.1 or lower. It is suggested that the pairwise comparison process needs to be refined and

repeated if its CR exceeds 0.2. In this study, the "acceptable" value of 0.1 was used to evaluate consistency, with scarce values between 0.1 and 0.2 being tolerated. However, these inconsistencies in Round Two were not incidental; they were mainly due to the large number of practices in those three matrices (over 15 in each).

As shown in **Table 6. 15**, the importance of practices' categories is ranked the same as challenges' categories in **Table 6. 9** (Page 151). The most important category of DevSecOps practices was "Organisation, People and Culture," with a normalised value of approximately 0.57, significantly higher than the other three categories. (All results were rounded to two decimal places for reporting.) The "Process Capabilities" category ranked second with a priority of 0.25, followed by "Business" with a priority of 0.10. The least important category was "Technology", with the lowest priority of 0.09.

Table 6. 15 - AHP results for categories of practices

Categories of Practices	Normalised Value	Idealised Value	Ranking
Business	0.097929712564908927	0.1718938552410755	3
Organisation, People and Culture	0.56971037404196745	1.0	1 (highest)
Process Capabilities	0.24617405097190503	0.4321038587122002	2
Technology	0.086185862421218548	0.15128013521984718	4
consistency ratio = 0.06457			

Table 6. 16 presents the priorities and rankings of 15 DevSecOps practices in the "Organisation, People & Culture" category. The top three most important OPC-related practices were: "P15 – Leadership support"; "P14 – Continuous improvement mindset"; and "P13 – Enhance transparency". However, it was worth noting that the consistency ratio was 0.19629, a comparatively high value, exceeding the "acceptable" value of 0.1 and approaching the "reasonable" value of 0.2.

Table 6. 16 - AHP results for practices in "Organisation, People and Culture" category

Practices in Organisation, People and Culture	Normalised Value	Idealised Value	Ranking
P01-Cultural shift to security	0.006584438860055216	0.024119483378639575	14
P02-Improving collaboration, communication and cooperation	0.006584438860055216	0.024119483378639575	15
P03-Shared and collective responsibility for security	0.01036833062122393	0.037980271880108732	11
P04-Shared knowledge	0.01036833062122393	0.037980271880108732	12

P05-Training, learning and education for security	0.015813993864790689	0.057928301906736611	10
P06-Security champions	0.01036833062122393	0.037980271880108732	13
P07-Recruiting success	0.019901473517872619	0.072901164385743719	9
P08-Continuous feedback loop	0.037172832222201427	0.13616794505596277	7
P09-Be reactive and responsive	0.037172832222201427	0.13616794505596277	8
P10-Shameless retrospectives	0.068277160330296432	0.25010632929067617	5
P11-Impose security policies	0.05086612768315258	0.18632790846193212	6
P12-Commitment and agreement	0.10388191260240399	0.38053023463474922	4
P13-Enhance transparency	0.1372427209646663	0.50273434038975628	3
P14-Continuous improvement mindset	0.21240454409698409	0.77805990453858709	2
P15-Leadership support	0.27299253291164821	1.0	1
consistency ratio = 0.19629 (over 0.1 but below 0.2)			

Table 6. 17 lists the priorities and rankings of 17 DevSecOps practices in the "Process Capabilities" category. The three most important practices in this category were: "P32 – Hybrid life cycles with data-security focus"; "P31 – Common weaknesses enumeration"; and "P29 – Security review and evaluation". In this category, the consistency ratio was 0.18026, higher than the "acceptable" value of 0.1 but lower than the "reasonable" value of 0.2.

Table 6. 17 - AHP results for practices in "Process Capabilities" category

Practices in Process Capabilities	Normalised Value	Idealised Value	Ranking
P16-Shifting security to the left (early)	0.0063535037969557938	0.026360284186661812	16
P17-Security-by-Design	0.0078431103620917719	0.032540567324623515	14
P18-Increase the visibility	0.0052739066536777422	0.02188110412895047	17
P19-Good documentation, logging and reporting	0.0063676988667622928	0.026419178630752348	15
P20-Compliance control	0.0081212004540705627	0.03369434547418064	13
P21-Risk management	0.013539033318288991	0.056172590319971426	12
P22-Vulnerability and incident management	0.022657158622828695	0.094003113753737655	10
P23-Privilege management	0.022657158622828695	0.094003113753737655	11
P24-Configuration management	0.033531348548391939	0.13911943789521772	7
P25-Package management	0.033531348548391939	0.13911943789521772	8
P26-Define metrics	0.032816161196096899	0.13615217094210177	9
P27-Software process maturity	0.061283855295691386	0.25426282776799675	6
P28-Define security requirements	0.10421802853751891	0.4323939888001524	5
P29-Security review and evaluation	0.13961088245665113	0.57923669438448711	3
P30-Keep credentials safe	0.1111898456571431c1	0.46131961573670727	4
P31-Common weaknesses enumeration	0.14998014067528503	0.62225808890683465	2
P32-Hybrid life cycles with data-security focus	0.24102561838732522	1.0	1
consistency ratio = 0.18026 (over 0.1 but below 0.2)			

Table 6. 18 reports the priorities and rankings of 23 DevSecOps challenges in the "Technology" category. The top three technological practices were: "P55 – Big data and behavioural analytic techniques"; "P54 – Integrate security issues within your general bug tracker"; and "P53 – Adopting DevSecOps in microservices-based applications". The consistency ratio was 0.19136, exceeding the "acceptable" value of 0.1 and approaching the "reasonable" value of 0.2.

Table 6. 18 - AHP results for practices in "Technology" category

Practices in Technology	Normalised Value	Idealised Value	Ranking
P33-Automate tools and security processes	0.0037739836021336202	0.021601388626174056	22
P34-Security-as-Code	0.0037739836021336202	0.021601388626174056	23
P35-Threat modelling	0.0041302030008101885	0.023640304127196452	21
P36-Continuous monitoring	0.0066342461029970708	0.037972853997431479	20
P37-Secure coding	0.0093315932941396978	0.05341183070698844	17
P38-Penetration testing	0.0073832716203575616	0.042260098722680862	19
P39-Cloud security	0.009983040143845644	0.057140558240896977	16
P40-Container security	0.0077722697697464507	0.044486632032221828	18
P41-Sensitive information scan	0.010129227035911468	0.057977297400489805	14
P42-Software Composition Analysis	0.010129227035911468	0.057977297400489805	15
P43-Red team security drills	0.013348444419712066	0.076403335537035474	13
P44-Fault injection (chaos engineering)	0.019604334617226476	0.11221056990943244	12
P45-RASP	0.026104136078213172	0.14941389460654289	11
P46-SAST	0.032186332630341714	0.184226947675726	10
P47-DAST	0.036315537941460793	0.20786154126333933	9
P48-IAST	0.044853828601513983	0.25673269551180206	8
P49-Immutable-as-Code	0.057288261266385289	0.32790444416169678	7
P50-Policy-as-Code	0.068646258115174649	0.39291492905274172	6
P51-Design-as-Code	0.083299189602290075	0.47678483971873831	5
P52-Compliance-as-Code	0.10145217531964677	0.58068823214055987	4
P53-Microservices security	0.12234416880140767	0.70026905652990601	3
P54-Integrate security issues within your general bug tracker	0.14680605615171288	0.84028310822978536	2
P55-Big data and behavioural analytic techniques	0.17471023124692758	1.0	1
consistency ratio = 0.19136 (over 0.1 but below 0.2)			

Table 6. 19 presents the AHP pairwise results for the five DevSecOps business practices. The three most important business practices were: "P58 – Business-driven security"; "P60 – Availability and business continuity management"; and "P56 – Consumable security services with APIs". In this category, the consistency ratio was 0.01245 (below 0.1) and considered acceptable.

Table 6. 19 - AHP results for practices in "Business" category

Practices in Business	Normalised Value	Idealised Value	Ranking
P56-Consumable security services with APIs	0.21981904009122546	0.65713966657308664	3
P57-Separation of duties	0.10693413460963942	0.31967504513484141	4
P58-Business-driven security	0.33450885903381594	1.0	1
P59-Linear scalability and affordable cost	0.085836958353192452	0.25660593444706076	5
P60-Availability and business continuity management	0.25290100791212683	0.75603680166378107	2
consistency ratio = 0.01245			

Overall priorities and rankings:

To derive the overall priorities/weights of the identified DevSecOps practices, the normalised value of each practices (in **Table 6. 16**, **Table 6. 17**, **Table 6. 18** and **Table 6. 19**) was multiplied by the normalised value of its corresponding category (in **Table 6. 15**), according to **Equation (5)**: $\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^{T}\mathbf{1} = 1 \end{cases}$ (Page 73).

Table 6. 20 presents the overall priorities of 60 DevSecOps practices, with higher priorities indicating greater importance. These results would not be analysed further due to the two limitations of concern (high inconsistency and recency bias), both of which could compromise the validity of the evaluation of DevSecOps practices. Both limitations are analysed and discussed in detail in Section 6.3.2.

Table 6. 20 - Overall priorities and rankings of DevSecOps practices

Overall Ranking	Overall Priority	DevSecOps Practices	Practices Category
1	0.155526678	P15-Leadership support	Organisation, People & Culture
2	0.121009072	P14-Continuous improvement mindset	Organisation, People & Culture
3	0.078188602	P13-Enhance transparency	Organisation, People & Culture
4	0.059334253	P32-Hybrid life cycles with data-security focus	Process Capabilities
5	0.059182603	P12-Commitment and agreement	Organisation, People & Culture
6	0.038898207	P10-Shameless retrospectives	Organisation, People & Culture
7	0.036921219	P31-Common weaknesses enumeration	Process Capabilities
8	0.034368576	P29-Security review and evaluation	Process Capabilities
9	0.032758356	P58-Business-driven security	Business
10	0.028978961	P11-Impose security policies	Organisation, People & Culture
11	0.027372055	P30-Keep credentials safe	Process Capabilities
12	0.025655774	P28-Define security requirements	Process Capabilities

13	0.024766523	P60-Availability and business continuity management	Business
14	0.021526815	P56-Consumable security services with APIs	Business
15	0.021177748	P08-Continuous feedback loop	Organisation, People & Culture
16	0.021177748	P09-Be reactive and responsive	Organisation, People & Culture
17	0.015086495	P27-Software process maturity	Process Capabilities
18	0.015057552	P55-Big data and behavioural analytic techniques	Technology
19	0.012652607	P54-Integrate security issues within your general bug tracker	Technology
20	0.011338076	P07-Recruiting success	Organisation, People & Culture
21	0.010544338	P53-Microservices security	Technology
22	0.010472029	P57-Separation of duties	Business
23	0.009009396	P05-Training, learning and education for security	Organisation, People & Culture
24	0.008743743	P52-Compliance-as-Code	Technology
25	0.008405989	P59-Linear scalability and affordable cost	Business
26	0.008254548	P24-Configuration management	Process Capabilities
27	0.008254548	P25-Package management	Process Capabilities
28	0.008078487	P26-Define metrics	Process Capabilities
29	0.007179212	P51-Design-as-Code	Technology
30	0.005916337	P50-Policy-as-Code	Technology
31	0.005906946	P03-Shared and collective responsibility for security	Organisation, People & Culture
32	0.005906946	P04-Shared knowledge	Organisation, People & Culture
33	0.005906946	P06-Security champions	Organisation, People & Culture
34	0.005577605	P22-Vulnerability and incident management	Process Capabilities
35	0.005577605	P23-Privilege management	Process Capabilities
36	0.004937438	P49-Immutable-as-Code	Technology
37	0.003865766	P48-IAST	Technology
38	0.003751223	P01-Cultural shift to security	Organisation, People & Culture
39	0.003751223	P02-Improving collaboration, communication and cooperation	Organisation, People & Culture
40	0.003332959	P21-Risk management	Process Capabilities
41	0.003129886	P47-DAST	Technology
42	0.002774007	P46-SAST	Technology
43	0.002249807	P45-RASP	Technology
44	0.001999229	P20-Compliance control	Process Capabilities
45	0.00193077	P17-Security-by-Design	Process Capabilities
46	0.001689616	P44-Fault injection (chaos engineering)	Technology
47	0.001567562	P19-Good documentation, logging and reporting	Process Capabilities
48	0.001564068	P16-Shifting security to the left (early)	Process Capabilities
49	0.001298299	P18-Increase the visibility	Process Capabilities
50	0.001150447	P43-Red team security drills	Technology
51	0.000872996	P41-Sensitive information scan	Technology
52	0.000872996	P42-Software Composition Analysis	Technology
53	0.000860397	P39-Cloud security	Technology
54	0.000804251	P37-Secure coding	Technology
55	0.00066986	P40-Container security	Technology

56	0.000636334	P38-Penetration testing	Technology
57	0.000571778	P36-Continuous monitoring	Technology
58	0.000355965	P35-Threat modelling	Technology
59	0.000325264	P33-Automate tools and security processes	Technology
60	0.000325264	P34-Security-as-Code	Technology

6.3.2 Limitations in Round Two – Inconsistencies and Recency Bias

After completing the data analysis for Round Two, two key limitations affecting the evaluation of DevSecOps practices were identified: high inconsistencies in the AHP pairwise comparisons (with CR values between 0.1 and 0.2) and abnormal practice rankings due to participant's recency bias (i.e., practices presented later in the list were rated significantly higher).

However, neither issue occurred in the first half of Round Two, specifically during the evaluation of revised challenges. It was likely that the comparison of a large number of observations in an AHP comparison matrix was the cause of both issues.

Hence, for the upcoming Round Three, the AHP structure and questionnaire were revised by adding an AHP layer of subcategories between categories and practices, aimed at reducing the volume of each AHP comparison matrix and mitigating those limitations.

Inconsistency:

In the latter half of Round Two, for the evaluation of DevSecOps practices, AHP results revealed high inconsistencies in **Table 6. 16**, **Table 6. 17**, and **Table 6. 18** (Pages 161, 162, and 163), the CR values were 0.19629, 0.18026, and 0.19136, all between 0.1 and 0.2, within the range between "acceptable" value and "reasonable" value.

Saaty (1982) states that the "reasonable" value of CR is lower than 0.2, and the "acceptable" value is lower than 0.1. Saaty (1982) also suggests that the pairwise comparison process should be refined and repeated if its CR exceeds 0.2. In this study, a threshold of 0.1 was adopted as the "acceptable" consistency value, with occasional CR values between 0.1 and 0.2 tolerated.

According to **Equation (6)**: $CR = \frac{CI}{RI}$ (Page 74), the value of the consistency ratio (CR) was obtained by comparing the consistency index (CI) of the matrix versus the consistency index of a random-like matrix (RI). The value of CI was obtained by using **Equation (7)**: $CI = \frac{\lambda_{max} - n}{n-1}$

(Page 74).

The value of RI depends on the matrix size. Saaty (2013) provides RI values for matrices of different sizes in **Table 6. 21**, supporting the recommended maximum matrix size of 15.

Table 6. 21 - RI values for the matrices of different sizes

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49	1.52	1.54	1.56	1.58	1.59

However, as shown in **Table 6. 16**, **Table 6. 17**, and **Table 6. 18** (Pages 161, 162, and 163), the numbers of the practices being compared in those three matrices were 15, 17, and 23, respectively. High inconsistencies in Round Two were not identified in the "Business practices" matrix or in the matrices for challenges, which had smaller sizes (5-10). Hence, it could be speculated that having a large number of observations within a single matrix, possibly any more than 15, to be compared, could be the underlying cause of high inconsistencies in AHP comparisons.

This study employed a combination of the AHP method and Delphi survey, allowing multiple survey rounds for participants and the researcher to refine and repeat the pairwise comparison process if an unacceptable CR occurred.

Recency bias:

In the latter half of Round Two, during the evaluation of DevSecOps practices, another limitation emerged: the recency bias. Many participants displayed recency bias when rating and ranking practices, reacting more heavily to the latter-ordered practices than to the former-ordered ones.

Table 6. 16, **Table 6. 17**, and **Table 6. 18** (on Pages 161, 162, and 163) show that the priorities and rankings of latter-ordered practices were generally higher than those of former-ordered practices, with the most recent being the highest.

Aren (2019) defines recency bias as the tendency to regard newly presented information as more important than earlier information without justification and to assign inconsistent significance to recent items when making decisions. Fudenberg and Levine (2014) state that recency bias can result from limited short-term memory for past observations.

Miller (1956) summarised evidence that people can remember 7 ± 2 observations in short-term

memory tasks. Cowan (2001) updated to 4 ± 1. Gobet and Clarkson (2004) argue that this number depends on presentation media (physical board or computer display) and participants' expertise level (experts or novices). The computer display provides higher readability and easier handling. Experts are more familiar with the observations to be remembered. Their findings support that experts can remember 15 chunks in a short-term memory task using a computer display.

Hence, recency bias could also result from the large volume of observations in an AHP comparison matrix. Considering that the participants in this Delphi-AHP study were experts, it could be speculated that more than 15 observations within a single AHP comparison matrix could lead to recency bias. It is surprisingly coincidental that the upper limits for the AHP matrix size and for the optimal short-term memory are both 15.

Potential solutions:

To address both limitations, the AHP structure for the Round Three survey was revised to include an additional AHP layer of sub-categories between the category and the practices. This revision ensured that each category contained two or three sub-categories, each with its own associated practices.

According to the cybersecurity domain map presented by Jiang (2021):

- Two sub-categories, "Training" and "Governance", were added under the "Organisation, People & Culture" category.
- Two sub-categories, "Security Process" and "Security Management", were added under the "Process Capabilities" category.
- Three sub-categories, "Security Architecture", "Application Security", and "Operation Security", were added under the "Technology" category.

Figure 6. 4 shows the revised AHP structure for DevSecOps practices. The number of observations in each pairwise comparison matrix was reduced to fewer than 10. Consequently, both limitations could be effectively mitigated. The AHP comparison process of DevSecOps practices was repeated in the following survey round (Round Three). To prevent these issues during the evaluation of DevSecOps metrics in Round Three, the AHP structure for DevSecOps metrics was proactively revised, as depicted in **Figure 6. 5**.

Figure 6. 4 - Revised AHP structure for DevSecOps practices

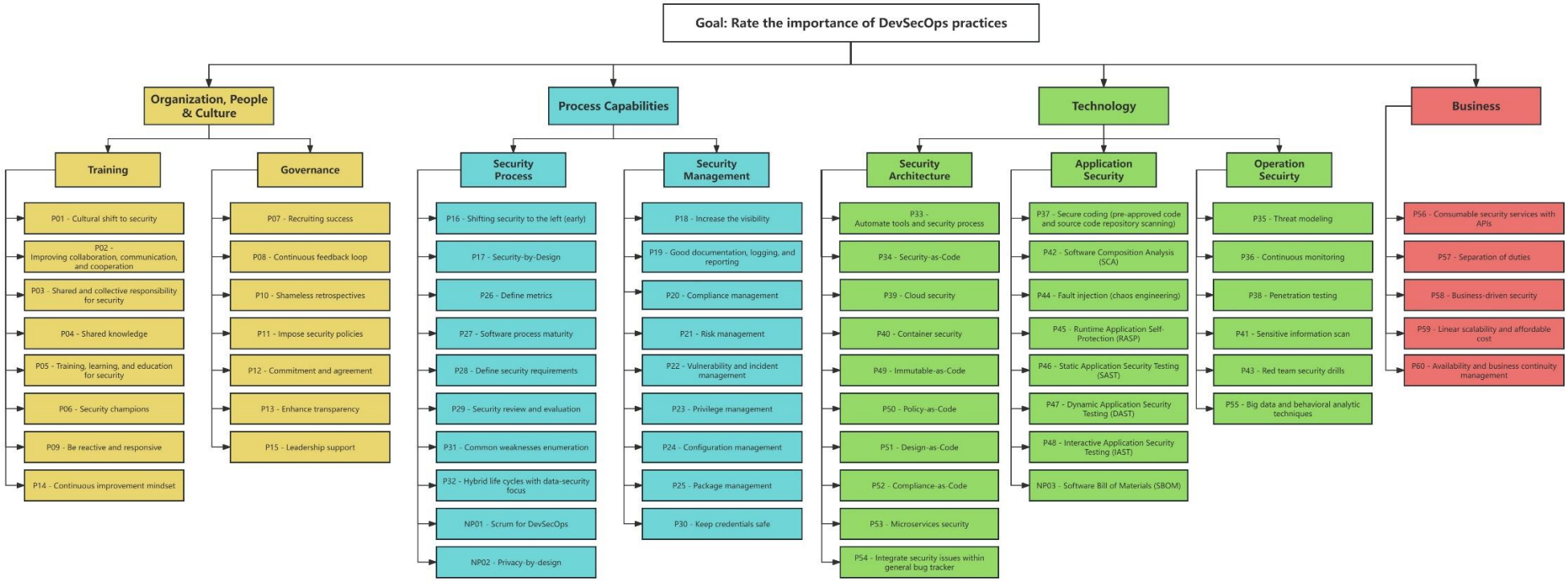
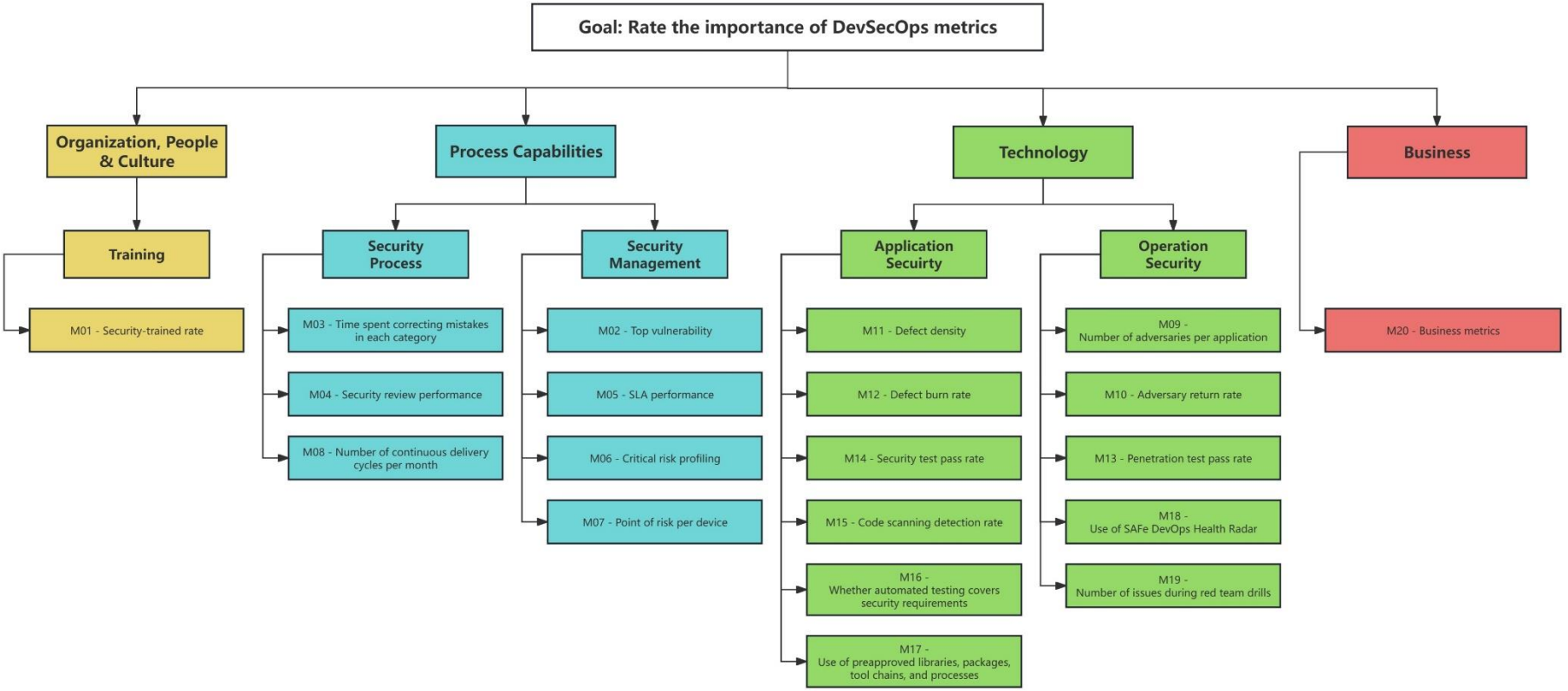


Figure 6. 5 - Revised AHP structure for DevSecOps metrics



Accordingly, the AHP structures for practices and metrics in the SuperDecisions tool were also rebuilt, as shown in **Figure 6. 6** and **Figure 6. 7**.

Figure 6. 6 - Revised AHP structure for DevSecOps practices in SuperDecisions

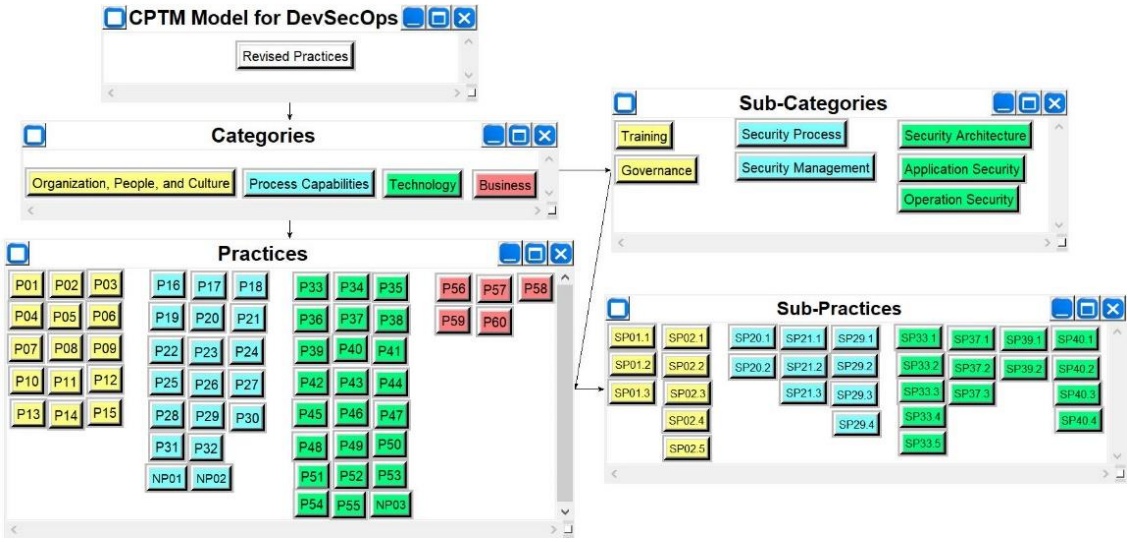
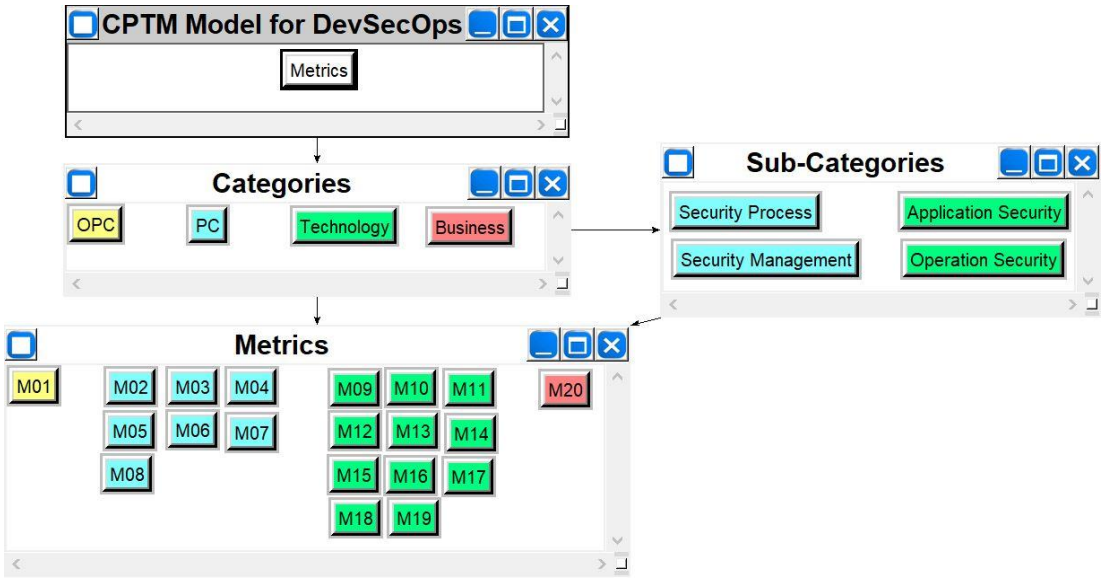


Figure 6. 7 - Revised AHP structure for DevSecOps metrics in SuperDecisions



6.3.3 Open-ended Question Results – Additional Practices

In addition to the AHP pairwise-comparison questions, Round Two included an open-ended question (Question 24) that allowed participants to provide their individual opinions on additional DevSecOps practices. Of the 14 participants, only 6 expressed their viewpoints on this question.

Similar to the data synthesis step in the MLR study, the Thematic Analysis (TA) method was adopted to code and theme the qualitative data, but performed in reverse order. It initially followed a deductive approach, attempting to map the newly identified codes to the existing themes in the DevSecOps CPTM Model (Version 1.0). Subsequently, coding and theming were directed by the content of new data in an inductive approach (Braun & Clarke, 2021).

As a result, three new themes/practices were derived and grouped into different categories, as presented in **Table 6. 22**. In the following Round Three, these emerging practices would be compared with the original ones to replace the least important initial ones. By doing so, the DevSecOps CPTM Model (Version 1.0) would be refined via iterative improvement.

Table 6. 22 - Additional new practices

Comment/Code	Handling	New Theme/Practice	Category
"Scrum for DevSecOps, having frequent but short stand-up meetings to review security issues."	Included	NP01 – Scrum for DevSecOps	Process Capabilities
"Privacy by design, incorporating privacy consideration and data protection."	Included	NP02 – Privacy by design	Process Capabilities
"Software Bill of Materials (SBOM). Given the extensive use of open-source software, it is vital to have a robust, reliable practice for software bill of materials."	Included	NP03 – Software Bill of Materials (SBOM)	Technology
"Have an open-source clearing house service to ensure that teams can ensure only pre-approved code is used."	Covered by "P37 – Secure coding", i.e., "SP37.1 – Source code repository and scanning" and "SP37.2 – Build preapproved code"	N/A	Technology
"Automation of triage to reduce cognitive workload."	Covered by "P31 – Common weaknesses enumeration" and "P33 – Automate tools and security process"	N/A	Technology
"Automation of compliance checks"	Covered by "P33 – Automate tools and security process" and "P52 – Compliance-as-code"	N/A	Technology

6.4 RQ4 – DevSecOps Practices in GSE Contexts

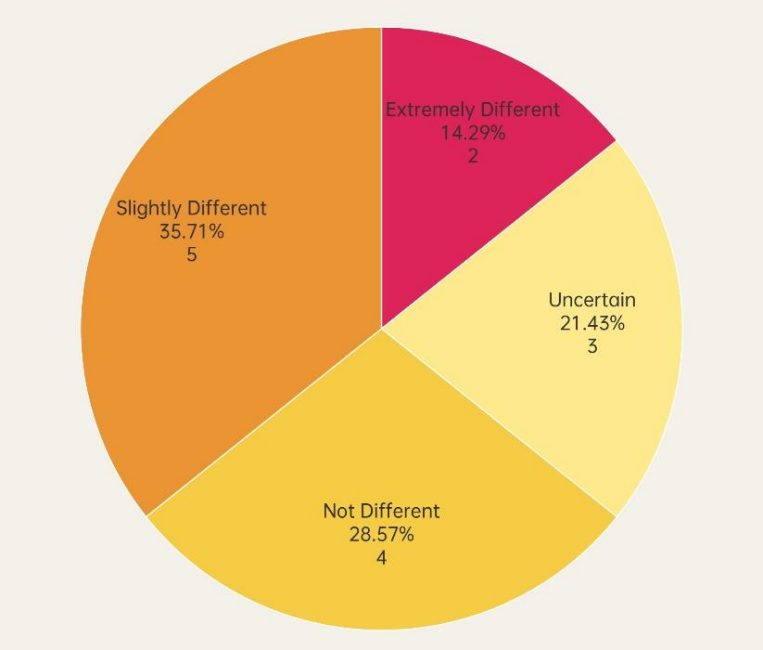
This section presents participants' opinions on the differences between local and global

DevSecOps practices by using a multiple-choice question (Question 25) and one open-ended question (Question 26), to address RQ4 and associated sub-questions.

To address sub-question 4.1, a multiple-choice question (Question 25) was raised to survey participants' opinions on "How do DevSecOps practices differ in local and global settings?", including four options: "Extremely Different", "Slightly Different", "Not Different", and "Uncertain".

Figure 6. 8 presents a pie chart showing the proportions of these options based on participants' responses. 35.71% of participants selected "Slightly Different," which ranked first. "Not Different" ranked second, with 28.57%, followed by "Uncertain" at 21.43%. Two participants (14.29%) opted for "Extremely Different".

Figure 6. 8 - Opinions on "How do DevSecOps practices differ in local and global settings?"



To address sub-question 4.2, Round Two posed an open-ended question (Question 26) to collect participants' opinions on differences between local and global DevSecOps practices; however, no one responded. Hence, most participants believed that DevSecOps practices differ slightly between local and global contexts, and these differences have not been well recognised.

6.5 Dissent Analysis for Round Two

In this section, a dissent analysis is presented for Round Two to examine the degree of consensus or dissent among participants using both quantitative and qualitative methods.

6.5.1 Dissent on AHP Results in Round Two

To address *Sub-question 3.2: "Will the experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*, a dissent analysis was conducted to discuss participants' agreements and disagreements on AHP comparison results, based on three levels:

- Individual level: to analyse the consensus or dissent among each participant regardless of the participant's position and role.
- Groups level: to analyse the consensus or dissent between the four participants' groups (i.e., Group AT, Group AM, Group IT, and Group IM).
- Roles level: to analyse the consensus or dissent between the four participants' roles (i.e., Academic, Industrial, Managerial, and Technical).

Based on the four-quadrant division for participants grouping in **Figure 6. 1** (Page 146), the datasets were merged as follows:

- Two datasets for Groups AM and AT were merged into a single dataset for the Academic role.
- Two datasets for Groups IM and IT were merged into a single dataset for the Industrial role.
- Two datasets for Groups AT and IT were merged into a single dataset for the Technical role.
- Two datasets for Groups AT and IT were merged into a single dataset for the Managerial role.

Based on previous experience in Round One, the AHP method is most suitable for small to medium-sized groups (e.g., 8 to 20 participants), rather than for individuals, tiny groups (fewer than five participants), or relatively large groups (over 30 participants). Using AHP with fewer than five participants may lead to inconsistencies in the comparisons. Using AHP with large groups may lead to overly-compromised comparison results (i.e., converging to the "Equally important" scale), narrowing the ranking gaps and rendering the AHP method less meritorious.

Therefore, as discussed in Section 5.4.1, Chapter 5 (Page 127), since the second round of the Delphi survey, the dissent analysis for AHP results has been conducted only based on participants' roles, rather than by small groups. This is also more satisfactory for the content of *Sub-question 3.2: "Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*

Additionally, coefficients of variations (CVs) were calculated to evaluate the degree of dissent at the individual level and to determine whether participants with similar roles would exhibit less dissent.

Dissent analysis based on individuals:

Coefficients of variations (CVs) of the AHP comparison results were calculated to evaluate the degree of dissent among participants at the individual level, to verify whether participants with similar roles would have more consensual opinions, or if individual variations are the primary factor. CV is defined as the ratio of the standard deviation to the arithmetic mean, and a higher value indicates greater dissimilarity among participants (Everitt, 2006).

In this research, the acceptable value of CV was defined to be less than 0.5 (Dajani, Sincoff, & Talley, 1979). Instead of reporting all 121 CVs individually for Round Two, **Table 6. 23** shows the average value for the overall CVs (i.e., all 14 participants in Round Two) and for the CVs across four groups and four roles. These results were compared and discussed to investigate whether participants in the same group or with the same roles had more consensual results. All the CV values were less than 0.5, i.e., within the acceptable range (Dajani, Sincoff, & Talley, 1979).

Table 6. 23 - Coefficient of variations in Round Two

Overall CVs (average value) for the evaluation of revised challenges	**CVs in 4 Groups (average value)**			
	Group AT: 0.35	Group AM: 0.31	Group IT: 0.32	Group IM: 0.43
0.45	**CVs in 4 Roles (average value)**			
	Technical: 0.38	Managerial: 0.47	Academic: 0.41	Industrial: 0.42
Overall CVs (average value) for the evaluation of practices	**CVs in 4 Groups (average value)**			
	Group AT: 0.31	Group AM: 0.30	Group IT: 0.27	Group IM: 0.22
0.34	**CVs in 4 Roles (average value)**			
	Technical: 0.33	Managerial: 0.29	Academic: 0.35	Industrial: 0.28

The average value of overall CVs for the evaluation of challenges was 0.45, all the four participants' groups had lower values than this value, respectively: Group Academic Technical (AT) – 0.35; Group Academic Managerial (AM) – 0.31; Group Industrial Technical (IT) – 0.32; and Group Industrial Managerial (IM) – 0.43, indicating that the participants in the same groups held more consensual opinions on the prioritisation of DevSecOps challenges.

If CVs were compared based on participants' roles rather than groups, technical experts, academic experts, and industrial experts held more consensual opinions, with CVs of 0.38, 0.41, and 0.42, respectively, which were lower than the overall value of 0.45. In contrast, managerial experts expressed more dissenting opinions, with a CV of 0.47, exceeding the overall value of 0.45. In other words, the degree of dissent among the seven participants with managerial roles (i.e., Pa5, Pa8, Pa14, Pa15, Pa16, Pa17, and Pa18) was slightly higher than that of others when they rated and ranked the importance of DevSecOps challenges in Round Two.

In comparison with the results of CVs in Round One (in **Table 5. 18** in Section 5.4.1, Chapter 5, Page 128), the results of CVs in Round Two presented a decline of different degrees, indicating that the degree of dissent among participants was lower, namely, participants with similar roles had less dissent in this round. Moreover, a similarity between Round One and Round Two was that managerial participants invariably gave more dissenting opinions on the importance of DevSecOps challenges.

A conjecture for this result is that these managerial experts work at the portfolio level, and they may possess unique insights and specialised opinions relevant to their organisations. During surveys, managerial participants were so confident in their judgments that they were more likely to assign higher scores in AHP pairwise comparisons (e.g., opting for "8" or "9," which represents "Extremely more important"). By contrast, other participants judged AHP comparisons more mildly and cautiously (e.g., they opted for "2" or "3", representing "Moderately more important"); hence, agreements were more likely to be obtained from them. Arnold et al. (2000) conducted relevant experiments, and the results indicated that highly experienced practitioners (partners, managers, and directors) exhibited greater levels of dissent and bias due to overconfidence than did moderately experienced and insolvency practitioners (seniors and staff).

The average value of overall CVs for the evaluation of practices was 0.34, all the four participants'

groups had lower values than this value, respectively: Group Academic Technical (AT) – 0.31; Group Academic Managerial (AM) – 0.30; Group Industrial Technical (IT) – 0.27; and Group Industrial Managerial (IM) – 0.22, indicating that the participants in the same groups held more consensual opinions on the prioritisation of DevSecOps practices.

If CVs were compared based on participants' roles rather than groups, industrial experts, managerial experts, and technical experts held more consensual opinions, with CVs of 0.28, 0.29, and 0.33, respectively, which were lower than the overall value of 0.34. The academic experts expressed more dissenting opinions, with a CV of 0.35, slightly exceeding the overall value of 0.34. This result might be caused by the lack of four academic participants in Round Two, resulting in a smaller sample of academic roles that led to a higher degree of dissent. The recency bias, which occurred universally in Round Two, could be another factor affecting the CV values.

Dissent analysis based on roles:

To address *Sub-question 3.2: "Will the experts have dissents on the prioritisation due to their different roles, e.g., academic, industrial, technical, and managerial?"*, the dissent analysis was further conducted by role rather than by group.

Table 6. 24 reports AHP results for the four categories of DevSecOps challenges and practices across four role types: Academic, Industrial, Technical, and Managerial.

Table 6. 24 - Round Two AHP results for categories based on roles

Categories of Challenges	**Priority and Ranking**				
	Overall	**Academic (Group AT+AM)**	**Industrial (Group IT+IM)**	**Managerial (Group AM+IM)**	**Technical (Group AT+IT)**
Organisation, People & Culture	**0.57, 1st**	0.53, 1st	0.58, 1st	0.51, 1st	0.58, 1st
Process Capabilities	**0.19, 2nd**	0.24, 2nd	0.14, 3rd (↓1)	0.13, 3rd (↓1)	0.26, 2nd
Business	**0.16, 3rd**	0.07, 4th (↓1)	0.21, 2nd (↑1)	0.29, 2nd (↑1)	0.09, 3rd
Technology	**0.08, 4th**	0.15, 3rd (↑1)	0.06, 4th	0.07, 4th	0.06, 4th
Categories of Practices	**Priority and Ranking**				
	Overall	**Academic (Group AT+AM)**	**Industrial (Group IT+IM)**	**Managerial (Group AM+IM)**	**Technical (Group AT+IT)**
Organisation, People & Culture	**0.56, 1st**	0.55, 1st	0.57, 1st	0.57, 1st	0.51, 1st
Process Capabilities	**0.24, 2nd**	0.26, 2nd	0.19, 2nd	0.15, 3rd (↓1)	0.30, 2nd
Business	**0.10, 3rd**	0.05, 4th (↓1)	0.17, 3rd	0.22, 2nd (↑1)	0.06, 4th (↓1)
Technology	**0.09, 4th**	0.13, 3rd (↑1)	0.08, 4th	0.06, 4th	0.13, 3rd (↑1)

Table 6. 24 shows that participants across all four roles reached consensus that "Organisation, People and Culture" was the most important category of DevSecOps challenges. For the "Process Capabilities" category, academic and technical roles ranked it second, consistent with the overall result, while industrial and managerial roles ranked it third. For the "Business" category, which was the most controversial, industrial and managerial roles ranked it second, ahead of "Process Capabilities"; technical roles ranked it third, consistent with the overall result; and academic roles rated it as the least important category. This suggests that industrial professionals, particularly those in managerial roles and higher-level positions, tend to prioritise business challenges. For the "Technology" category, academic roles ranked it third, over "Business"; the other roles ranked it as the least important category, consistent with the overall result.

Figure 6. 9 compares the rankings of challenge categories in Round One and Round Two, based on participants' roles. It indicates that most priorities and rankings of categories have stabilised. A noticeable change was that academic participants ranked the "Technology" category over "Business" in Round Two. This might be attributed to the absence of three participants in Group AM (Academic Managerial), resulting in a higher proportion of academic participants who were "technical-role" researchers, focusing more on technological challenges than business ones.

Figure 6. 9 - Dissents on categories of challenges in Round One and Round Two

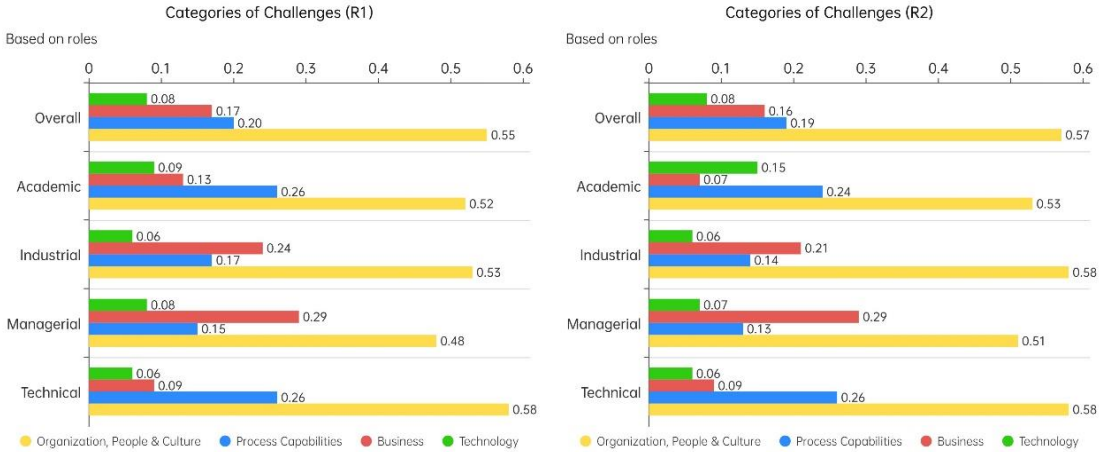


Table 6. 24 also presents the Round Two AHP results for the four practice categories based on participants' roles, which were approximately identical to those for the challenge categories.

Participants across all four roles reached consensus that "Organisation, People, and Culture" is the most important category of practices. For the "Process Capabilities" category, academic, industrial, and technical roles ranked it second, consistent with the overall ranking; whereas managerial roles ranked it third. For the "Business" category, managerial roles ranked it second, ahead of "Process Capabilities"; industrial roles ranked it third, consistent with the overall result; academic and technical roles rated it as the least important category. For the "Technology" category, academic and industrial roles ranked it over "Business" in third place. In contrast, industrial and managerial roles rated it as the least important category, consistent with the overall result. The dissenting rankings were also due to the absence of three participants in Group AM (Academic Managerial), resulting in a higher proportion of technical-oriented researchers among the academic participants, who focused more on technological practices than on business ones.

Table 6. 25 provides the overall priorities and rankings of DevSecOps challenges based on participants' roles. The dissent analysis for the evaluation of DevSecOps practices was skipped in Round Two due to abnormal results, caused by high inconsistencies and the recency bias noted in Section 6.3.2. Hence, it was placed in the next chapter, when the third round of the Delphi survey had been completed.

Table 6. 25 - Round Two overall priorities and rankings of DevSecOps challenges based on roles (order by overall ranking)

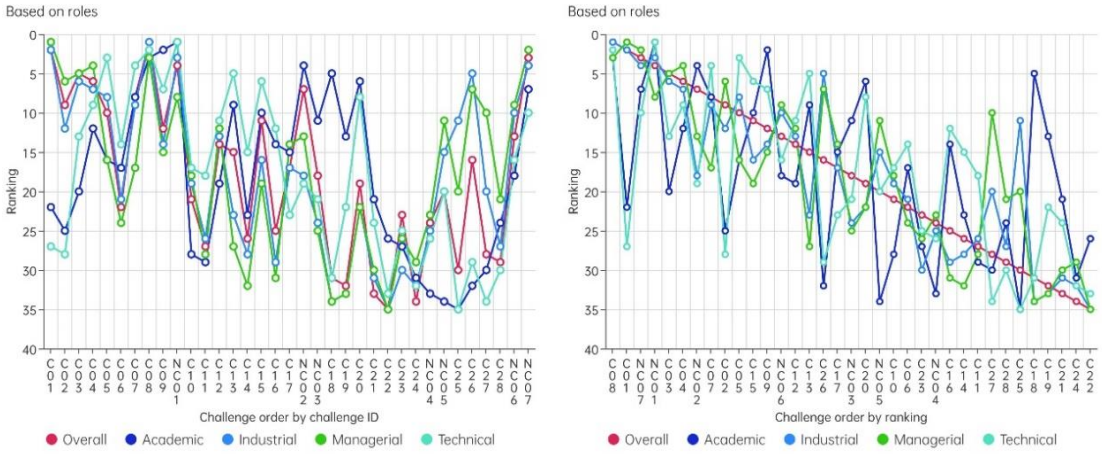
DevSecOps Challenge (Category: OPC, PC, B, T)	**Overall Priority, Ranking within Category, Overall Ranking**				
	Overall	**Academic**	**Industrial**	**Managerial**	**Technical**
C08-Challenges of governance and leadership (OPC)	**0.108, 1, 1**	0.091, 3 (↓2), 3 (↓2)	0.132, 1 (-), 1 (-)	0.056, 2 (↓1), 3 (↓2)	0.148, 2 (↓1), 2 (↓1)
C01-Cultural resistance and organisational opposition (OPC)	**0.108, 2, 2**	0.010, 9 (↓7), 22 (↓20)	0.089, 2 (-), 2 (-)	0.186, 1 (↑1), 1 (↑1)	0.008, 9 (↓7), 27 (↓25)
NC07-Balance between risk, cost and impact to business (B)	**0.090, 1, 3**	0.039, 1 (-), 7 (↓4)	0.076, 1 (-), 4 (↓1)	0.157, 1 (-), 2 (↑1)	0.033, 1 (-), 10 (↓7)
NC01-Product teams are going rogue and not following DevSecOps practices (OPC)	**0.079, 3, 4**	0.182, 1 (↑2), 1 (↑3)	0.089, 3 (↓2), 3 (↑1)	0.040, 6 (↓3), 8 (↓4)	0.168, 1 (↑2)), 1 (↑3)
C03-Neglecting security (OPC)	**0.067, 4, 5**	0.013, 8 (↓4), 20 (↓15)	0.053, 4 (-), 6 (↓1)	0.053, 4 (-), 5 (-)	0.019, 7 (↓3), 13 (↓8)
C04-Lack of security awareness and responsibility (OPC)	**0.067, 5, 6**	0.030, 5 (-), 12 (↓6)	0.053, 5 (-), 7 (↓1)	0.053, 5 (-), 4 (↑2)	0.035, 6 (↓1), 9 (↓3)
NC02-Improper or inadequate risk assessment and management (PC)	**0.039, 1, 7**	0.066, 1 (-), 4 (↑3)	0.020, 4 (↓3), 18 (↓11)	0.020, 3 (↓2), 13 (↓6)	0.016, 8 (↓7), 19 (↓12)

C07-Inconsistent security polices design (OPC)	**0.034, 6, 8**	0.039, 4 (↑2), 8 (-)	0.047, 7 (↓1), 9 (↓1)	0.019, 9 (↓3), 17 (↓9)	0.059, 4 (↑2), 4 (↑4)
C02-Challenges of collaboration, communication and coordination (OPC)	**0.034, 7, 9**	0.007, 10(↓3), 25(↓16)	0.026, 8 (↓1), 12 (↓3)	0.053, 3 (↑4), 6 (↑3)	0.008, 10 (↓3), 28(↓19)
C05-Lack of security knowledge and skills, need for training (OPC)	**0.034, 8, 10**	0.015, 6 (↑2), 16 (↓6)	0.053, 6 (↑2), 8 (↑2)	0.019, 7 (↑1), 16 (↑6)	0.059, 3 (↑5), 3 (↑7)
C15-Ignoring processes and security essentials leading to technical and security debt (PC)	**0.031, 2, 11**	0.037, 3 (↓1), 10 (↑1)	0.020, 2 (-), 16 (↓5)	0.017, 5 (↓3), 19 (↓8)	0.058, 2 (-), 6 (↑5)
C09-Lacking confidence (OPC)	**0.028, 9, 12**	0.134, 2 (↑7), 2 (↑10)	0.022, 9 (-), 14 (↓2)	0.019, 8 (↑1), 15 (↓3)	0.053, 5 (↑4), 7 (↑5)
NC06-Challenges of security/DevSecOps requirements for businesses (B)	**0.025, 2, 13**	0.015, 2 (-), 18 (↓5)	0.037, 3 (↓1), 10 (↑3)	0.038, 3 (↓1), 9 (↑4)	0.016, 2 (-), 16 (↓3)
C12-Compliance requirements (PC)	**0.022, 3, 14**	0.013, 7 (↓4), 19 (↓5)	0.024, 1 (↑2), 13 (↑1)	0.022, 1 (↑2), 12 (↑2)	0.030, 3 (-), 11 (↑3)
C13-Neglecting change control in security (PC)	**0.022, 4, 15**	0.037, 2 (↑2), 9 (↑6)	0.012, 6 (↓2), 23 (↓8)	0.008, 7 (↓3), 27 (↓12)	0.058, 1 (↓2), 5 (↑10)
C26-Conflicts between security and business (B)	**0.021, 3, 16**	0.003, 5 (↓2), 32 (↓16)	0.056, 2 (↑1), 5 (↑11)	0.044, 2 (↑1), 7 (↑9)	0.005, 3 (-), 29 (↓13)
C17-Inadequate privileged credentials and access controls causing cyberattacks (PC)	**0.020, 5, 17**	0.017, 6 (↓1), 15 (↑2)	0.020, 3 (↑2), 17 (-)	0.020, 2 (↑3), 14 (↑3)	0.009, 10 (↓5), 23 (↓6)
NC03-Lack of reference model for DevSecOps process (PC)	**0.020, 6, 18**	0.037, 4 (↑2), 11 (↑7)	0.010, 7 (↓1), 24 (↓6)	0.010, 6 (-), 25 (↓7)	0.010, 9 (↓3), 21 (↓3)
C20-Challenges of legacy system refactoring (T)	**0.019, 1, 19**	0.043, 1 (-), 6 (↑13)	0.013, 2 (↓1), 22 (↓3)	0.013, 2 (↓1), 22 (↓3)	0.038, 1 (-), 8 (↑11)
NC05-Code security/code developers working on their dependences/ business logic, excluding external tools (T)	**0.019, 2, 20**	0.003, 8 (↓6), 34 (↓14)	0.021, 1 (↑1), 15 (↑5)	0.024, 1 (↑1), 11 (↑9)	0.016, 2 (-), 20 (-)
C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance (PC)	**0.015, 7, 21**	0.005, 9 (↓2), 28 (↓7)	0.017, 5 (↑2), 19 (↑2)	0.018, 4 (↑3), 18 (↑3)	0.016, 6 (↑1), 17 (↑4)
C06-Recruiting challenges (OPC)	**0.012, 10, 22**	0.015, 7 (↑3), 17 (↑5)	0.014, 10 (-), 21 (↑1)	0.010, 10 (-), 24 (↓2)	0.019, 8 (↑2), 14 (↑8)
C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth (T)	**0.009, 3, 23**	0.006, 5 (↓2), 27 (↓4)	0.005, 4 (↓1), 30 (↓7)	0.009, 4 (↓1), 26 (↓3)	0.008, 5 (↓2), 25 (↓2)
NC04-Domain of application is also matters like cyber physical systems (T)	**0.009, 4, 24**	0.003, 9 (↓5), 33 (↓9)	0.008, 3 (↑1), 25 (↓1)	0.012, 3 (↑1), 23 (↑1)	0.008, 6 (↑2), 26 (↓2)
C16-Poor visibility of security track	**0.009,**	0.017,	0.006,	0.004,	0.031,

record (PC)	**8, 25**	5 (↑3), 14 (↑9)	10 (↓2), 29 (↓4)	10 (↓2), 31 (↓6)	4 (↑4), 12 (↑13)
C14-Lack of standards (PC)	**0.007, 9, 26**	0.009, 8 (↑1), 23 (↑3)	0.006, 9 (-), 28 (↓2)	0.004, 9 (-), 32 (↓6)	0.017, 5 (↑4), 15 (↑11)
C11-Using unsuitable metrics (PC)	**0.007, 10, 27**	0.005, 10 (-), 29 (↓2)	0.008, 8 (↑2), 26 (↑1)	0.006, 8 (↑2), 28 (↓1)	0.016, 7 (↑3), 18 (↑9)
C27-Customer readiness for frequent releases (B)	**0.007, 4, 28**	0.004, 4 (-), 30 (↓2)	0.015, 5 (↓1), 20 (↑8)	0.024, 4 (-), 10 (↑18)	0.002, 5 (↓1), 34 (↓6)
C28-Training users for using advanced tools (B)	**0.007, 5, 29**	0.008, 3 (↑2), 24 (↑5)	0.008, 6 (↓1), 27 (↑2)	0.014, 6 (↓1), 21 (↑8)	0.005, 4 (↑1), 30 (↓1)
C25-Challenges of cost control (B)	**0.006, 6, 30**	0.001, 6 (-), 35 (↓5)	0.029, 4 (↑2), 11 (↑19)	0.015, 5 (↑1), 20 (↑10)	0.0018, 6 (-), 35 (↓5)
C18-Lack of mature tools for automation and security (T)	**0.005, 5, 31**	0.043, 2 (↑3), 5 (↑26)	0.001, 8 (↓3), 34 (↓3)	0.003, 8 (↓3), 34 (↓3)	0.005, 7 (↓2), 31 (-)
C19-Complexity in managing different tools (T)	**0.005, 6, 32**	0.028, 3 (↑3), 13 (↑19)	0.003, 7 (↓1), 33 (↓1)	0.003, 7 (↓1), 33 (↓1)	0.009, 3 (↑3), 22 (↑10)
C21-Use of cloud and serverless computing brings security complications (T)	**0.005, 7, 33**	0.012, 4 (↑3), 21 (↑12)	0.003, 5 (↑2), 31 (↑2)	0.004, 5 (↑2), 30 (↑3)	0.008, 4 (↑3), 24 (↑9)
C24-Continuous deployment chaos (T)	**0.005, 8, 34**	0.004, 7 (↑1), 31 (↑3)	0.003, 6 (↑2), 32 (↑2)	0.004, 6 (↑2), 29 (↑5)	0.004, 8 (-), 32 (↑2)
C22-Containers and other tools come with their own risks (T)	**0.002, 9, 35**	0.006, 6 (↑3), 26 (↑9)	0.001, 9 (-), 35 (-)	0.002, 9 (-), 35 (-)	0.003, 9 (-), 33 (↑2)

The line graphs in **Figure 6. 10** compare the overall ranking of DevSecOps challenges with the rankings based on four participants' roles, indicating significant dissents. The red line represents the overall rankings of 35 revised DevSecOps challenges; the other four coloured lines represent the rankings given by four participants' roles: Academic, Industrial, Managerial, and Technical.

Figure 6. 10 - Round Two dissents on rankings of DevSecOps challenges based on roles



There are two distinct viewpoints illustrated in **Figure 6. 10**. On the left-hand side, 35 revised challenges are ordered by their identifiers within categories, showing differences in rankings within each category. For example, it shows that the challenges in the "Organisation, People and Culture" category (C01 – NC01) have more distinct rankings than those in the other categories. The line graph on the right-hand side orders the DevSecOps challenges by rankings from first to 35th place, showing the differences in rankings more visually. The further a point is from the red line, the greater the difference between the two rankings. For example, the academic role (purple line) shows more dissenting opinions than the other roles.

By comparing **Figure 6. 10** with **Figure 5. 4** (in Section 5.4.1, Chapter 5, Page 136), which showed differences in rankings of DevSecOps challenges in Round One, it was found that the rough consensus on rankings of DevSecOps challenges that had been achieved in Round One was broken in Round Two. Participants with different roles held more dissenting opinions than in Round One, despite having reviewed Round One results before responding to Round Two. On the other hand, as discussed above, CVs in Round Two were lower than in Round One (**Table 5. 18** in Section 5.4.1, Chapter 5, Page 128), indicating that participants with similar roles had a lower degree of dissent in this round.

By summarising the AHP results of Round Two and comparing them with the results of Round One, it could be concluded that, after the first and second iterations of the Delphi study, the participants with similar roles had relatively consistent opinions on the prioritisation of DevSecOps challenges; the participants with different roles had significantly dissenting opinions.

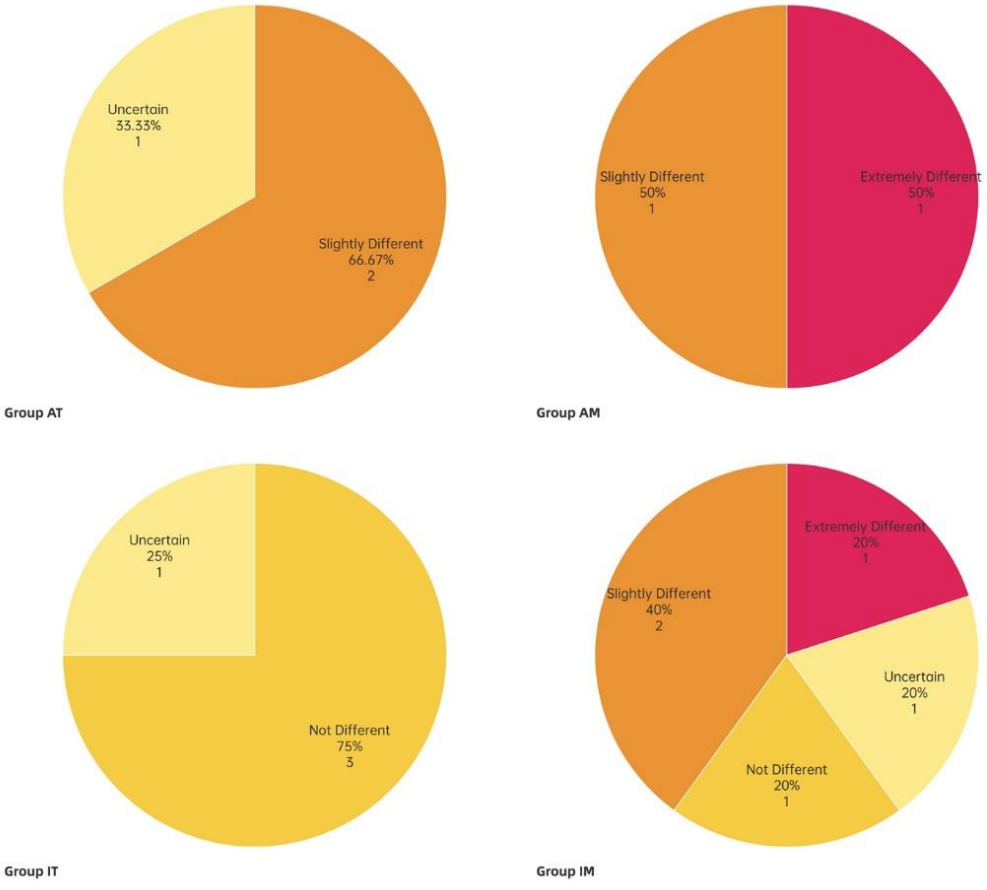
6.5.2 Dissent on Global DevSecOps Practices in Round Two

As shown in **Figure 6. 8** in Section 6.4 (Page 173), the opinions of "Slightly Different" and "Not Different" ranked first and second, respectively, with 35.71% and 28.57%, together accounting for 64.28%.

The four pie charts in **Figure 6. 11** compare the opinions of four participants' groups. The two opinions of "Slightly Different" and "Not Different" were evenly distributed in each group. Two participants opted for "Extremely Different", and they were all in managerial roles – one industrial managerial participant (Pa18) from New Zealand and one academic managerial participant (Pa5)

from Ireland. As narrow island countries with developed economies, New Zealand and Ireland are inclined to be somewhat insular; geographical distances may pose additional challenges due to limited access to various resources, as previously discussed in Section 5.4.1, Chapter 5 (Page 127). In such cases, practices in those countries would differ significantly from those in other countries when adopting DevSecOps in a GSE context. For instance, as commented by participant Pa18, "New Zealand companies often do not have the budgets to do everything, so there are more trade-offs".

Figure 6. 11 - Opinions on "How do DevSecOps practices differ in local and global settings?" in four groups



In addition, three of the four groups had one participant who opted for "Uncertain", and none of the participants suggested adding any global DevSecOps practice in this round. Both results reveal a consensus that global DevSecOps practices have not been systematically recognised, at least not to the same extent as the challenges dimension. To sum up, participants reached consensus that

DevSecOps practices differ slightly between local and global contexts, and these differences have not been well recognised.

6.6 Chapter Summary

This chapter presents the results of the Round Two Delphi survey. 14 participants were involved in Round Two, down from the 18 in Round One. Four academic participants were absent or withdrew in this round.

In Round Two, the importance of 35 DevSecOps challenges (revised) and 60 DevSecOps practices were compared through AHP pairwise comparisons across four categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology. The priorities and rankings of these challenges and practices were collected, calculated, and reported by using multiple tools, including Qualtrics, SuperDecisions, and MS Excel spreadsheets. Consistency ratios were evaluated to ensure that the AHP results fell within an acceptable range.

In addition to AHP pairwise comparisons, three additional DevSecOps practices were identified from participants' comments. Furthermore, participants' opinions on the differences between local and global DevSecOps practices were surveyed. Most participants believed that DevSecOps practices differ slightly between local and global contexts, and these differences have not been well recognised.

A dissent analysis was conducted to examine participants' agreement and disagreement with the AHP comparison results and the global DevSecOps opinions, using both quantitative and qualitative methods. Based on the results of Rounds One and Two, it can be concluded that participants with similar roles held more consistent opinions on prioritising DevSecOps challenges; participants with different roles had significantly differing opinions. Regarding the difference between local and global DevSecOps practices, the participants reached a consensus in Round Two, focusing on "Slightly Different" and "Not Different".

During the evaluation of DevSecOps practices in Round Two, the AHP results revealed two limitations: high inconsistencies in CR scores (ranging from 0.1 to 0.2) and abnormal rankings of practices due to recency bias among participants (i.e., the rankings of later-ordered practices were

significantly higher than those of earlier-ordered practices). A large number of observations in a single matrix to be compared could be the cause of both limitations. Hence, for Round Three, the AHP structure and questionnaire were revised by adding an AHP layer of sub-category between the category and the practices, aimed at reducing the volume of observations in a single pairwise comparison matrix and mitigating both limitations.

To sum up, the evaluation of DevSecOps challenges has been completed in this round; the above findings have partly answered RQ3, RQ4, and associated sub-questions, i.e., have addressed the dimension of challenges. On the other hand, the evaluation of DevSecOps practices was affected by two limitations, i.e., inconsistencies and recency bias. The AHP structure and the questionnaire were revised to mitigate both limitations before conducting Round Three. Therefore, a deep discussion was not presented in this chapter. It was placed in the next chapter once the third round of the Delphi survey was completed.

Next, Chapter 7 reports the results of Round Three, in which participants reassessed the importance of the revised DevSecOps practices and subsequently assessed the importance of 20 identified DevSecOps metrics. Round Two results were shared with participants to assist them in responding to Round Three, so that the findings could be iteratively verified and improved as the Delphi process progressed.

7 Chapter 7: Round Three Results – Revised DevSecOps Practices and DevSecOps Metrics

This chapter presents the results of the Round Three Delphi survey, conducted between March and June 2025, involving 15 participants. Round Three was the final iteration of the Delphi study. In this round, the importance of 63 DevSecOps practices (revised) and 20 DevSecOps metrics was compared by using the AHP method (pairwise comparisons) across the four categories: Business, Organisation, People & Culture (OPC), Process Capabilities (PC), and Technology.

First, the priorities and rankings of 63 revised DevSecOps practices (60 initially identified practices plus three new ones added during Round Two) and four categories were calculated and reported to confirm the findings of Round Two.

Second, the priorities and rankings of 20 identified DevSecOps metrics were calculated and reported. Consistency ratios for the AHP comparisons were calculated, and all were within an acceptable range.

Third, based on participants' comments, two additional metrics were identified and incorporated to improve the DevSecOps CPTM Model (Version 1.0).

Fourth, participants' opinions on the differences between local and global DevSecOps metrics were collected and reported.

Finally, a dissent analysis was conducted to examine participants' agreement and disagreement with the AHP comparison results and global DevSecOps opinions, by using both quantitative and qualitative methods.

In Section 7.1, the context and background for conducting the Round Three survey are provided, including the survey objectives, questionnaire content, and participants' information. The AHP comparison results for the revised DevSecOps practices are reported and analysed in Section 7.2. Next, the AHP comparison results for DevSecOps metrics are presented in Section 7.3, along with additional metrics newly identified from participants' comments. In Section 7.4, participants' opinions on the difference between local and global DevSecOps metrics are reported and analysed.

Section 7.5 provides information on the dissent analysis for the Round Three survey. Section 7.6 concludes the Round Three survey and summarises the entire Delphi study.

7.1 Context and Background of Round Three

This section provides the context and background for the Round Three Delphi survey, including its objectives, questionnaire content, and participants' information.

7.1.1 Survey Objectives in Round Three

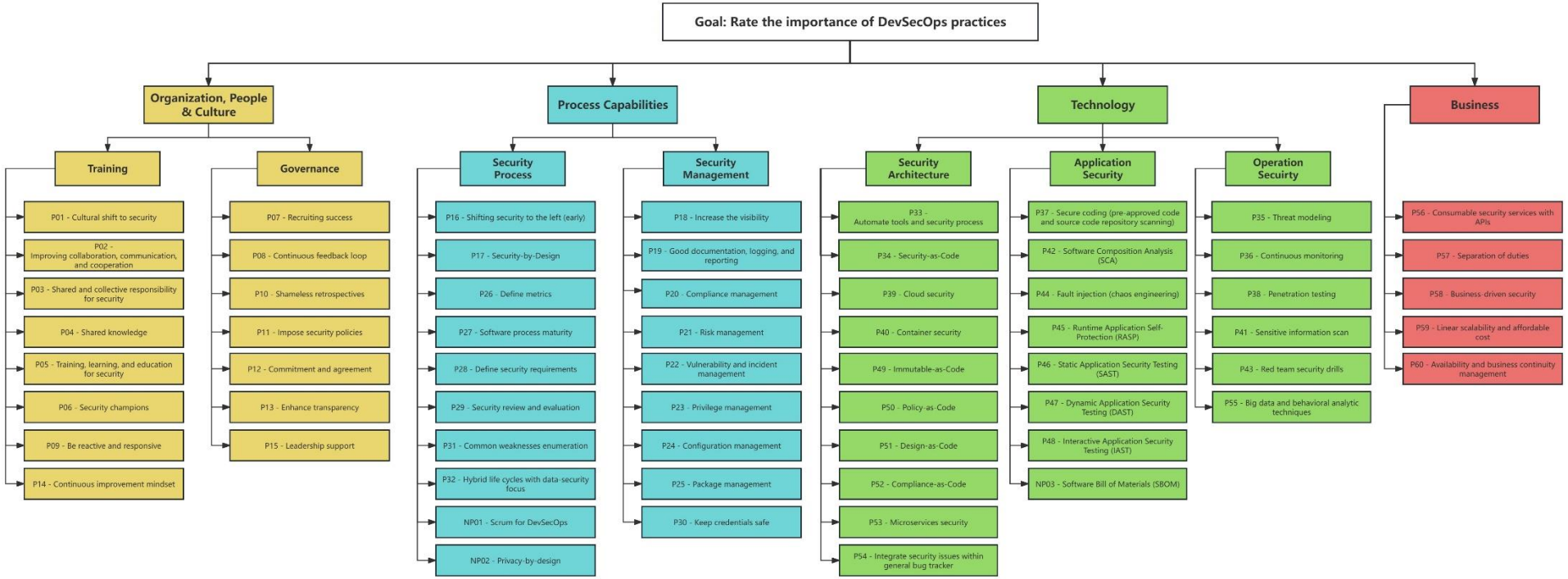
The third round of the Delphi survey was conducted for multiple objectives.

First, to verify and refine the results of the previous Round Two, at the beginning of Round Three, the importance of 63 revised DevSecOps practices was reassessed using AHP-based closed-ended questions (i.e., AHP pairwise comparisons). Before conducting Round Three, the Round Two survey results were shared with participants to help them reassess these practices.

During the evaluation of DevSecOps practices in Round Two, the AHP results revealed two limitations: high inconsistencies in CR scores (ranging from 0.1 to 0.2) and abnormal rankings of practices due to recency bias among participants (i.e., later-ordered practices were ranked significantly higher than earlier-ordered practices). A large number of observations in a single matrix to be compared could be the cause of both limitations. Hence, for Round Three, the AHP structure and questionnaire were revised by adding an AHP layer of sub-category between the category and the practices, aimed at reducing the volume of observations in a single pairwise comparison matrix and mitigating both limitations.

Hence, as shown in **Figure 7. 1**, the AHP structure was revised by including an additional AHP layer of sub-category between the category and the practices, aimed to reduce the volume of observations in a single pairwise comparison matrix. The final results showed that the issues caused by both limitations have been effectively mitigated in the Round Three Delphi survey.

Figure 7. 1 - Revised AHP structure for DevSecOps practices

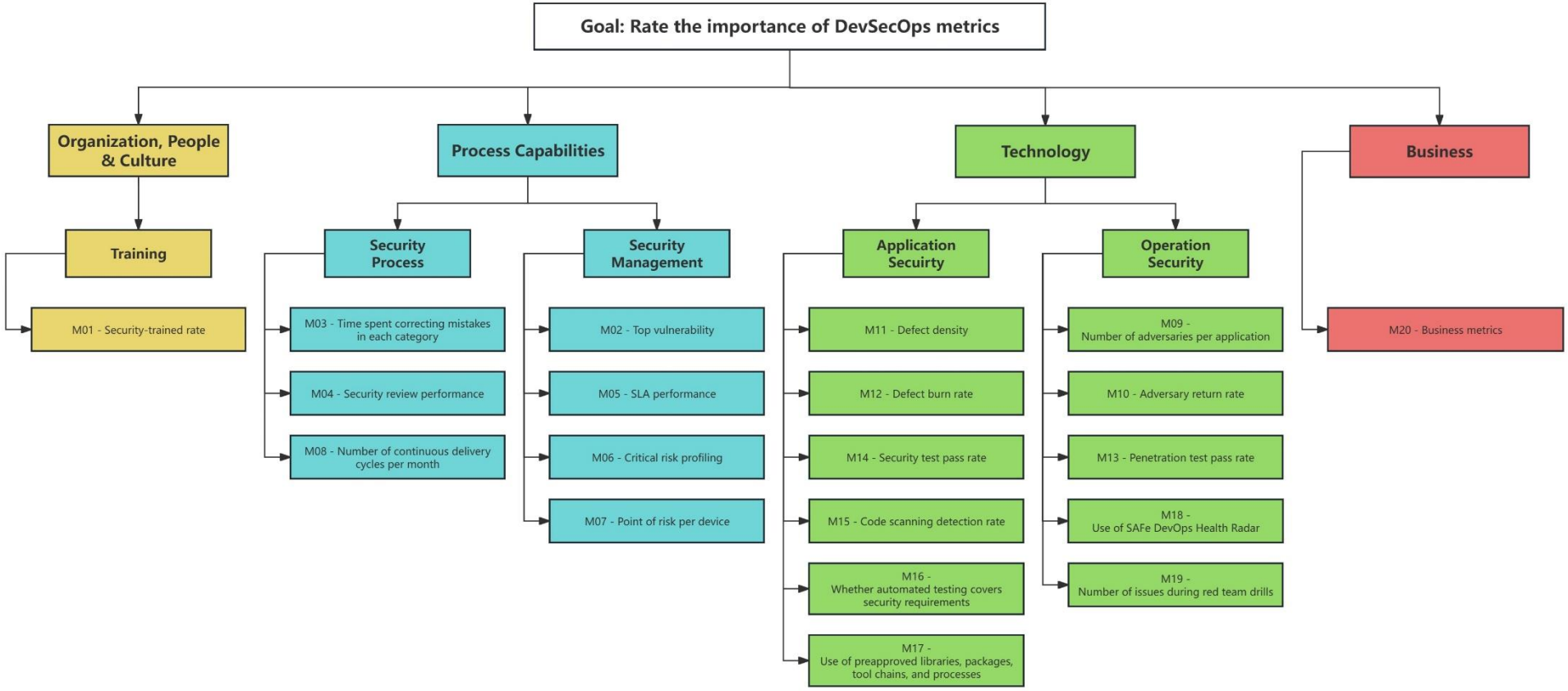


The other primary purpose of Round Three was to rate and rank the importance of 20 identified DevSecOps metrics through AHP pairwise comparisons, to answer *RQ3: "How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?"*

Unlike the challenges and practices of DevSecOps, the metrics dimension was considered less extensively researched and less widely understood. Therefore, a glossary of the identified metrics was provided to participants before the Round Three survey.

To prevent the issues of respondent inconsistency and recency bias occurring in the evaluation of DevSecOps metrics, the AHP structure and the questionnaire were proactively revised, as depicted on **Figure 7. 2**. Moreover, participants were expected to identify new DevSecOps metrics, to address *Sub-question 3.1: "What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?"* New findings on metrics were evaluated, discussed, and integrated with the DevSecOps CPTM Model.

Figure 7. 2 - Revised AHP structure for DevSecOps metrics



To address *Sub-question 3.2: "Do experts have dissents on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*, the participants were divided in four groups: Group Academic Managerial (AM), Group Industrial Managerial (IM), Group Academic Technical (AT), and Group Industrial Technical (IT). All these groups answered the same questionnaire, enabling an investigation into whether participants held dissenting opinions due to differences in role and organisational level. A dissent analysis was conducted accordingly for this survey round.

Finally, Participants' opinions on the differences between DevSecOps metrics in local and global settings enabled the addressing of *RQ4: "What are the experts' opinions on DevSecOps in GSE contexts?"* and its two related sub-questions: *"Sub-question 4.1: How is DevSecOps different between local and global settings?"* and *"Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?"*

7.1.2 Questionnaire Contents in Round Three

To achieve the above survey objectives and address the corresponding research questions, the Round Three survey questionnaire consisted of 22 questions, employing multiple formats: AHP pairwise comparisons (Questions 1–19), multiple-choice (Question 21), and open-ended questions (Questions 20 and 22).

Table 7. 1 and **Table 7. 2** list these questions, including objectives, format types, and descriptions. Due to the repetition of evaluating practices and the large volume of AHP comparisons, the minimum number of comparisons was used to assess importance rather than completing all pairwise comparisons. Once participants had completed these comparisons, the researcher calculated the remaining comparison judgments. Minimising the number of AHP comparisons eliminated redundant questions, reduced unnecessary workloads and survey durations for participants, and mitigated inconsistencies caused by redundant pairwise comparisons (Mu & Pereyra-Rojas, 2017). A sample of the Round Three survey is provided in Appendix B.10, "Sample of Delphi Survey – Round Three" (Page 343), or it can be assessed at the Qualtrics platform (https://aut.au1.qualtrics.com/jfe/form/SV_009hlLHQHeCyMmi).

Table 7. 1 - List of questions in Round Three for the evaluation of revised practices

Question ID	Objective	Format	Description
Q1	To rate the importance of the four categories of DevSecOps practices	6 AHP pairwise comparisons	Business : OPC Business : PC Business : Technology OPC : PC OPC : Technology PC : Technology
Q2	To rate the importance of two sub-categories of the OPC (Organisation, People & Culture) category	1 AHP pairwise comparison	People Training : Organisational Governance
Q3	To rate the importance of eight DevSecOps practices in the "People Training" sub-category in the OPC (Organisation, People & Culture) category	7 AHP pairwise comparisons	P01 : P02; P02 : P03; P03 : P04; P04 : P05; P05 : P06; P06 : P09; P09 : P14
Q4	To rate the importance of seven DevSecOps practices in the "Organisational Governance" sub-category, in the OPC (Organisation, People & Culture) category	6 AHP pairwise comparisons	P07 : P08; P08 : P10; P10 : P11; P11 : P12; P12 : P13; P13 : P15
Q5	To rate the importance of two sub-categories of the PC (Process Capabilities) category	1 AHP pairwise comparison	Security Process : Security Management
Q6	To rate the importance of eight initially identified DevSecOps practices and two newly collected practices in the "Security Process" sub-category, in the PC (Process Capabilities) category	9 AHP pairwise comparisons	P16 : P17; P17 : P26; P26 : P27; P27 : P28; P28 : P29; P29 : P31; P31 : P32; P32 : NP01; NP01: NP02
Q7	To rate the importance of nine DevSecOps practices in the "Security Management" sub-category, in the PC (Process Capabilities) category	8 AHP pairwise comparisons	P18 : P19; P19 : P20; P20 : P21; P21: P22; P22 : P23; P23 : P24; P24 : P25; P25 : P30
Q8	To rate the importance of three sub-categories of the Technology category	3 AHP pairwise comparisons	Security Architecture : Application Security; Security Architecture : Operation Security; Application Security : Operation Security

Q9	To rate the importance of ten DevSecOps practices in the "Security Architecture" sub-category, in the Technology category	9 AHP pairwise comparisons	P33 : P34; P34 : P39; P39 : P40; P40: P49; P49 : P50; P50 : P51; P51 : P52; P52 : P53; P53 : P54
Q10	To rate the importance of seven initially identified DevSecOps practices and one newly collected practice in the "Application Security" sub-category, in the Technology category	7 AHP pairwise comparisons	P37 : P42; P42 : P44; P44 : P45; P45: P46; P46 : P47; P47 : P48; P48 : NP03
Q11	To rate the importance of six DevSecOps practices in the "Operation Security" sub-category, in the Technology category	5 AHP pairwise comparisons	P35 : P36; P36 : P38; P38 : P41; P41: P43; P43 : P55
Q12	To rate the importance of the five DevSecOps practices in the Business category	4 AHP pairwise comparisons	P56 : P57; P57 : P58; P58 : P59; P59 : P60

Table 7. 2 - List of questions in Round Three for the evaluation of metrics

Question ID	Objective	Format	Description
Q13	To rate the importance of the four categories of DevSecOps metrics	6 AHP pairwise comparisons	Business : OPC Business : PC Business : Technology OPC : PC OPC : Technology PC : Technology
Q14	To rate the importance of two sub-categories of the PC (Process Capabilities) category	1 AHP pairwise comparison	Security Process : Security Management
Q15	To rate the importance of 3 DevSecOps metrics in the "Security Process" sub-category, in the PC (Process Capabilities) category	2 AHP pairwise comparisons	M03 : M04; M04 : M08
Q16	To rate the importance of 4 DevSecOps metrics in the "Security Management" sub-category, in the PC (Process Capabilities) category	3 AHP pairwise comparisons	M02 : M05; M05 : M06; M06 : M07
Q17	To rate the importance of two sub-categories of the Technology category	1 AHP pairwise comparison	Application Security : Operation Security
Q18	To rate the importance of 6 DevSecOps metrics in the "Application Security" sub-category, in the Technology category	5 AHP pairwise comparisons	M11 : M12; M12 : M14; M14 : M15; M15 : M16; M16 : M17
Q19	To rate the importance of 5 DevSecOps metrics in the "Operation Security" sub-category, in the Technology category	4 AHP pairwise comparisons	M09 : M10; M10 : M13; M13 : M18; M18 : M19
Q20	To collect additional DevSecOps metrics	1 open-ended question	Add more metrics
Q21	To investigate how DevSecOps metrics differ in local and global settings	1 multiple-choice question	4 options: Extremely different; Slightly different; Not different; Uncertain
Q22	To collect differences between local and global DevSecOps metrics	1 open-ended question	List the differences if opted for "Extremely/Slightly Different" in Q21

7.1.3 Participants in Round Three

15 participants completed the Round Three Delphi survey on schedule. Compared to the 18 initial participants in Round One (**Table 5. 2** in Section 5.1.3, Chapter 5, Page 112), three academic participants, i.e., Pa4, Pa7, and Pa9, dropped out of Rounds Two and Three. Compared to the 14 participants in Round Two (**Table 6. 2** in Section 6.1.3, Chapter 6, Page 144), an academic participant, Pa6, returned in Round Three. This participant had finished Round Two but was late; as a result, his responses were not counted.

While the Delphi survey ought to be anonymous, **Table 7. 3** provides brief information about the participants in the Round Three survey.

Table 7. 3 - List of participants in Round Two

Group	Part. ID	Occupation / Role	Academic	Industry			Country
				Portfolio	Program	Team	
AT	Pa1	Researcher	1				Belgium
	Pa2	Assistant Professor	1				Canada
	Pa3	Senior Lecturer	1				NZ
AM	Pa5	Professor	1				Ireland
	Pa6	Associate Professor	1				Ireland
	Pa8	Researcher / Technical Manager / Senior Security Architect	1	1	1		Germany
IT	Pa10	Principal Consultant			1	1	USA
	Pa11	Principal Software Engineer		1	1	1	Germany
	Pa12	Senior DevOps Engineer			1	1	China
	Pa13	DevOps Engineer			1	1	USA
IM	Pa14	Head Marketing and Communications		1	1		India
	Pa15	General Manager (Operations side)		1	1	1	NZ
	Pa16	Senior Security Consultant		1	1		NZ
	Pa17	Senior Agile Coach		1	1		NZ, China
	Pa18	Consultant (Adviser) / Researcher	1	1	1		NZ

To ensure the representativeness of participants and the reliability of the survey, participants were recruited from around the world and distributed across New Zealand, Europe, Asia, and North America. These participants have diverse roles and work in different fields. They could be academic, industrial, managerial, technical, or have multiple identities.

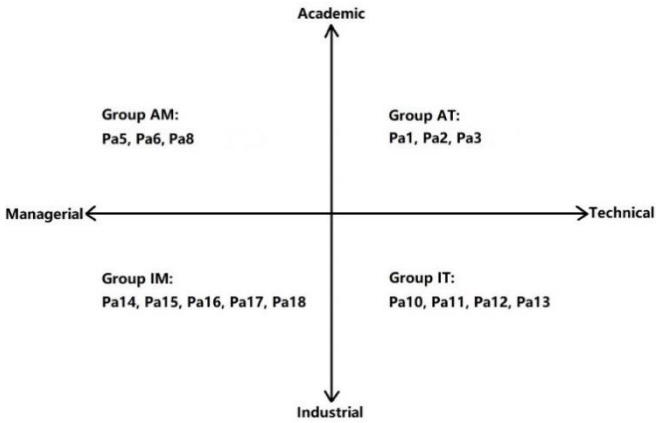
- Participants Pa1, Pa2, Pa3, Pa5, Pa6, and Pa8 are academic experts.

- Pa10 – Pa18 are industrial experts.
- Pa5, P6, Pa8, and Pa14 – Pa18 are managerial experts.
- Pa1 – Pa3 and Pa10 – 13 are technical experts.

According to the four-quadrant division for experts grouping, which has been presented in Section 4.5.2.3, Chapter 4 (Page 89), the 15 participants were further divided into four groups, as shown in **Figure 7. 3**:

- Participants Pa1, Pa2, and Pa3 are in the Group Academic Technical (AT).
- Pa5, Pa6, and Pa8 are in the Group Academic Managerial (AM).
- Pa10, Pa11, Pa12, Pa13 are in the Group Industrial Technical (IT).
- Pa14, Pa15, Pa16, Pa17, and Pa18 are in the Group Industrial Managerial (IM).

Figure 7. 3 - Four-quadrant division for participants grouping in Round Three



Regardless of the group placement, all participants were to answer the same questionnaire. Industrial participants were from large-scale or globally distributed organisations, and were further differentiated into three organisational levels: Portfolio level, Program level, and Team level (Beecham et al., 2021). By doing so, it enabled an investigation into whether participants held dissenting opinions due to differences in roles and organisational levels.

7.2 RQ3 – Evaluation of Revised DevSecOps Practices

This section presents the results of the AHP pairwise comparisons from Questions 1 to 12 (refer to **Table 7. 1**) that were conducted to reassess the importance of 60 initially identified DevSecOps practices plus three newly identified practices from Round Two, within their categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology, and their associated sub-categories. The priorities and rankings of these 63 challenges were calculated, discussed, and reported by comparing them with the results from Round Two.

All consistency ratios were within the acceptable range (between 0 and 0.1) (Saaty, 1982). The degree of dissent among participants, i.e., the value of Coefficient of Variation (CV), was within the acceptable range (between 0 and 0.5) (Dajani, Sincoff, & Talley, 1979) and close to that in Round Two. Additionally, the AHP results showed that the two issues identified in Round Two, i.e., high inconsistency and recency bias, have been effectively mitigated in this round. Hence, the evaluation of the practices dimension of DevSecOps has been conducted thoroughly across the second and third iterations of the Delphi survey.

7.2.1 AHP Results of Revised DevSecOps Practices

In Round Two, due to the repetition of evaluating challenges, the minimum number of comparisons was used to assess importance rather than completing all AHP pairwise comparisons. By doing so, redundant questions were removed, thereby reducing unnecessary workloads, survey durations, and inconsistencies in AHP comparisons (Mu & Pereyra-Rojas, 2017). Once the participants completed a minimum number of comparisons, the remaining comparison judgments were calculated.

Refer to **Table 7. 1**, Questions 1 – 12 were used to conduct AHP pairwise comparisons for 63 revised DevSecOps practices, their categories, and sub-categories. The data and responses were collected by using the Qualtrics online survey platform (Qualtrics, 2024). The original response sheets contain identifiable participant information, such as IP addresses; therefore, they cannot be publicly disclosed due to confidentiality constraints. The processed dataset of responses is available at zenodo.org (https://doi.org/10.5281/zenodo.16932278).

A total of 15 participants were involved in Round Three. The AHP method was employed to build

compromise and consensus among participants, so the data were analysed at the group level rather than the individual level. The extracted data were averaged to evaluate the AHP pairwise comparisons and obtain a set of group judgements. The Consistency Ratio (CR) was calculated to assess the consistency of the AHP comparisons. The Coefficient of Variation (CV) was calculated to assess the degree of consensus or dissent among participants. Section 4.5.3 in Chapter 4 (Page 103) has introduced the data analysis process in detail.

AHP comparison matrices:

Table 7. 4 presents a standard 9-point AHP Scale, which was used to quantify priorities (Saaty, 2013).

Table 7. 4 - 9-point AHP comparison scale

Scale	Numerical score	Reciprocal
Equally important	1	1
Moderately more important	3	1/3
Strongly more important	5	1/5
Very strongly more important	7	1/7
Extremely more important	9	1/9
Intermediate values	2, 4, 6, 8	1/2, 1/4, 1/6, 1/8

Table 7. 5 provides the AHP comparison matrix for the four categories of DevSecOps practices: Organisation, People and Culture (OPC), Process Capabilities (PC), Technology, and Business.

Table 7. 5 - AHP comparison matrix for categories of revised practices

Categories of Practices	Business	OPC	PC	Technology
Business	1/1	1/1	1/1	3/1
OPC	1/1	1/1	4/1	4/1
PC	1/1	1/4	1/1	2/1
Technology	1/3	1/4	1/2	1/1

Table 7. 6 provides the AHP comparison matrix for the sub-categories that were newly added in this round to address the issues of high inconsistency and recency bias.

Table 7. 6 - AHP comparison matrix for sub-categories of revised practices

Sub-categories in OPC	Training	Governance	
Training	1/1	2/1	
Governance	1/2	1/1	
Sub-categories in PC	**Process**	**Management**	
Process	1/1	2/1	
Management	1/2	1/1	
Sub-categories in Technology	**Architecture**	**Application**	**Operation**
Architecture	1/1	1/1	1/1
Application	1/1	1/1	1/1
Operation	1/1	1/1	1/1

Next, **Table 7. 7**, **Table 7. 8**, **Table 7. 9**, **Table 7. 10**, **Table 7. 11**, **Table 7. 12**, **Table 7. 13**, and **Table 7. 14** provide the AHP comparison matrices for 63 revised DevSecOps practices within their sub-category and category, by using the minimum number of comparisons (Mu & Pereyra-Rojas, 2017).

For example, in **Table 7. 7**, the minimum number of comparisons is comprised of only the comparison judgments in the diagonal (shaded) above the unit diagonal (filled with 1s) of the comparison matrix, e.g., $\frac{P01}{P02}$, $\frac{P02}{P03}$, $\frac{P03}{P04}$, $\frac{P04}{P05}$, $\frac{P05}{P06}$, etc. After the participants completed these comparisons, the researcher calculated the remaining comparison judgments in the upper and lower parts of the matrix. As shown in **Table 7. 7**, we knew the values $\frac{P01}{P02} = \frac{1}{1}$ and $\frac{P02}{P03} = \frac{1}{2}$, then we calculated the value $\frac{P01}{P03} = \frac{P01}{P02} * \frac{P02}{P03} = \frac{1}{1} * \frac{1}{2} = \frac{1}{2}$, and the value of $\frac{P03}{P01}$ was equal to reciprocal of the value $\frac{C01}{C03} = \frac{2}{1} = 2$. The exact process was repeated for the remaining cells in the matrix using Excel spreadsheets.

Table 7. 7 - AHP comparison matrix for revised practices in "Training" sub-category, in "Organisation, People & Culture" category

Practices in Organisation, People and Culture (Training)	P01	P02	P03	P04	P05	P06	P09	P14
P01	1/1	1/1	1/2	3/2	1/2	1/1	1/1	1/4
P02	1/1	1/1	1/2	3/2	1/2	1/1	1/1	1/4
P03	2/1	2/1	1/1	3/1	1/1	2/1	2/1	1/2
P04	2/3	2/3	1/3	1/1	1/3	2/3	2/3	1/6
P05	2/1	2/1	1/1	3/1	1/1	2/1	2/1	1/2
P06	1/1	1/1	1/2	3/2	1/2	1/1	1/1	1/4
P09	1/1	1/1	1/2	3/2	1/2	1/1	1/1	1/4
P14	4/1	4/1	2/1	6/1	2/1	4/1	4/1	1/1

Table 7. 8 - AHP comparison matrix for revised practices in "Governance" sub-category, in "Organisation, People & Culture" category

Practices in Organisation, People and Culture (Governance)	P07	P08	P10	P11	P12	P13	P15
P07	1/1	1/3	1/6	1/3	0/1	0/1	0/1
P08	3/1	1/1	1/2	1/1	1/4	1/4	0/1
P10	6/1	2/1	1/1	2/1	1/2	1/2	1/6
P11	3/1	1/1	1/2	1/1	1/4	1/4	0/1
P12	12/1	4/1	2/1	4/1	1/1	1/1	1/3
P13	12/1	4/1	2/1	4/1	1/1	1/1	1/3
P15	36/1	12/1	6/1	12/1	3/1	3/1	1/1

Table 7. 9 - AHP comparison matrix for revised practices in "Security Process" sub-category, in "Process Capabilities" category

Practices in Process Capabilities (Security Process)	P16	P17	P26	P27	P28	P29	P31	P32	NP01	NP02
P16	1/1	1/1	4/1	2/1	2/3	2/3	2/3	2/3	2/3	1/6
P17	1/1	1/1	4/1	2/1	2/3	2/3	2/3	2/3	2/3	1/6
P26	1/4	1/4	1/1	1/2	1/6	1/6	1/6	1/6	1/6	0/1
P27	1/2	1/2	2/1	1/1	1/3	1/3	1/3	1/3	1/3	0/1
P28	3/2	3/2	6/1	3/1	1/1	1/1	1/1	1/1	1/1	1/4
P29	3/2	3/2	6/1	3/1	1/1	1/1	1/1	1/1	1/1	1/4
P31	3/2	3/2	6/1	3/1	1/1	1/1	1/1	1/1	1/1	1/4
P32	3/2	3/2	6/1	3/1	1/1	1/1	1/1	1/1	1/1	1/4
NP01	3/2	3/2	6/1	3/1	1/1	1/1	1/1	1/1	1/1	1/4
NP02	6/1	6/1	24/1	12/1	4/1	4/1	4/1	4/1	4/1	1/1

Table 7. 10 - AHP comparison matrix for revised practices in "Security Management" sub-category, in "Process Capabilities" category

Practices in Process Capabilities (Security Management)	P18	P19	P20	P21	P22	P23	P24	P25	P30
P18	1/1	1/2	1/2	1/8	0/1	0/1	0/1	0/1	0/1
P19	2/1	1/1	1/1	1/4	1/8	1/8	0/1	0/1	0/1
P20	2/1	1/1	1/1	1/4	1/8	1/8	0/1	0/1	0/1
P21	8/1	4/1	4/1	1/1	1/2	1/2	1/4	1/4	0/1
P22	16/1	8/1	8/1	2/1	1/1	1/1	1/2	1/2	1/6
P23	16/1	8/1	8/1	2/1	1/1	1/1	1/2	1/2	1/6
P24	32/1	16/1	16/1	4/1	2/1	2/1	1/1	1/1	1/3
P25	32/1	16/1	16/1	4/1	2/1	2/1	1/1	1/1	1/3
P30	96/1	48/1	48/1	12/1	6/1	6/1	3/1	3/1	1/1

Table 7. 11 - AHP comparison matrix for revised practices in "Security Architecture" sub-category, in "Technology" category

Practices in Technology (Security Architecture)	P33	P34	P39	P40	P49	P50	P51	P52	P53	P54
P33	1/1	2/1	2/1	1/1	1/1	1/2	1/6	0/1	0/1	0/1
P34	1/2	1/1	1/1	1/2	1/2	1/4	0/1	0/1	0/1	0/1
P39	1/2	1/1	1/1	1/2	1/2	1/4	0/1	0/1	0/1	0/1
P40	1/1	2/1	2/1	1/1	1/1	1/2	1/6	0/1	0/1	0/1
P49	1/1	2/1	2/1	1/1	1/1	1/2	1/6	0/1	0/1	0/1
P50	2/1	4/1	4/1	2/1	2/1	1/1	1/3	1/9	1/9	0/1
P51	6/1	12/1	12/1	6/1	6/1	3/1	1/1	1/3	1/3	1/6
P52	18/1	36/1	36/1	18/1	18/1	9/1	3/1	1/1	1/1	1/2
P53	18/1	36/1	36/1	18/1	18/1	9/1	3/1	1/1	1/1	1/2
P54	36/1	72/1	72/1	36/1	36/1	18/1	6/1	2/1	2/1	1/1

Table 7. 12 - AHP comparison matrix for revised practices in "Application Security" sub-category, in "Technology" category

Practices in Technology (Application Security)	P37	P42	P44	P45	P46	P47	P48	NP03
P37	1/1	3/1	3/1	3/2	3/4	1/4	1/4	3/4
P42	1/3	1/1	1/1	1/2	1/4	0/1	0/1	1/4
P44	1/3	1/1	1/1	1/2	1/4	0/1	0/1	1/4
P45	2/3	2/1	2/1	1/1	1/2	1/6	1/6	1/2
P46	4/3	4/1	4/1	2/1	1/1	1/3	1/3	1/1
P47	4/1	12/1	12/1	6/1	3/1	1/1	1/1	3/1
P48	4/1	12/1	12/1	6/1	3/1	1/1	1/1	3/1
NP03	4/3	4/1	4/1	2/1	1/1	1/3	1/3	1/1

Table 7. 13 - AHP comparison matrix for revised practices in "Operation Security" sub-category, in "Technology" category

Practices in Technology (Operation Security)	P35	P36	P38	P41	P43	P55
P35	1/1	1/2	1/2	1/6	0/1	0/1
P36	2/1	1/1	1/1	1/3	1/6	1/6
P38	2/1	1/1	1/1	1/3	1/6	1/6
P41	6/1	3/1	3/1	1/1	1/2	1/2
P43	12/1	6/1	6/1	2/1	1/1	1/1
P55	12/1	6/1	6/1	2/1	1/1	1/1

Table 7. 14 - AHP comparison matrix for revised practices in "Business" category

Practices in Business	P56	P57	P58	P59	P60
P56	1/1	1/2	1/6	0/1	0/1
P57	2/1	1/1	1/3	1/6	0/1
P58	6/1	3/1	1/1	1/2	1/4
P59	12/1	6/1	2/1	1/1	1/2
P60	24/1	12/1	4/1	2/1	1/1

Priorities, rankings, and consistency ratios:

The AHP tool SuperDecisions (SuperDecisions, 2023) was employed to calculate and derive the priorities/weights and the consistency ratios. The calculation method and equations have been presented and discussed in Chapters 4 and 5, so further elaboration is not provided in this chapter.

Table 7. 15 reports the AHP results for four categories of revised DevSecOps practices. **Table 7. 16**, **Table 7. 17**, **Table 7. 18**, and **Table 7. 19** present the AHP results for the practices within each category and sub-category. In these tables:

- "Normalised Value" represents the priority/weight within its own matrix/category (SuperDecisions, 2023). It may easily be confused with another AHP terminology, "Local priority/weight", which is explicitly used for alternatives, rather than criteria and sub-criteria (Mu & Pereyra-Rojas, 2017). Here, categories are criteria, and challenges are sub-criteria.
- "Idealised Value" represents the relative proportion compared with the highest-priority object, which has an idealised value of "1" by default (SuperDecisions, 2023).
- "Ranking" refers to the ranking of each object within its own matrix/category, ordered by normalised/idealised values from high to low (SuperDecisions, 2023).
- "Consistency Ratio" represents the consistency among all pairwise comparisons in a matrix; its acceptable value should be between 0 and 0.1, and a lower value means more consistency (Saaty, 1982). In Round Three, for the assessment of practices, all consistency ratios were below 0.1, indicating that the AHP results were consistent (Mu & Pereyra-Rojas, 2017).

In addition to reporting the AHP results of Round Three, these tables also present comparisons with the AHP results of Round Two, to show how participants adjusted their judgements after reviewing the Round Two results.

Table 7. 15 - AHP results for categories of practices

Categories of Practices	Normalised Value	Idealised Value	Ranking
Round 3 (consistency ratio = 0.06395)			
Business	0.28455819606011173	0.64853201474020616	2
Organisation, People and Culture (Sub-Categories: Training; Governance)	0.43877278159367689 (Training: 0.67; Governance: 0.33)	1.0 (Training: 1.0; Governance: 0.5)	1 (highest)
Process Capabilities (Sub-Categories: Security Process; Security Management)	0.18325993725634487 (Process: 0.67; Management: 0.33)	0.41766477991347173 (Process: 1.0; Management: 0.5)	3
Technology (Sub-Categories: Architecture; Application; Operation)	0.093409085089866353 (Arch: 0.33; App: 0.33; Op: 0.33)	0.21288714571262379 (Arch: 1.0; App: 1.0; Op: 1.0)	4
Round 2 (consistency ratio = 0.06457)			
Business	0.097929712564908927	0.1718938552410755	3
Organisation, People and Culture	0.56971037404196745	1.0	1 (highest)
Process Capabilities	0.24617405097190503	0.4321038587122002	2
Technology	0.086185862421218548	0.15128013521984718	4

As shown above in **Table 7. 15**, the most important category of DevSecOps practices was "Organisation, People and Culture" with a normalised value of approximately 0.44, significantly higher than the other three categories. (All results were rounded to two decimal places for reporting.) The "Business" category ranked second with a priority of 0.28, followed by "Process Capabilities" with a priority of 0.18. The least important category was "Technology", with the lowest priority of 0.09. The normalised and idealised values of sub-categories are also provided.

Compared to the results of Round Two, in Round Three, the priority of the "Business" category increased sharply. It ranked higher than the "Process Capabilities" category, so that the second and third places reversed, while the top and bottom remained consistent. Thus, it can be concluded that the assessment of the importance of practice categories has achieved a stable result after two iterations of the Delphi survey.

This result has also validated the findings from Rounds One and Two, revealing the different focuses of DevSecOps challenges and practices in reality and in the literature. The MLR findings presented in Chapter 3 indicate that a majority of the identified challenges and practices are technology-related, with high frequency across both the white and grey literature. In contrast, few findings in the grey literature are business-related. On the contrary, the results of the three survey rounds show that the "Technology" category is the least important. This difference implies that technological DevSecOps challenges are easiest to identify and overcome through corresponding

technological practices. Business attention is critical to DevSecOps adoption in practice, but it lacks theoretical research.

Table 7. 16 lists the priorities and rankings of 15 DevSecOps practices in the "Organisation, People & Culture" category, within two sub-categories: "Training" and "Governance". Participants ranked "Training" over "Governance", with weights of 0.67 and 0.33, respectively. To obtain the normalised value of each practice within a category, its normalised value within the sub-category was multiplied by the weight of the sub-category. Hence, the top three important practices in this category were: "P14 – Continuous improvement mindset" (ranked 2nd in Round Two); "P15 – Leadership support" (ranked 1st in Round Two); and "P05 – Training, learning and education for security" (ranked 10th in Round Two).

Table 7. 16 - AHP results for practices in "Organisation, People and Culture" category

Practices in Organisation, People and Culture (Weight on Training: 0.67)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P01-Cultural shift to security	0.082157795023884733	4	0.054771863	7
P02-Improving collaboration, communication and cooperation	0.082157795023884733	5	0.054771863	8
P03-Shared and collective responsibility for security	0.15744761754447811	3	0.104965078	4
P04-Shared knowledge	0.055314356544698282	8	0.036876238	11
P05-Training, learning and education for security	0.15744763814841631	2	0.104965092	3
P06-Security champions	0.075289822520593422	6	0.050193215	9
P09-Be reactive and responsive	0.075289822520593422	7	0.050193215	10
P14-Continuous improvement mindset	0.31489515267345097	1	0.209930102	1
consistency ratio = 0.00551				
Practices in Organisation, People and Culture (Weight on Governance : 0.33)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P07-Recruiting success	0.02192786309115257	7	0.007309288	15
P08-Continuous feedback loop	0.048185086665473782	5	0.016061696	13
P10-Shameless retrospectives	0.092998308829064208	4	0.030999436	12
P11-Impose security policies	0.048185086665473782	6	0.016061696	14
P12-Commitment and agreement	0.17681959267736311	2	0.058939864	5
P13-Enhance transparency	0.17681959267736311	3	0.058939864	6
P15-Leadership support	0.43506446939410959	1	0.14502149	2
consistency ratio = 0.02077				

Table 7. 17 reports the priorities and rankings of 19 DevSecOps practices in the "Process Capabilities" category, within two sub-categories: "Security Process" and "Security

Management". "Security Process" was ranked higher than "Security Management", with weights of 0.67 and 0.33, respectively. The three most important practices in this category were: "NP02 – Privacy-by-Design" (a new practice identified in Round Two); "P30 – Keep credentials safe" (ranked 4th in Round Two); and "P28 – Define security requirements" (ranked 5th in Round Two).

Table 7. 17 - AHP results for practices in "Process Capabilities" category

Practices in Process Capabilities (Weight on Security Process: 0.67)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P16-Shifting security to the left (early)	0.078780363843781787	7	0.052520243	10
P17-Security-by-Design	0.078780363843781787	8	0.052520243	11
P26-Define metrics	0.018440278613983473	10	0.012293519	16
P27-Software process maturity	0.033062084202399498	9	0.022041389	14
P28-Define security requirements	0.088547195882594368	2	0.059031464	3
P29-Security review and evaluation	0.088547195882594368	3	0.059031464	4
P31-Common weaknesses enumeration	0.088547195882594368	4	0.059031464	5
P32-Hybrid life cycles with data-security focus	0.088547195882594368	5	0.059031464	6
NP01-Scrum for DevSecOps	0.088547195882594368	6	0.059031464	7
NP02-Privacy-by-Design	0.34820093008308167	1	0.232133953	1
consistency ratio = 0.00986				
Practices in Process Capabilities (Weight on Security Management: 0.33)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P18-Increase the visibility	0.014502876231687617	9	0.004834292	19
P19-Good documentation, logging and reporting	0.019015698342913168	7	0.006338566	17
P20-Compliance control	0.019015698342913168	8	0.006338566	18
P21-Risk management	0.05686132701106178	6	0.018953776	15
P22-Vulnerability and incident management	0.10105780503400265	4	0.033685935	12
P23-Privilege management	0.10105780503400265	5	0.033685935	13
P24-Configuration management	0.16089290350220609	2	0.053630968	8
P25-Package management	0.16089290350220609	3	0.053630968	9
P30-Keep credentials safe	0.36670298299900694	1	0.122234328	2
consistency ratio = 0.05383				

Table 7. 18 reports the priorities and rankings of 24 DevSecOps practices in the "Technology" category, within three sub-categories: "Security Architecture", "Application Security", and "Operation Security", and they were given the same weight of 0.33. In Round Three, the three most important technology-related practices were: "P43 – Red team security drills" (ranked 13th in Round Two); "P55 – Big data and behavioural analytic techniques" (ranked 1st in Round Two); and "P54 – Integrate security issues within general bug tracker" (ranked 2nd in Round Two).

Table 7. 18 - AHP results for practices in "Technology" category

Practices in Technology (Weight on Architecture: 0.33)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P33-Automate tools and security processes	0.025027813582196509	6	0.008342605	20
P34-Security-as-Code	0.016696306435258119	9	0.005565435	23
P39-Cloud security	0.016696306435258119	10	0.005565435	24
P40-Container security	0.025027813582196509	7	0.008342605	21
P49-Immutable-as-Code	0.025027813582196509	8	0.008342605	22
P50-Policy-as-Code	0.04228434818937464	5	0.014094783	16
P51-Design-as-Code	0.1125330155712036	4	0.037511005	9
P52-Compliance-as-Code	0.21638074687198461	2	0.072126916	6
P53-Microservices security	0.21638074687198461	3	0.072126916	7
P54-Integrate security issues within your general bug tracker	0.30394508887834693	1	0.10131503	3
consistency ratio = 0.03976				
Normalised in Technology (Weight on Application: 0.33)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P37-Secure coding	0.08946994758527986	5	0.029823316	12
P42-Software Composition Analysis	0.028686304202513804	7	0.009562101	18
P44-Fault injection (chaos engineering)	0.028686304202513804	8	0.009562101	19
P45-RASP	0.051395672050449183	6	0.017131891	15
P46-SAST	0.10279126991502208	4	0.034263757	11
P47-DAST	0.29808961606458723	1	0.099363205	4
P48-IAST	0.29808961606458723	2	0.099363205	5
NP03-Software Bill of Materials (SBOM)	0.10279126991504678	3	0.034263757	10
consistency ratio = 0.00370				
Practices in Technology (Weight on Operation: 0.33)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
P35-Threat modelling	0.032519925881579204	6	0.010839975	17
P36-Continuous monitoring	0.058819178436442278	4	0.019606393	13
P38-Penetration testing	0.058819178436442278	5	0.019606393	14
P41-Sensitive information scan	0.17645744647239736	3	0.058819149	8
P43-Red team security drills	0.33669213538656945	1	0.112230712	1
P55-Big data and behavioural analytic techniques	0.33669213538656945	2	0.112230712	2
consistency ratio = 0.00221				

Table 7. 19 presents the AHP pairwise results for the five business practices of DevSecOps. There are no sub-categories due to the small volume of practices in this category. In Round Three, the three most important business-related practices were: "P60 – Availability and business continuity management" (ranked 2nd in Round Two); "P59 – Linear scalability and affordable cost" (ranked 5th in Round Two); and "P58 – Business-driven security" (ranked 1st in Round Two).

Table 7. 19 - AHP results for practices in "Business" category

Practices in Business	Normalised Value within Sub-category	Ranking within Sub-category	Normalised Value within Category	Ranking within Category
P56-Consumable security services with APIs	N/A	N/A	0.033366036931069307	5
P57-Separation of duties	N/A	N/A	0.053211403763870402	4
P58-Business-driven security	N/A	N/A	0.15181740733937329	3
P59-Linear scalability and affordable cost	N/A	N/A	0.28397664176042642	2
P60-Availability and business continuity management	N/A	N/A	0.47762851020526059	1
consistency ratio = 0.02052				

Overall priorities and rankings:

To derive the overall priorities/weights of 63 revised DevSecOps practices, the normalised value of each practices (in **Table 7. 16**, **Table 7. 17**, **Table 7. 18**, and **Table 7. 19**) was multiplied by the normalised value of its corresponding category (in **Table 7. 15**), according to **Equation (5)**: $\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^{T}\mathbf{1} = 1 \end{cases}$ (Page 73).

The overall priorities of 63 revised DevSecOps practices are provided in **Table 7. 20**, ordered by ranking, with higher priorities indicating greater importance.

Table 7. 20 - Overall priorities and rankings of revised DevSecOps practices

Overall Ranking	Overall Priority	DevSecOps Practices	Practices Category
1	0.135913107	P60-Availability and business continuity management	Business
2	0.092111615	P14-Continuous improvement mindset	Organisation, People & Culture
3	0.080807881	P59-Linear scalability and affordable cost	Business
4	0.063631482	P15-Leadership support	Organisation, People & Culture
5	0.046055825	P05-Training, learning and education for security	Organisation, People & Culture
6	0.046055819	P03-Shared and collective responsibility for security	Organisation, People & Culture
7	0.043200888	P58-Business-driven security	Business
8	0.042540854	NP02-Privacy-by-Design	Process Capabilities
9	0.025861208	P12-Commitment and agreement	Organisation, People & Culture
10	0.025861208	P13-Enhance transparency	Organisation, People & Culture
11	0.024032403	P01-Cultural shift to security	Organisation, People & Culture
12	0.024032403	P02-Improving collaboration, communication and cooperation	Organisation, People & Culture
13	0.022400655	P30-Keep credentials safe	Process Capabilities
14	0.022023417	P06-Security champions	Organisation, People & Culture
15	0.022023417	P09-Be reactive and responsive	Organisation, People & Culture

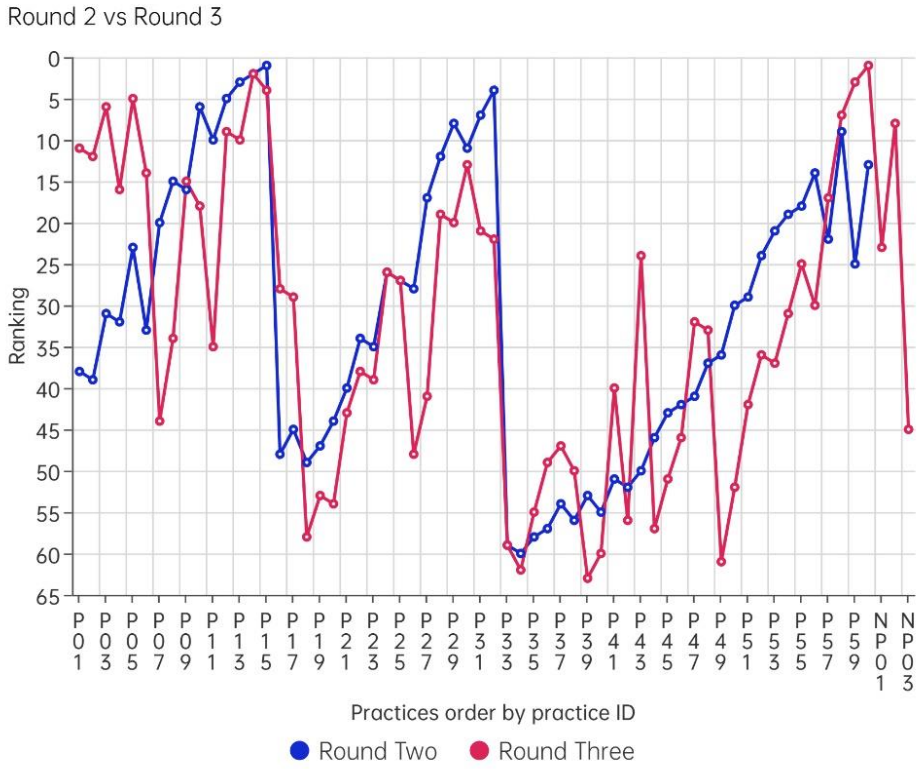
16	0.016180289	P04-Shared knowledge	Organisation, People & Culture
17	0.015141741	P57-Separation of duties	Business
18	0.013601709	P10-Shameless retrospectives	Organisation, People & Culture
19	0.010818102	P28-Define security requirements	Process Capabilities
20	0.010818102	P29-Security review and evaluation	Process Capabilities
21	0.010818102	P31-Common weaknesses enumeration	Process Capabilities
22	0.010818102	P32-Hybrid life cycles with data-security focus	Process Capabilities
23	0.010818102	NP01-Scrum for DevSecOps	Process Capabilities
24	0.010483368	P43-Red team security drills	Technology
25	0.010483368	P55-Big data and behavioural analytic techniques	Technology
26	0.009828408	P24-Configuration management	Process Capabilities
27	0.009828408	P25-Package management	Process Capabilities
28	0.009624856	P16-Shifting security to the left (early)	Process Capabilities
29	0.009624856	P17-Security-by-Design	Process Capabilities
30	0.009494579	P56-Consumable security services with APIs	Business
31	0.009463744	P54-Integrate security issues within your general bug tracker	Technology
32	0.009281426	P47-DAST	Technology
33	0.009281426	P48-IAST	Technology
34	0.007047435	P08-Continuous feedback loop	Organisation, People & Culture
35	0.007047435	P11-Impose security policies	Organisation, People & Culture
36	0.006737309	P52-Compliance-as-Code	Technology
37	0.006737309	P53-Microservices security	Technology
38	0.006173282	P22-Vulnerability and incident management	Process Capabilities
39	0.006173282	P23-Privilege management	Process Capabilities
40	0.005494243	P41-Sensitive information scan	Technology
41	0.004039304	P27-Software process maturity	Process Capabilities
42	0.003503869	P51-Design-as-Code	Technology
43	0.003473468	P21-Risk management	Process Capabilities
44	0.003207116	P07-Recruiting success	Organisation, People & Culture
45	0.003200546	NP03-Software Bill of Materials (SBOM)	Technology
46	0.003200546	P46-SAST	Technology
47	0.002785769	P37-Secure coding	Technology
48	0.00225291	P26-Define metrics	Process Capabilities
49	0.001831415	P36-Continuous monitoring	Technology
50	0.001831415	P38-Penetration testing	Technology
51	0.001600274	P45-RASP	Technology
52	0.001316581	P50-Policy-as-Code	Technology
53	0.001161605	P19-Good documentation, logging and reporting	Process Capabilities
54	0.001161605	P20-Compliance control	Process Capabilities
55	0.001012552	P35-Threat modelling	Technology
56	0.000893187	P42-Software Composition Analysis	Technology
57	0.000893187	P44-Fault injection (chaos engineering)	Technology
58	0.000885932	P18-Increase the visibility	Process Capabilities
59	0.000779275	P33-Automate tools and security processes	Technology

60	0.000779275	P40-Container security	Technology
61	0.000779275	P49-Immutable-as-Code	Technology
62	0.000519862	P34-Security-as-Code	Technology
63	0.000519862	P39-Cloud security	Technology

7.2.2 Termination of the Survey for Practices Evaluation

Figure 7. 4 compares the overall rankings of DevSecOps practices between Rounds Two and Three. The practices are ordered by ID on the x-axis (only display the odd numbered practices due to page limit), and their rankings are on the y-axis. The blue line denotes the overall ranking in Round Two, and the red line denotes the overall ranking in Round Three.

Figure 7. 4 - Comparison of rankings of Practices between Round Two and Round Three



The blue line shows three similar peaks, indicating that in Round Two the ranking of practices increased consecutively across three categories (i.e., P01 – P15 in the "OPC" category, P16 – P32 in the "PC" category, and P33 – P55 in the "Technology" category). That was precisely what was caused by participants' recency bias in Round Two. The red line for Round Three, by contrast, has significantly mitigated this issue.

The rankings and weights of the four practice categories in Rounds Two and Three were close, except that the rankings of the "Business" and "Process Capabilities" categories were reversed. As shown in the above graph, there is no significant difference between the general trends of the red and blue lines, indicating that the rankings have achieved relative stability across two iterations of the Delphi survey. In addition to the stable ranking, no new DevSecOps practices were included in Round Three.

Furthermore, as shown previously in **Table 7. 15**, **Table 7. 16**, **Table 7. 17**, **Table 7. 18**, and **Table 7. 19**, all consistency ratios of the AHP comparisons in this round were in an acceptable range (between 0 and 0.1). Hence, based on the results of Rounds Two and Three, a conclusion can be drawn: the evaluation of the practices dimension of DevSecOps has been completed, and it will stop at Round Three, since all stopping criteria have been met.

7.3 RQ3 – Evaluation of DevSecOps Metrics

This section presents the results of the AHP pairwise comparisons from Questions 13 to 19 (refer to **Table 7. 2** on Page 193), assessing the importance of the 20 identified DevSecOps metrics within four categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology, and their associated sub-categories. The priorities and rankings of these DevSecOps metrics were calculated, discussed, and reported. A glossary of the identified metrics was provided to participants before this survey round, as it was recognised that DevSecOps metrics might not be as well-known as the challenges and practices of DevSecOps.

In addition to these AHP pairwise-comparison questions, an open-ended question (Question 20) was included to allow participants to provide additional DevSecOps metrics. The data and responses were collected, calculated, analysed, discussed, and reported using multiple tools (e.g., Qualtrics, SuperDecisions, and MS Excel spreadsheets) to partially address RQ3 and its associated sub-questions, specifically regarding the metrics dimension of DevSecOps.

All consistency ratios for the AHP comparisons on metrics were all in an acceptable range (between 0 and 0.1) (Saaty, 1982). The degree of dissent among participants, i.e., the value of the Coefficient of Variation (CV), was also in an acceptable range (between 0 and 0.5) (Dajani,

Sincoff, & Talley, 1979). Considering that both criteria were met and based on participants' cognition and feedback, a decision was made not to repeat the evaluation of the metrics in a new round. The entire Delphi survey was terminated at Round Three.

7.3.1 AHP Results of DevSecOps Metrics

The minimum number of AHP comparisons was used to assess the importance of 20 identified DevSecOps metrics. Once participants had completed the minimum number of comparisons, the researcher calculated the remaining comparison judgments.

Refer to **Table 7. 1**, Questions 13 – 19 were used to conduct AHP pairwise comparisons of the identified practices. Response data was collected via the Qualtrics online survey platform (Qualtrics, 2024). The extracted data were averaged to build a group compromise for evaluating AHP pairwise comparisons. The Consistency Ratio (CR) was calculated to assess the consistency of the AHP comparisons. The Coefficient of Variation (CV) was calculated to assess the degree of consensus or dissent among participants.

AHP comparison matrices:

Table 7. 21 presents the AHP comparison matrix for four categories: Organisation, People and Culture (OPC), Process Capabilities (PC), Technology, and Business.

Table 7. 21 - AHP comparison matrix for categories of metrics

Categories of Metrics	Business	OPC	PC	Technology
Business	1/1	1/1	2/1	1/1
OPC	1/1	1/1	2/1	2/1
PC	1/2	1/2	1/1	1/1
Technology	1/1	1/2	1/1	1/1

Table 7. 22 provides the AHP comparison matrix for the sub-categories newly added in Round Three to avoid the risks of high inconsistency and recency bias.

Table 7. 22 - AHP comparison matrix for sub-categories of metrics

Sub-categories in PC	Security Process	Security Management
Security Process	1/1	2/1
Security Management	1/2	1/1
Sub-categories in Technology	**Application Security**	**Operation Security**
Application Security	1/1	1/1
Operation Security	1/1	1/1

Next, **Table 7. 23**, **Table 7. 24**, **Table 7. 25**, and **Table 7. 26** present the AHP comparison matrices for DevSecOps metrics within their sub-category and category, using the minimum number of comparisons (Mu & Pereyra-Rojas, 2017). The "OPC" and "Business" categories each involved only one metric, so there was no AHP comparison matrix for them.

Table 7. 23 - AHP comparison matrix for metrics in "Security Process" sub-category, in "Process Capabilities" category

Practices in Process Capabilities (Security Process)	M03	M04	M08
M03	1/1	1/2	1/1
M04	2/1	1/1	2/1
M08	1/1	1/2	1/1

Table 7. 24 - AHP comparison matrix for metrics in "Security Management" sub-category, in "Process Capabilities" category

Practices in Process Capabilities (Security Management)	M02	M05	M06	M07
M02	1/1	1/1	1/2	1/1
M05	1/1	1/1	1/2	1/1
M06	2/1	2/1	1/1	2/1
M07	1/1	1/1	1/2	1/1

Table 7. 25 - AHP comparison matrix for metrics in "Application Security" sub-category, in "Technology" category

Practices in Technology (Application Security)	M11	M12	M14	M15	M16	M17
M11	1/1	1/2	1/4	1/8	0/1	0/1
M12	2/1	1/1	1/2	1/4	1/8	1/8
M14	4/1	2/1	1/1	1/2	1/4	1/4
M15	8/1	4/1	2/1	1/1	1/2	1/2
M16	16/1	8/1	4/1	2/1	1/1	1/1
M17	16/1	8/1	4/1	2/1	1/1	1/1

Table 7. 26 - AHP comparison matrix for metrics in "Operation Security" sub-category, in "Technology" category

Practices in Technology (Operation Security)	M09	M10	M13	M18	M19
M09	1/1	1/1	1/4	1/2	1/6
M10	1/1	1/1	1/4	1/2	1/6
M13	4/1	4/1	1/1	2/1	2/3
M18	2/1	2/1	1/2	1/1	1/3
M19	6/1	6/1	3/2	3/1	1/1

Priorities, rankings, and consistency ratios:

The AHP tool SuperDecisions (SuperDecisions, 2023) was employed to calculate and derive the priorities/weights and the consistency ratios for these AHP comparison matrices. **Table 7. 27** presents the AHP results for four categories of DevSecOps metrics, and **Table 7. 28** and **Table 7. 29** present the AHP results for practices within categories. The "OPC" and "Business" categories each involved only one metric, so there was no AHP comparison matrix for them.

Table 7. 27 shows that the most important category of DevSecOps practices was "Organisation, People and Culture" with a normalised value of approximately 0.34. (All results were rounded to two decimal places for reporting.) The "Business" category ranked second with a priority of 0.29, followed by "Technology" with a priority of 0.20. The least important category was "Process Capabilities", with the lowest priority of 0.17. The normalised and idealised values of sub-categories are also provided.

Table 7. 27 - AHP results for categories of metrics

Categories of Metrics	Normalised Value	Idealised Value	Ranking
Business	0.28792354452028202	0.85105979062711523	2
Organisation, People and Culture	0.33831177044343919	1.0	1
Process Capabilities (Sub-Categories: Security Process; Security Management)	0.16915588522171959 (Process: 0.67; Management: 0.33)	0.5 (Process: 1.0; Management: 0.5)	4
Technology (Sub-Categories: Application Security; Operation Security)	0.20460879981455921 (Application: 0.5; Operation: 0.5)	0.6047936184613678 (Application: 1.0; Operation: 1.0)	3
consistency ratio = 0.02271			

Table 7. 28 reports the priorities and rankings of seven DevSecOps metrics in the "Process Capabilities" category, within its two sub-categories: "Security Process" and "Security Management". "Security Process" was ranked higher than "Security Management", with the weights of 0.67 and 0.33, respectively. To obtain the normalised value of each metric within a category, its normalised value within a sub-category was multiplied by the weight of the sub-category. The three most important PC-related metrics were all in the sub-category of "Security Process", respectively: "M04 – Security review performance"; "M03 – Time spent correcting mistakes in each category"; and "M08 – Number of continuous delivery cycles per month".

Table 7. 28 - AHP results for metrics in "Process Capabilities" category

Metrics in Process Capabilities (Weight on Security Process: 0.67)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
M03-Time spent correcting mistakes in each category	0.25	2	0.166666667	2
M04-Security review performance	0.5	1	0.333333333	1
M08-Number of continuous delivery cycles per month	0.25	3	0.166666667	3
consistency ratio = 0				
Metrics in Process Capabilities (Weight on Security Management: 0.33)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
M02-Top vulnerability	0.2	2	0.066666667	5
M05-SLA performance	0.2	3	0.066666667	6
M06-Critical risk profiling	0.4	1	0.133333333	4
M07-Point of risk per device	0.2	4	0.066666667	7
consistency ratio = 0				

Table 7. 29 presents the priorities and rankings of eleven DevSecOps metrics in the "Technology" category, within two sub-categories: "Application Security" and "Operation Security", with equal weight of 0.5. The three most important technology-related metrics were: "M19 – Number of issues during red team drills"; "M16 – Whether automated testing covers security"; and "M17 – Use of preapproved libraries, packages, and tools".

Table 7. 29 - AHP results for metrics in "Technology" category

Metrics in Technology (Weight on Application: 0.5)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
M11-Defect density	0.02780401076157801	6	0.013902005	11
M12-Defect burn rate	0.045064151028743997	5	0.022532076	10
M14-Security test pass rate	0.090128302057487994	4	0.045064151	7
M15-Code scanning detection rate	0.18025660411497599	3	0.090128302	5
M16-Whether automated testing covers security	0.32837346601860701	1	0.164186733	2
M17-Use of preapproved libraries, packages, tools	0.32837346601860701	2	0.164186733	3
consistency ratio = 0.00891				
Metrics in Technology (Weight on Operation: 0.5)	**Normalised Value within Sub-category**	**Ranking within Sub-category**	**Normalised Value within Category**	**Ranking within Category**
M09-Number of adversaries per application	0.071765431549833808	4	0.035882716	8
M10-Adversary return rate	0.071765431549833808	5	0.035882716	9
M13-Penetration test pass rate	0.31358025616468571	2	0.156790128	4
M18-Use of SAFe DevOps Health Radar	0.1435307835437595	3	0.071765392	6
M19-Number of issues during red team drills	0.39935809719188731	1	0.199679049	1
consistency ratio = 0.00443				

Overall priorities and rankings:

To derive the overall priorities/weights of the identified DevSecOps metrics, the normalised value of each metrics (in **Table 7. 28** and **Table 7. 29**) was multiplied by the normalised value of its corresponding category (in **Table 7. 27**), according to **Equation (5)**: $\begin{cases} \mathbf{Aw} = \lambda_{max}\mathbf{w} \\ \mathbf{w}^{T}\mathbf{1} = 1 \end{cases}$ (Page 73).

The overall priorities of 20 metrics are provided in **Table 7. 30**, ordered by ranking, with higher priorities indicating greater importance.

Table 7. 30 - Overall priorities and rankings of DevSecOps metrics

Overall Ranking	Overall Priority	DevSecOps Metrics	Metrics Category
1	0.33831177	M01-Security-trained rate	Organisation, People & Culture
2	0.287923545	M20-Business metrics	Business
3	0.056385295	M04-Security review performance	Process Capabilities
4	0.04085609	M19-Number of issues during red teaming drills	Technology
5	0.03359405	M16-Whether automated testing covers security requirements	Technology
6	0.03359405	M17-Use of preapproved libraries, packages, tool chains, and processes	Technology
7	0.03208064	M13-Penetration test pass rate	Technology
8	0.028192648	M03-Time spent correcting mistakes in each category	Process Capabilities
9	0.028192648	M08-Number of continuous delivery cycles per month	Process Capabilities
10	0.022554118	M06-Critical risk profiling	Process Capabilities
11	0.018441044	M15-Code scanning detection rate	Technology
12	0.014683831	M18-Use of SAFe DevOps Health Radar	Technology
13	0.011277059	M02-Top vulnerability	Process Capabilities
14	0.011277059	M05-SLA performance	Process Capabilities
15	0.011277059	M07-Point of risk per device	Process Capabilities
16	0.009220522	M14-Security test pass rate	Technology
17	0.007341919	M09-Number of adversaries per application	Technology
18	0.007341919	M10-Adversary return rate	Technology
19	0.004610261	M12-Defect burn rate	Technology
20	0.002844473	M11-Defect density	Technology

7.3.2 Open-ended Question Results – Additional Metrics

In addition to the AHP pairwise-comparison questions, Round Three included an open-ended

question (Question 20) that allowed participants to provide comments on additional DevSecOps metrics. As predicted, participants' opinions on DevSecOps metrics were uncertain and inconclusive. Hence, of the 15 participants, only 4 expressed their viewpoints on this question.

Unlike Rounds One and Two, where participants directly added new challenges and practices, Round Three saw participants make suggestions about metrics rather than provide specific answers. After performing a short Thematic Analysis (TA) process, as listed in **Table 7. 31**, a new OPC-related metric, "Collaboration between DevOps/Dev and security teams", and a new business-related metric, "Security costs and benefits", were included.

Table 7. 31 - Additional new metrics

Comment/Code	Handling	New Theme/Metric	Category
"Devops/Dev teams' collaboration specifically"	Included	Collaboration between DevOps/Dev and security teams, i.e., the degree of contribution of the DevOps/Dev team to security	Organisation, People & Culture
"Dev/sec collaboration"	Included		
"Need to consider, to what extent do reviews in development help avoid incidents in production?"	Included		Organisation, People & Culture
"Consider referring to ISO27004 or NIST SP 800-55 for additional metrics."	Included on the tools list	Considered both as security measurement tools	Process Capabilities
"My concern is that things like M04 only look at cost and ignore benefit."	Included	Security costs and benefits.	Business
"Nothing specific, but try to consider the benefit side for measured investments. Try to justify investments in terms of quantified value."	Included		

Good collaboration, communication, and cooperation between the DevOps/Developers team and the security team are key factors in the success of DevSecOps implementation. Therefore, A new metric was collected and included: "Collaboration between DevOps/Dev and security teams." It is not trivial to measure this quantitatively, as it appears to be more of a qualitative metric. A participant (Pa10) provided us with an advisable proposal, *"Need to consider, to what extent do reviews in development help avoid incidents in production"*. His idea aligns with a recent study by Caniglia et al. (2025), which introduces the Collaboration DevSec Index (CDSI) to measure developers' contributions to security quantitatively. The number of tests represents the value of CDSI, ranging from a negative result after the first test to a positive result. CDSI ideally should tend to zero; a smaller value indicates fewer iterations of remediation actions carried out by

developers and, conversely, a greater contribution of each remediation action to passing the security test (Caniglia et al., 2025).

Participant Pa10 also mentioned that both benefits and costs should be considered as business-related metrics. The previous MLR study (Zhao, Clear, & Lal, 2024b) has underlined an absence of specific business-related metrics of DevSecOps. Only a few general business metrics, such as Return on Investment (ROI) and Business Value Increment (BVI) (Nisha & Khandebharad, 2022), have been identified. On the other hand, all DevSecOps metrics have business impacts on time, costs, and profits (Caniglia et al., 2025). For example, faster security reviews and shorter remediation processes accelerate the entire SDLC, resulting in shorter time-to-market, which in turn fosters customer trust and enhances profit potential. Predicted vulnerabilities and lower defect density enable the proactive handling and mitigation of recurring bugs, thereby stretching the security budget. A more exhaustive coverage of automated tests reduces the need for human resources and effort to address late-stage defects, though it requires high upfront costs (Caniglia et al., 2025).

To address the gap in business-related DevSecOps metrics, Caniglia et al. (2025) proposed the Framework of Business Index Concerning Security (FOBICS), which connects various metrics to quantify DevSecOps performance. They validated their framework through two projects. Almost all their developed metrics align with the identified metrics in this research, except for a few whose main objective is not purely security-related. For instance, the FOBICS framework unifies different types of security metrics but excludes those focused solely on DevOps. Regarding the cons, the authors noted that multiple factors contribute to the FOBICS framework, meaning a higher index value is not necessarily attributable to a specific sector, but could be due to one or more sectors, or all sectors. Another issue is that using the novel framework entails additional business costs for monitoring data collection and employee training (Caniglia et al., 2025).

All things considered, for the business-related metrics of DevSecOps, a decision was made by the researcher to remain with the traditional and generic metrics, e.g., calculating Return on Investment (ROI) for financial attributes to evaluate the balance between security costs and benefits, or using Business Value Increment (BVI) to judge the success of DevSecOps adoption

(Nisha & Khandebharad, 2022; Zhao, Clear, & Lal, 2024b). In practice, organisations adopt DevSecOps to varying extents, driven by different business concerns (NCCoE, 2025). It is more straightforward for them to select and tailor metrics separately, rather than bundling all metrics together, as in the FOBICS framework by Caniglia et al. (2025), even though they state that it is "clear and easy-to-interpret" to unify different types of metrics and measurements.

In addition to the OPC-related and business-related metrics, one participant (Pa8) recommended to "consider referring to ISO27004 or NIST SP 800-55 for additional metrics". ISO/IEC 27004 is an international standard for information security, while NIST SP 800-55 serves as a measurement guide for information security.

Those standards or documents are not specific metrics, and not applicable to the definition of "Metrics", which is given in **Table 3. 4**, Chapter 3 (Page 45), "the means to track progress, facilitate decision-making, and improve performance of DevSecOps practices by measuring implementation, effectiveness, efficiency, and impact". By contrast, they guide organisations on how to define metrics and measure performance, thereby being more like a tool that supports the practice "P26 – Define metrics", rather than metrics themselves. Hence, ISO27004 and NIST SP 800-55 were included as "Security measurement tools" and added to the DevSecOps tool list, so organisations could use them to select, define, or tailor their metrics as needed.

7.4 RQ4 – DevSecOps Metrics in GSE Contexts

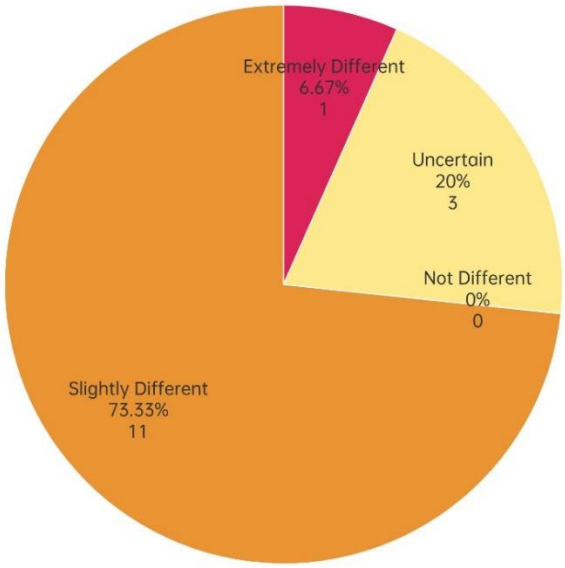
This section presents the findings from participants' responses to a multiple-choice question (Question 21) and an open-ended question (Question 22) on the differences between local and global DevSecOps metrics, to address RQ4 and its associated sub-questions.

To answer the sub-question 4.1, a multiple-choice question (Question 21) was included in the Round Three Delphi survey: "How do DevSecOps metrics differ in local and global settings?" This question included four options: "Extremely Different", "Slightly Different", "Not Different", and "Uncertain" for participants to choose the most relevant one based on their expertise and experience.

Figure 7. 5 depicts the proportions for these four options. 77.33% of the participants opted for

"Slightly Different", which ranked first. "Uncertain" ranked second with 20%, followed by "Extremely Different" with 6.67%. No participants opted for "Not Different".

Figure 7. 5 - Opinions on "How do DevSecOps metrics differ in local and global settings?"



To address sub-question 4.2, the Round Three survey included an open-ended question (Question 22) to gather participants' opinions on the differences between local and global DevSecOps metrics. However, no responses were received to this open-ended question.

The same as the results of Rounds One and Two, this result is again the proof of the MLR findings: "there is a notable absence of the global dimension of DevSecOps in both of the white and grey literature" (Zhao, Clear, & Lal, 2024b). Four assumptions have been presented in Section 3.3.2, Chapter 3 (Page 60).

The first assumption is that there are no distinguishing characteristics of DevSecOps, whether it is adopted in a local or global setting. Assumption two may be that security is typically a centralised and control-oriented function in organisations, so global aspects are not prominent. The third assumption is that there is a research gap between DevSecOps and GSE, and the fourth one is due to an improper search string.

This result corroborates the first and second assumptions. Most participants believe that DevSecOps metrics differ slightly between local and global settings, which may lead to insufficient attention to research on global DevSecOps, thereby supporting the third assumption.

7.5 Dissent Analysis for Round Three

This section presents the dissent analysis for Round Three to examine the degree of consensus or dissent among participants regarding the AHP comparison results and global DevSecOps metrics, using both quantitative and qualitative methods.

7.5.1 Dissent on AHP Results in Round Three

To address *Sub-question 3.2: "Will the experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"*, a dissent analysis was conducted to discuss participants' agreements and disagreements on AHP comparison results, based on three levels:

- Individual level: to analyse the consensus or dissent among each participant regardless of the participant's position and role.
- Groups level: to analyse the consensus or dissent between the four participants' groups (i.e., Group AT, Group AM, Group IT, and Group IM).
- Roles level: to analyse the consensus or dissent between four participants' roles (i.e., Academic, Industrial, Managerial, and Technical).

According to the four-quadrant division for participants grouping in **Figure 7. 3** (Page 195), the datasets were merged as follows:

- Two datasets for Groups AM and AT were merged into a single dataset for the Academic role.
- Two datasets for Groups IM and IT were merged into a single dataset for the Industrial role.
- Two datasets for Groups AT and IT were merged into a single dataset for the Technical role.
- Two datasets for Groups AT and IT were merged into a single dataset for the Managerial role.

The previous experience in Rounds One and Two supports that the AHP method is most suitable for small to medium-sized groups (e.g., 8 to 20 participants), rather than for individuals, tiny groups (fewer than five participants), or relatively large groups (over 30 participants). Using AHP with fewer than five participants may lead to inconsistencies in the comparisons. Using AHP with

large groups may lead to overly-compromised comparison results (i.e., converging to the "Equally important" scale), narrowing the ranking gaps and rendering the AHP method less meritorious.

As discussed in Section 5.4.1, Chapter 5 (Page 127), in the second and third rounds of the Delphi survey, the dissent analysis for AHP results would be conducted only based on participants' roles, rather than small groups. This is also more satisfactory for the content of *Sub-question 3.2: "Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?"* In addition, Coefficients of variation (CVs) were calculated to evaluate the degree of dissent at the individual level, thereby analysing consensus or dissent for each participant regardless of position or role, to determine whether participants with similar roles would have more consensual opinions or whether individual variation is the primary factor.

Dissent analysis based on individuals:

Coefficient of variations (CVs) of the AHP comparison results were calculated to evaluate the degree of dissent among participants at the individual level, to verify whether participants with similar roles would have more consensual opinions, or if individual variations are the primary factor. CV is defined as the ratio of the standard deviation to the arithmetic mean, and a higher value indicates greater dissent among participants (Everitt, 2006). In this research, the acceptable value of CV was defined to be less than 0.5 (Dajani, Sincoff, & Talley, 1979).

Rather than listing all 88 values of CVs in Round Three, in **Table 7. 32**, the average value of overall CVs (i.e., CVs for 15 participants in Round Three) and the CVs in four groups and four roles was compared, to investigate if participants in the same group or with same roles had more consensual results, in two dimensions of practices and metrics of DevSecOps. All the CV values were between 0 and 0.5, i.e., in an acceptable range (Dajani, Sincoff, & Talley, 1979).

Table 7. 32 - Coefficient of variations in Round Three

Overall CVs (average value) for the evaluation of revised practices	**CVs in 4 Groups (average value)**			
	Group AT: 0.32	Group AM: 0.18	Group IT: 0.35	Group IM: 0.32
0.36	**CVs in 4 Roles (average value)**			
	Technical: 0.38	Managerial: 0.31	Academic: 0.32	Industrial: 0.37
Overall CVs (average value) for the evaluation of metrics	**CVs in 4 Groups (average value)**			
	Group AT: 0.31	Group AM: 0.17	Group IT: 0.41	Group IM: 0.22
0.34	**CVs in 4 Roles (average value)**			
	Technical: 0.41	Managerial: 0.22	Academic: 0.29	Industrial: 0.34

In Round Three, the mean of overall CVs for the evaluation of revised practices was 0.36, the CVs for four participants' groups were less than this value, respectively were: Group Academic Technical (AT) – 0.32; Group Academic Managerial (AM) – 0.18; Group Industrial Technical (IT) – 0.35; and Group Industrial Managerial (IM) – 0.32, indicating that the participants in the same groups held more consensual opinions on the prioritisation of DevSecOps practices. If we compared CVs based on participants' roles rather than groups, managerial participants and academic participants held more consensual opinions, with CVs of 0.31 and 0.22, respectively, below the overall value of 0.36. In contrast, technical roles and industrial roles elicited more dissenting opinions, with CVs of 0.38 and 0.37, respectively, slightly exceeding the overall value of 0.36. In other words, participants in technical or industrial roles showed slightly higher levels of dissent when evaluating the importance of DevSecOps practices than other participants.

The mean of overall CVs for the evaluation of DevSecOps metrics was 0.34. The CVs for the three participant groups were lower than this value: Group Academic Technical (AT) – 0.31; Group Academic Managerial (AM) – 0.17; and Group Industrial Managerial (IM) – 0.22. Group Industrial Technical (IT) reported the highest CV of 0.41, greater than the overall value of 0.34. If CVs were compared based on participants' roles rather than their groups, managerial participants and academic participants held more consensual opinions, with CVs of 0.22 and 0.29, respectively, below the overall value of 0.36. In contrast, technical roles and industrial roles had more dissenting opinions, with CVs of 0.41 and 0.34, respectively, which are slightly higher or equal to the overall value of 0.34. The results showed that participants in technical or industrial roles, particularly those in technical positions, expressed more dissenting opinions on prioritising DevSecOps metrics compared to other participants when rating and ranking their importance.

In retrospect, the CVs across all three iterations of the Delphi study showed a gradual decline, from over 0.45 to below 0.35. As the survey progressed, the adverse factors affecting CVs, e.g., participants' unfamiliarity with survey methods in Round One and recency bias in Round Two, have been mitigated or eliminated, so that the degree of dissents has dropped to a credible value, i.e., approximately 0.35, which could represent the proper level of the dissents among participants in the entire Delphi study.

Dissent analysis based on roles:

To address *Sub-question 3.2: "Will the experts have dissents on the prioritisation due to their different roles, e.g., academic, industrial, technical, and managerial?"*, a dissent analysis was performed based on participants' roles.

As shown in **Figure 7. 6**, three roles of participants, i.e., Academic, Technical, and Managerial roles, reached a consensus that they rated "Organisation, People and Culture" as the most important category of DevSecOps practices. In contrast, Industrial roles rated "Business" as the most important, slightly higher than "Organisation, People and Culture". Technical and Managerial roles ranked "Business" second, consistent with the overall result, while the Academic role ranked it third. For the "Process Capabilities" category, three Industrial, Technical, and Managerial roles ranked it third, consistent with the overall ranking, while the Academic roles rated it as the least important. For the "Technology" category, the Academic roles ranked second, ahead of "Business" and "Process Capabilities"; the other roles rated it as the least important category, consistent with the overall result.

Figure 7. 6 - Round Three AHP results for categories of practices based on roles

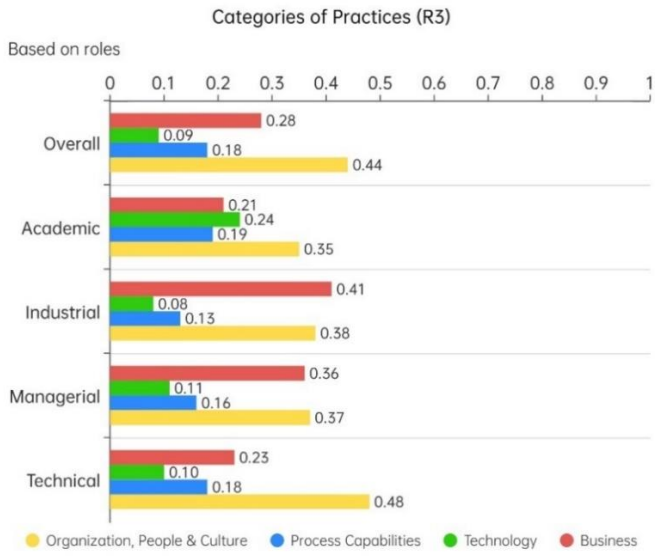


Figure 7. 7 compares the rankings of practices' categories in Rounds Two and Three, based on participants' roles. It shows that most priorities and rankings of categories have achieved relatively stable results. A significant difference was that in Round Three, the "Business" category

ranked much higher than it had in Round Two.

Figure 7. 7 - Dissents on categories of practices in Round Two and Round Three

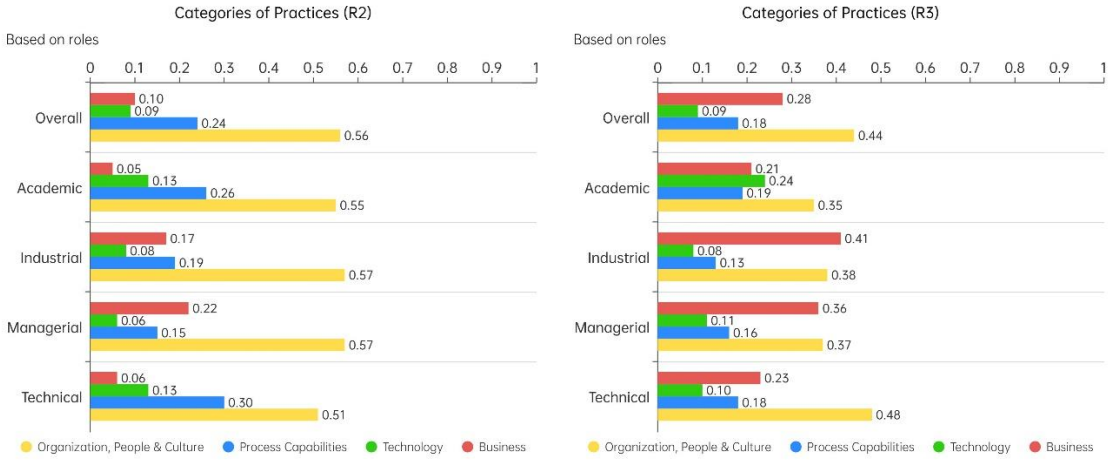


Figure 7. 8 compares the overall ranking of DevSecOps practices with rankings based on four participants' roles, indicating significant dissent. The red line represents the overall ranking of 63 revised DevSecOps practices; the other four coloured lines represent the rankings given by four participants' roles: Academic, Industrial, Managerial, and Technical roles.

Figure 7. 8 - Round Three dissents on rankings of DevSecOps practices based on roles

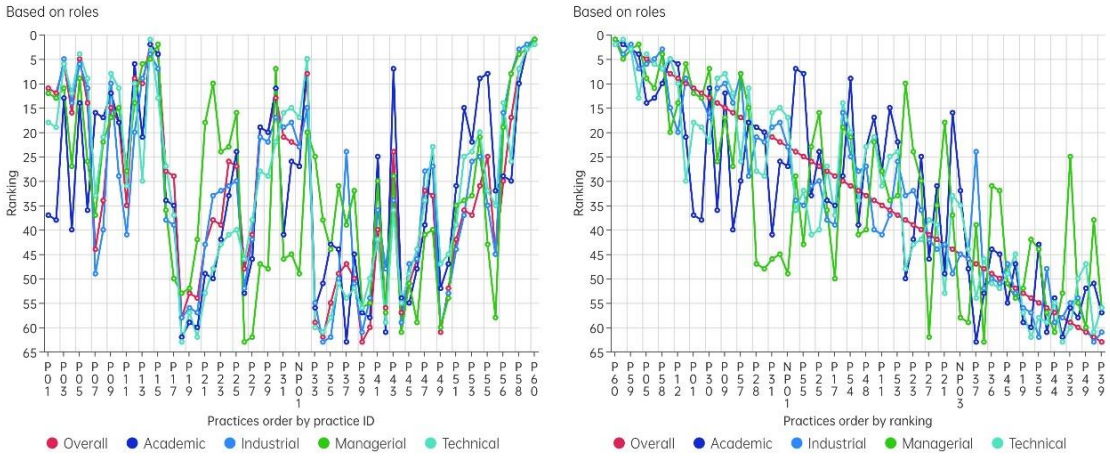


Figure 7. 8 presents two separate viewpoints. On the left-hand side, 63 revised practices are ordered by their identifiers within categories, showing the differences in rankings within each category. The line graph on the right-hand side ranks DevSecOps practices from first to 63rd place, showing the differences more visually. The further from the red line, the more different the

rankings. For example, the Academic role (purple line) and the Managerial role (green line) have more dissenting opinions than the other roles.

Figure 7. 9 indicates a significant difference in the rankings of metrics' categories given by four participants' roles. Only the result of Industrial role was consistent with the overall result; the rankings from highest to lowest were: "Organisation, People and Culture", "Business", "Process Capabilities", and "Technology". Besides, the other three participants' roles all gave dissenting opinions on the importance of the metrics' categories.

Figure 7. 9 - Round Three AHP results for categories of metrics based on roles

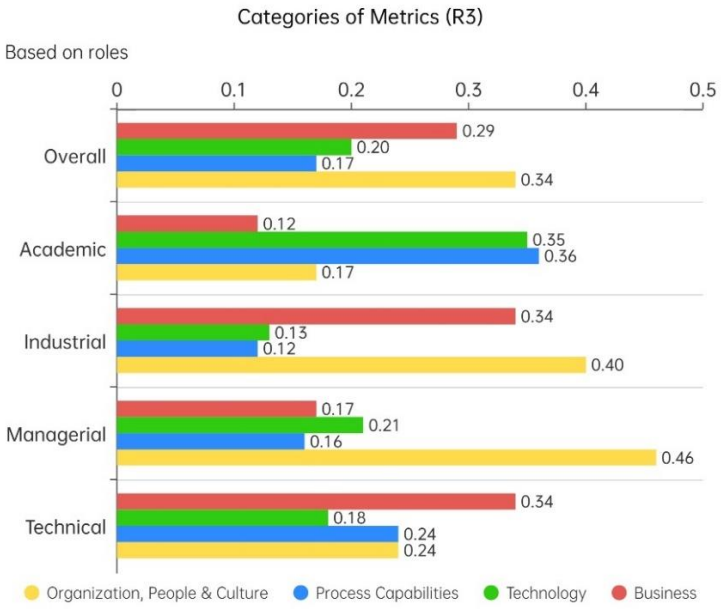
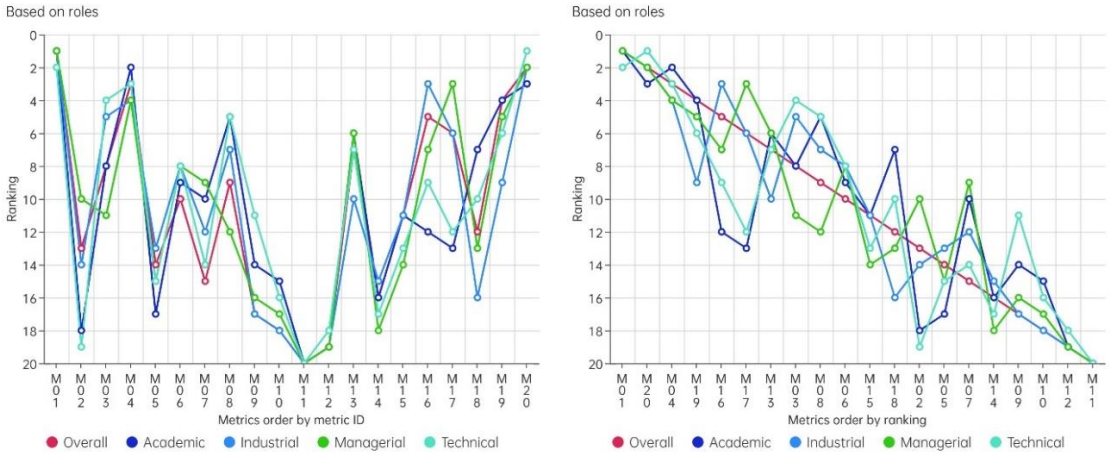


Figure 7. 10 compares the overall ranking of the identified DevSecOps metrics with rankings based on the four participants' roles. On the left-hand side, 20 metrics are ordered by their identifiers within categories, showing that the rankings given by four roles were relatively consistent within each category. Thus, the primary factor behind the differing rankings of DevSecOps metrics was variation across their respective categories. As discussed above in Section 7.3, participants' opinions on DevSecOps metrics were much more uncertain and inconclusive than those on challenges and practices.

Figure 7. 10 - Round Three dissents on rankings of DevSecOps metrics based on roles



By summing up the AHP results of Round Three and comparing with the results of Round Two, for the evaluation of DevSecOps practices and metrics, it could be concluded that: after the second and third iterations of the Delphi study, the participants with similar roles had relatively consistent opinions on the prioritisation of DevSecOps practices and DevSecOps metrics; whereas the participants with different roles had significantly dissenting opinions.

7.5.2 Dissent on Global DevSecOps Metrics in Round Three

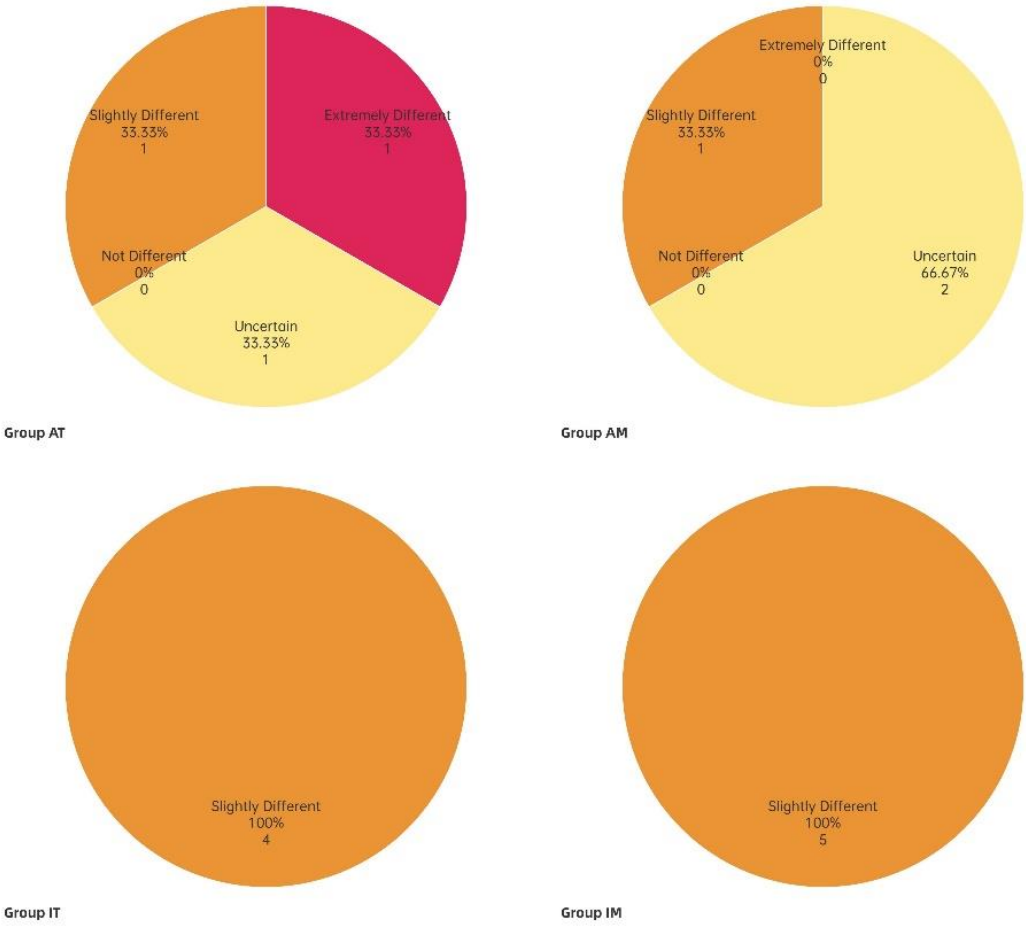
As shown in **Figure 7. 5** in Section 7.4 (Page 218), eleven participants (73.33%) believed that DevSecOps metrics were "Slightly Different" (ranked first) between local and global settings.

The four pie charts in **Figure 7. 11** show the responses of the four participants' groups. All participants in the Group Industrial Technical (IT) and Group Industrial Managerial (IM) opted for the "Slightly Different" option. In other words, all nine industrial participants believe that DevSecOps metrics are "Slightly Different" in these two settings.

Two academic participants also opted for "Slightly Different," with one in the Group Academic Technical (AT) and the other in the Group Academic Managerial (AM). However, three other academic participants selected "Uncertain", and one participant in the Group AT selected "Extremely Different" but did not provide any supporting responses for "what the difference is".

To sum up, participants have reached a consensus that "DevSecOps metrics differ slightly between local and global contexts; these differences in metrics must be well understood to enable metrics to be adopted and practised accordingly".

Figure 7. 11 - Opinions on "How do DevSecOps practices differ in local and global settings?" in four groups



7.6 Chapter Summary

This chapter presents the results of Round Three of the Delphi survey (the final round). The results show that the saturation point for the Delphi investigation has been reached, and any additional Delphi survey rounds will yield limited information for further enhancing the DevSecOps CPTM Model. A total of 15 participants were involved in Round Three, down from the 18 in Round One. Importantly, one academic expert who missed Round Two returned in Round Three.

Round Three was conducted to evaluate the importance of 63 DevSecOps practices and 20 DevSecOps metrics using AHP pairwise comparisons across four categories: Business, OPC (Organisation, People & Culture), PC (Process Capabilities), and Technology, and their associated sub-categories. The priorities and rankings of these practices and metrics were collected,

calculated, and reported using multiple tools, including Qualtrics, SuperDecisions, and Microsoft Excel spreadsheets. Consistency ratios were evaluated to ensure that the AHP results fell within an acceptable range. The two issues identified in Round Two, high inconsistency and recency bias, were mitigated in Round Three.

In addition to AHP pairwise comparisons, the Round Three survey responses enabled the identification of two additional DevSecOps metrics. Finally, analysis of participants' responses on the differences between local and global DevSecOps metrics showed a consensus that DevSecOps metrics differ slightly across the two contexts.

A dissent analysis was conducted to evaluate participants' agreement or disagreement with the AHP comparison results and their global DevSecOps opinions, using both quantitative and qualitative methods. Based on the results of Rounds Two and Three, a solid conclusion could be drawn: participants with similar roles had relatively consistent opinions on the prioritisation of DevSecOps practices and DevSecOps metrics, whereas participants with different roles had significantly dissenting opinions. Regarding the difference between local and global DevSecOps metrics, participants reached consensus that the two contexts are "Slightly Different" and acknowledged that global DevSecOps has not been well recognised.

To summarise, the evaluation of DevSecOps challenges has been fully completed based on the results of Rounds One and Two; the evaluation of DevSecOps practices has been completed based on the results of Rounds Two and Three. For the evaluation of DevSecOps metrics, participants' opinions were less certain and more inconclusive than for the results of challenges and practices. After analysing the results and considering participants' cognition and feedback, it was decided not to repeat the evaluation of metrics with a new round.

According to Beiderbeck et al. (2021), the stopping criteria were defined in terms of time, participants, and consensus, with the stability of responses also considered an additional factor. After carefully evaluating all stopping criteria, e.g., time constraint, participants' engagement and feedback, and the degree of consensus or stability of responses, the entire Delphi survey was terminated at Round Three.

The findings from Rounds One to Three have mostly answered RQ3, RQ4, and associated sub-questions, i.e., have sorted out the dimensions of challenges, practices, and metrics. The

DevSecOps tools were not assessed using the Delphi-AHP method for some reasons:

- Tools' themes overlapped with some practices in the "Technology" category.
- Making pairwise comparisons is the fundamental principle of the AHP method; however, tools of different types are incommensurable. For instance, a security testing tool and a communication tool are not comparable in perceived importance, as they serve different purposes.
- It is unrealistic to assume that participants have experience in using all kinds of tools across the DevSecOps lifecycle, even though they are experts.

Hence, tools were not evaluated using the Delphi survey method. Instead, they were validated, complemented, and revised by referring to two related works. The first one is a recently published MLR study (Prates & Pereira, 2025) that focuses on identifying tools used to support DevSecOps practices. The other is an industry publication, "Periodic Table of DevSecOps Tools" from Digital.ai (2025), which identifies top tools across the DevSecOps lifecycle. These two related publications provide current, comprehensive lists of DevSecOps tools but use different taxonomies, reflecting the latest academic and industry perspectives. The evaluation of DevSecOps tools is discussed in Chapter 8.

Based on the survey results of three iterations, Chapter 8 summarises the findings and provides a detailed discussion on the four aspects of DevSecOps: Challenges, Practices, Metrics, and Tools, to answer the research questions and finalise the DevSecOps CPTM Model (Version 2.0).

8 Chapter 8: Summary and Discussion

This chapter summarises and discusses the findings of the Delphi-AHP study, addressing the research questions in Sections 8.1 and 8.2. Based on these findings, the updated conceptual framework titled "DevSecOps CPTM Model (Version 2.0)" is presented in Section 8.3, which includes the model design and upgrades. An illustrative scenario is provided as a worked example to guide approaches to navigation of the model. Section 8.4 concludes this chapter. The following section moves on to Research Question Three.

8.1 RQ3 – Identification and Prioritisation of DevSecOps Challenges, Practices, Metrics, and Tools

(RQ3: How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?

Sub-question 3.1: What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?

Sub-question 3.2: Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?

To address RQ3 and associated sub-questions, an empirical investigation was conducted to validate, refine, and improve the findings of the MLR study, i.e., the identified DevSecOps challenges, practices, tools, metrics, and the original version of the conceptual framework. It employed qualitative research methodology, incorporating a quantitative survey that combined Delphi and Analytic Hierarchy Process (AHP) methods (Zhao, Md Ali, & Ahmad, 2023).

The Delphi-AHP study consisted of three survey rounds with 18 participants, who are DevSecOps experts with various roles, including academic, industrial, managerial, and technical. The data were collected via an online survey that used multiple question formats, including AHP pairwise comparisons, multiple-choice questions, and open-ended questions.

In addition, a dissent analysis was conducted to investigate whether experts with different roles agree or disagree on the identification and prioritisation of DevSecOps (Beiderbeck et al., 2021).

8.1.1 Identification and Prioritisation of DevSecOps Challenges, Practices, and Metrics Using the Delphi-AHP Method

Over three iterations of the Delphi-AHP study, the importance of the identified DevSecOps challenges, practices, and metrics was rated and ranked, and new learnings were identified and incorporated into the findings.

8.1.1.1 DevSecOps Challenges

By conducting the first and second rounds of the Delphi-AHP study, the priorities and rankings of 35 DevSecOps challenges (28 initially identified challenges in MLR, plus seven new ones that were added during Round One) and four categories were calculated and reported, as shown in **Figure 8. 1** and **Figure 8. 2**.

Figure 8. 1 - Revised DevSecOps challenges in AHP format

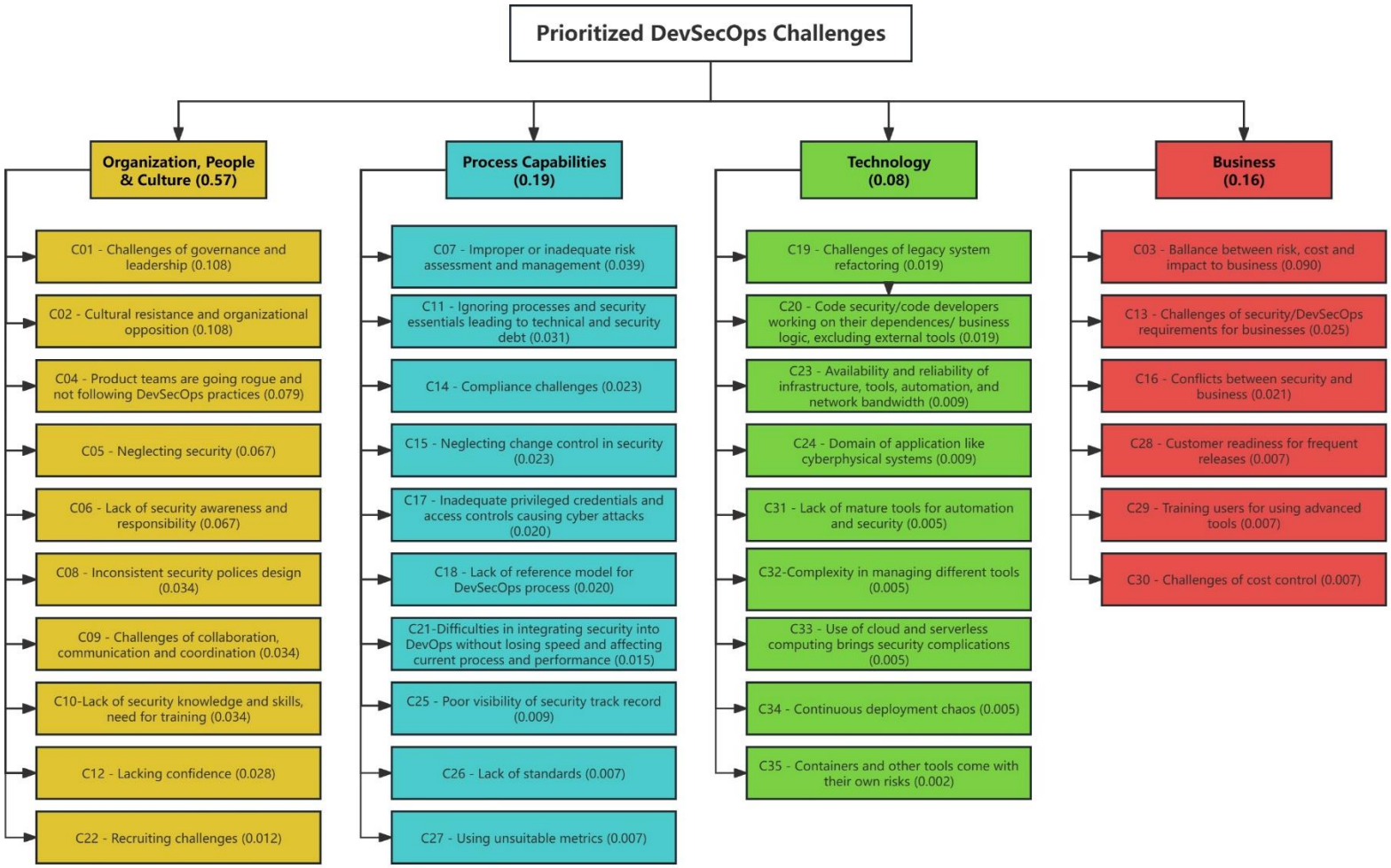
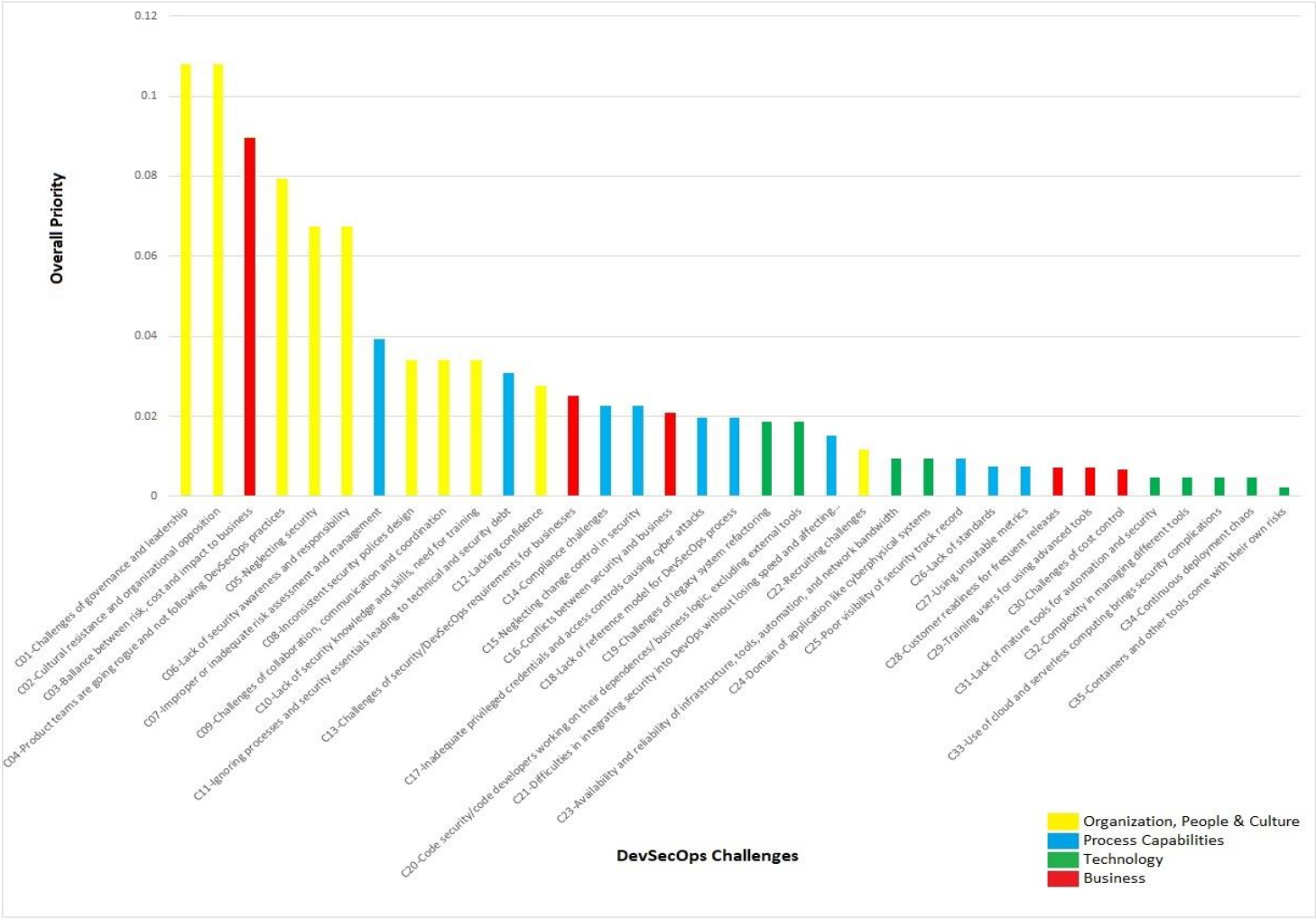


Figure 8. 2 - Prioritisation of DevSecOps challenges



To indicate the prioritisation more visually, the identifiers of these challenges were re-numbered according to their priority rankings. For instance, C01 represents the most important challenge, C02 the second most important, and so on. Four colour schemes stand for four categories of challenges. By default, a smaller identifier represents a higher ranking within each category.

As shown in **Figure 8. 1**, "Organisation, People and Culture" is ranked as the most important category of DevSecOps challenges, with a priority of 0.57, significantly higher than the other three categories. "Process Capabilities" is ranked second with a priority of 0.19, followed by "Business" with a priority of 0.16. The least important category of DevSecOps challenges is "Technology", with the lowest priority of 0.08.

As shown in **Figure 8. 2**, eight of the top ten most important challenges are in the "Organisation, People and Culture" category, because of its highest category priority, e.g., the top two important challenges "C01 – Challenges of governance and leadership" and "C02 – Cultural resistance and organisational opposition". "C07 – Improper or inadequate risk assessment and management" is the most important in the "Process Capabilities" (PC) category and is the only PC-related challenge in the top ten.

By contrast, one business challenge, "C03 – Balance between risk, cost, and impact to business", ranks third despite not having a high category priority, reflecting its individual high importance. The lowest priority of the "Technology" category results in low overall rankings of technological challenges. "C19 – Challenges of legacy system refactoring" is the most important technology-related challenge, but it just ranks 19th out of 35 challenges.

8.1.1.2 DevSecOps Practices

After conducting the second and third rounds of the Delphi-AHP study, the evaluation of the importance of 63 DevSecOps practices (60 initially identified practices in MLR, plus three new practices added during Round Two) and four categories was completed, as shown in **Figure 8. 3** and **Figure 8. 4**.

Figure 8. 3 - Revised DevSecOps practices in AHP format

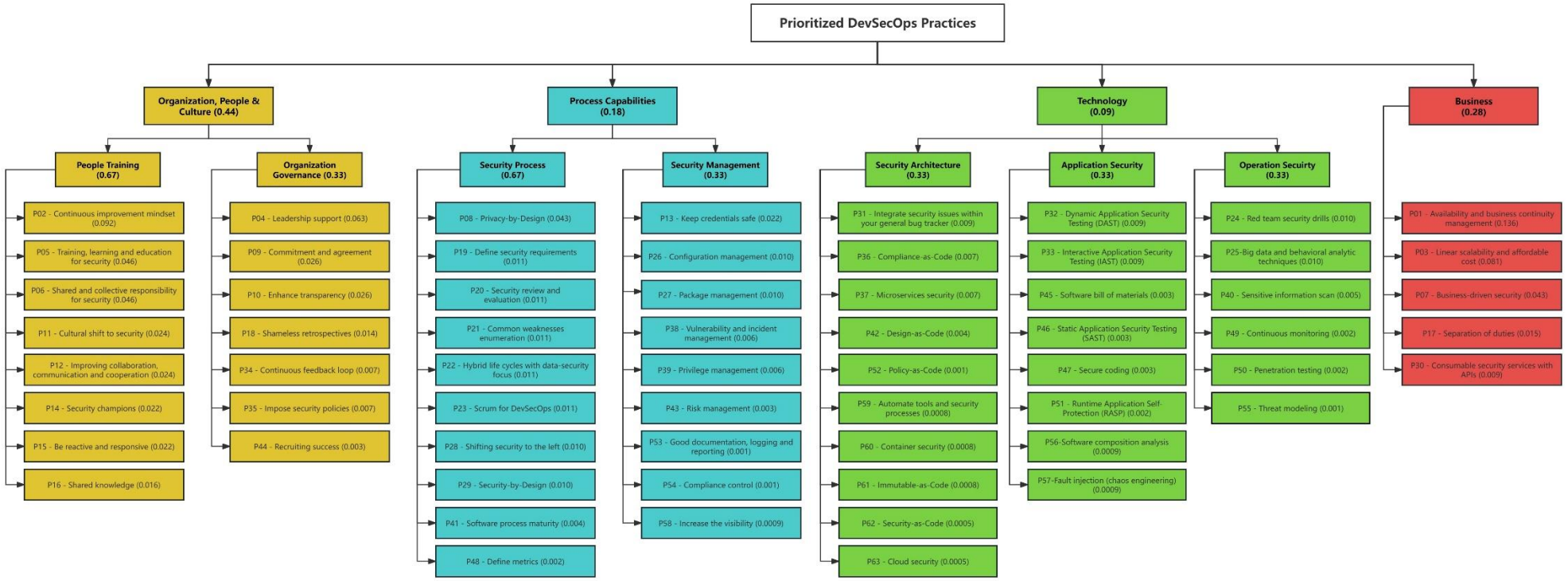
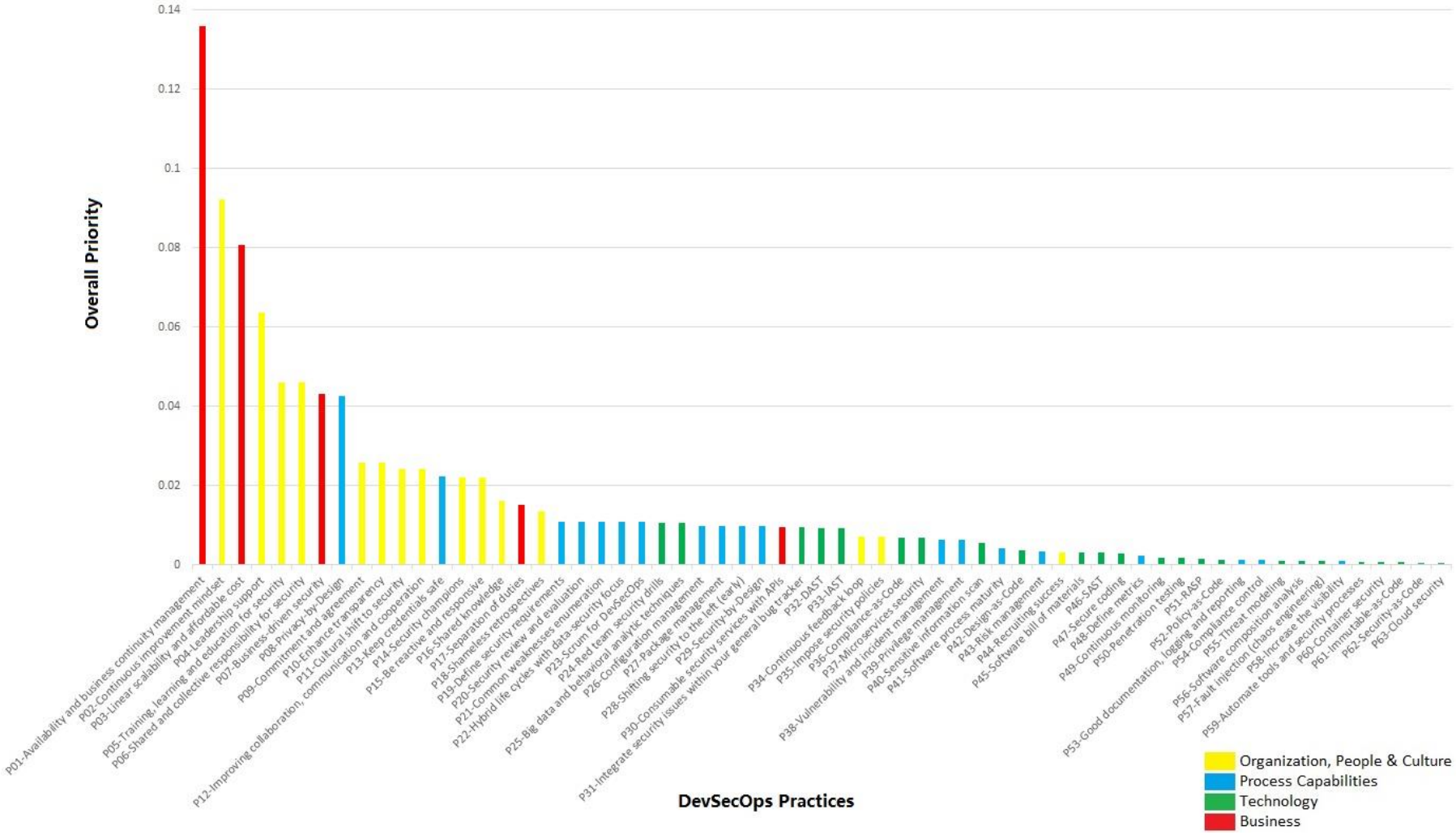


Figure 8. 4 - Prioritisation of DevSecOps practices



Same as the re-numbered challenges, practices' identifiers were re-numbered according to their priority rankings. A smaller identifier represents a higher overall ranking and a higher ranking within the category, by default.

As shown in **Figure 8. 3**, consistent with the results of the challenges evaluation, the most and least important categories of practices are "Organisation, People and Culture" (priority - 0.44) and "Technology" (priority - 0.09), respectively. "Business" is ranked second in importance (priority - 0.28), followed by "Process Capabilities" (priority - 0.18).

Figure 8. 4 shows that six of the top ten most important practices are in the "Organisation, People and Culture" category, because of its highest category priority, e.g., "P02 – Continuous improvement mindset", "P04 – Leadership support", "P05 – Training, learning and education for security", and "P06 – Shared and collective responsibility for security". Even so, the first and third places are business-related practices, i.e., "P01 – Availability and business continuity management" and "P03 – Linear scalability and affordable cost".

"P08 – Privacy-by-Design" is the most important practice in the "Process Capabilities" category and the only PC-related practice in the top ten; the lowest priority for the "Technology" category results in numerous technological practices clustering at the bottom. The most important technology-related practice is "P24 – Red team security drills", which is a more comprehensive test that covers software, hardware, and even personnel, compared to other specific security tests.

8.1.1.3 DevSecOps Metrics

The third round of the Delphi-AHP study was conducted to calculate and report the priorities and rankings of 20 DevSecOps metrics and four categories. Based on participants' comments, the DevSecOps metric list was revised, and the metrics' identifiers were renumbered according to their rankings, as depicted in **Figure 8. 5** and **Figure 8. 6**.

Figure 8. 5 - Revised DevSecOps metrics in AHP format

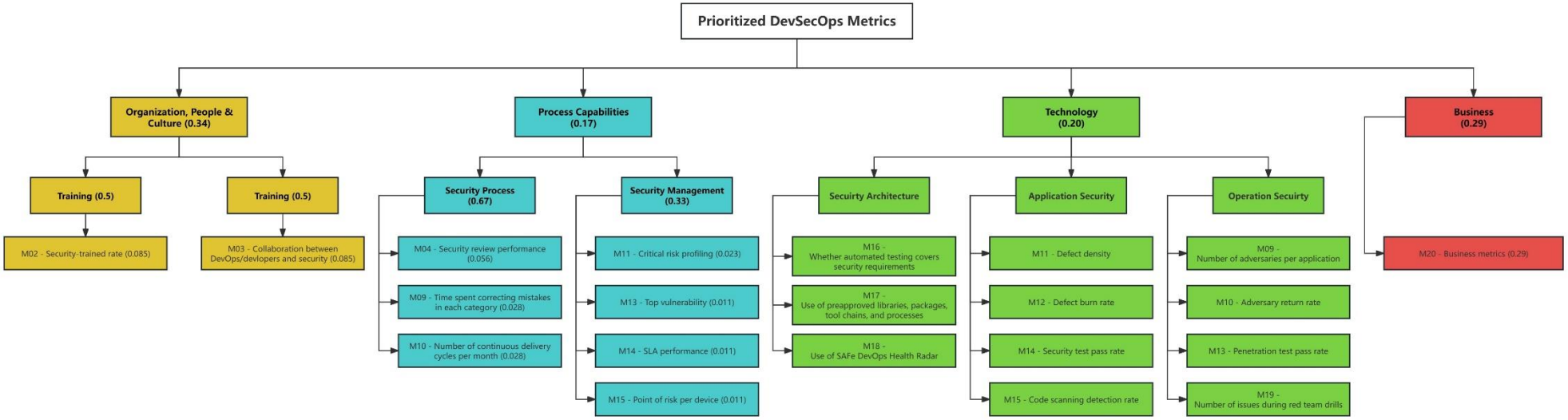


Figure 8. 6 - Prioritisation of DevSecOps metrics

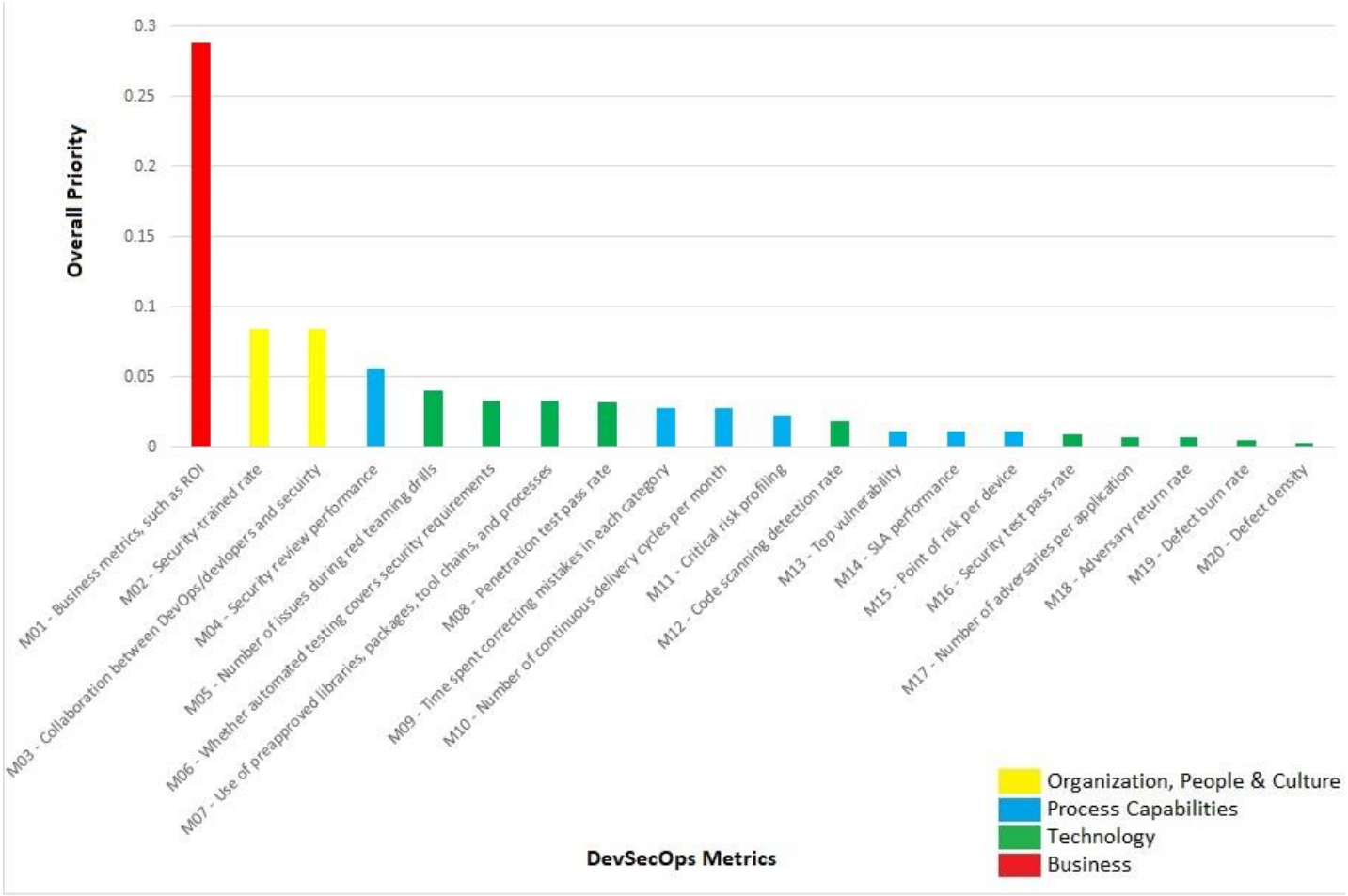


Figure 8. 5 indicates that the most important category of DevSecOps metrics is "Organisation, People and Culture", with a priority of 0.34, consistent with the evaluation results for challenges and practices. A notable difference is that the priorities of the "Business" and "Technology" categories have increased to 0.29 and 0.2, respectively, ranking second and third. "Process Capabilities" is ranked as the least important category, with the lowest priority of 0.17. As shown in **Figure 8. 6**, the top three important metrics are in the "Business" and "Organisation, People and Culture" categories. Nevertheless, there is a lack of identified metrics in both categories.

The MLR findings (Chapter 3) have highlighted the absence of specific business-related metrics for DevSecOps. Only a few business KPIs, such as Return on Investment (ROI) and Business Value Increment (BVI) (Nisha & Khandebharad, 2022), were identified. On the other hand, all DevSecOps metrics have business impacts on time, costs, and profits. For instance, faster security reviews and shorter remediation processes accelerate the entire SDLC, resulting in a short time-to-market, which in turn fosters customer trust and enhances profit prospects. Predicted vulnerabilities and lower defect density enable the proactive handling and mitigation of recurring bugs, thereby stretching the security budget. A more exhaustive coverage of automated tests reduces the need for human resources and effort to address late-stage defects, though it requires high upfront costs.

To address the gap in business-related DevSecOps metrics, Caniglia et al. (2025) recently presented the Framework of Business Index Concerning Security (FOBICS), which connects various metrics to quantify DevSecOps performance. Almost all their proposed metrics align with the identified metrics in this PhD research, except for a few metrics that are not purely security-related. For instance, the FOBICS framework covers all security metrics but excludes those focused solely on DevOps. However, the authors noted that multiple factors contribute to the FOBICS framework, meaning that a higher index value is not necessarily attributable to a specific sector, but could be due to one or more sectors, or even all sectors collectively. Another issue is that using this novel framework requires additional business costs for monitoring data collection and employee training (Caniglia et al., 2025).

All things considered, for the business-related metrics of DevSecOps, a decision was made to

remain with the traditional and general metric, e.g., "M01 – Business metrics such as using Return on Investment (ROI) and Business Value Increment (BVI)" (Nisha & Khandebharad, 2022) to evaluate the balance between security costs and benefits in DevSecOps paradigms. Organisations adopt DevSecOps to varying extents, driven by different business concerns. It is more trivial for them to separately select metrics as needed, rather than bundle all metrics together.

As reported previously in **Figure 8. 6**, two metrics in the "Organisation, People and Culture" (OPC) category, "M02 – Security-trained rate", and "M03 - Collaboration between DevOps/developers and security, or DevOps/developers' contributions to security", are respectively ranked as the second and third important metrics. M02 is the only OPC-related metric identified in the MLR study. M03 is a new OPC-related metric added based on participants' survey responses. "M04 – Security review performance" is ranked as the most important metric in the "Process Capabilities" category. "M05 – Number of issues during red teaming drills" is ranked as the most important metric in the "Technology" category, which aligns with the most important technology-related practice, "P24 – Red team security drills".

In addition, a few metrics have been reclassified: "Use of SAFe DevOps Health Radar," initially identified, and "consider referring to ISO27004 or NIST SP 800-55," newly collected. SAFe DevOps Health Radar is a framework for measuring the maturity of the CI/CD pipeline. ISO/IEC 27004 is an international standard for information security, while NIST SP 800-55 serves as a measurement guide for information security.

Those standards, documents, or frameworks are not specific metrics, and not applicable to the definition of "Metrics", which is previously given in **Table 3. 4**, Chapter 3 (Page 45), "the means to track progress, facilitate decision-making, and improve performance of DevSecOps practices by measuring implementation, effectiveness, efficiency, and impact". By contrast, they guide organisations on how to define metrics and measure performance, thereby being more like a type of tool that can be used to support the practice "P26 – Define metrics", rather than metrics themselves. Hence, the SAFe DevOps Health Radar, ISO27004 and NIST SP 800-55 have been moved from the metrics list to the tools list, as considering them as tools would be more sensible.

8.1.2 Identification and Prioritisation of DevSecOps Tools by Merging Three Tool Lists

The findings from three survey rounds have almost answered RQ3 and its associated sub-questions, i.e., have sorted out the dimensions of challenges, practices, and metrics. However, DevSecOps tools were not assessed using the Delphi-AHP method for some reasons:

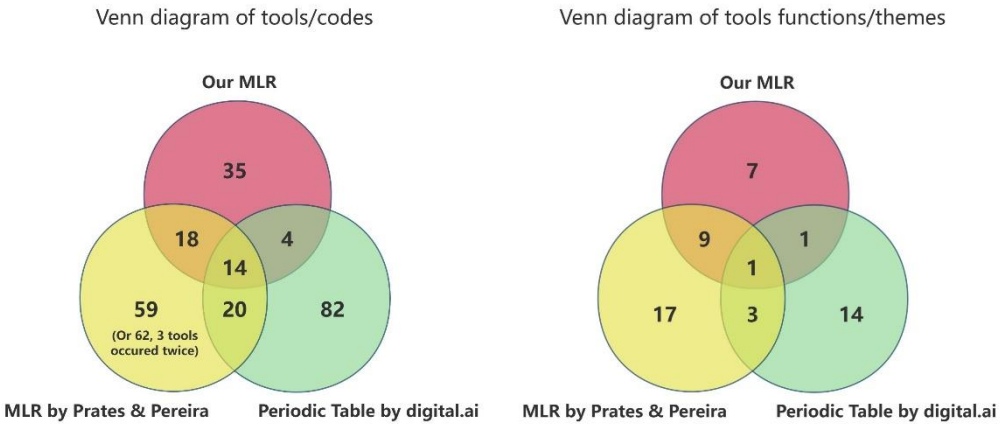
- The themes of tools overlapped with some themes of practices in the "Technology" category, so that "Tools" could be a subset of "Practices", especially technology-related practices.
- Making pairwise comparisons is the fundamental principle of the AHP method; however, tools of different types are incommensurable. For instance, a security testing tool and a communication tool are not comparable in perceived importance, as they serve different purposes.
- It is unrealistic to assume that participants have experience in using all kinds of tools, even though they are experts.

Therefore, DevSecOps tools were not evaluated using the Delphi survey method. Instead, they were validated, complemented, and revised by referring to two related works. The first one is a recently published MLR study (Prates & Pereira, 2025) that focuses on identifying tools used to support DevSecOps practices. The other is an industry publication, "Periodic Table of DevSecOps Tools" from Digital.ai (2025), which identifies top tools across the DevSecOps lifecycle. These two related publications provide current, comprehensive lists of DevSecOps tools but use different taxonomies, reflecting the latest academic and industry perspectives.

As presented in Chapter 3, this research's MLR identified 71 DevSecOps tools/codes and classified them into 18 themes/functions. The MLR study conducted by Prates and Pereira (2025) includes 114 tools and corresponds them with 27 identified DevSecOps practices. The "Periodic Table of DevSecOps Tools" by Digital.ai (2025) identifies 120 tools in 19 categories. Compared to this research's MLR findings, both related works provide broader coverage of the tools and offer updated insights on classification.

Figure 8. 7 shows two Venn diagrams that illustrate comparisons among three tool lists, in terms of tools/codes and functions/themes.

Figure 8. 7 - Venn diagrams of DevSecOps tools and functions



There are not many overlapping tools, and even fewer overlapping functions. Collectively, the three lists identify 14 tools and one function in common. This research's MLR (Chapter 3) and the MLR by Prates and Pereira (2025) yield more similar findings to those of the industrial tool table (Digital.ai, 2025). This is because both MLRs employ similar research methods and refer to the DevSecOps model proposed by Gartner (MacDonald & Head, 2016). In contrast, the "Periodic Table of DevSecOps Tools" (Digital.ai, 2025) takes a more general classification of tools' functions. For instance, one of its categories is "Security", which encompasses all security-related tools without distinguishing between specific functions, such as various types of security testing.

Listed below are other factors to consider that could result in different classifications of tools:

- A tool can have multiple functions with indistinct boundaries. For example, major brands like Amazon and Microsoft have launched comprehensive services or platforms that cover a variety of DevSecOps functions. These versatile instruments with multiple applications may be classified differently depending on the classifiers' subjective understandings of DevSecOps.
- The dynamic of the tools market demands constant change and adjustment, resulting in many tools being gradually phased out. Alternatively, a tool may undergo functional changes or integration, leading to a significant shift from its initial positioning.

By analysing and comparing the three tool lists, a comprehensive list of DevSecOps tools was

created, as presented in **Table 8. 1**, covering 230 tools with 35 functions across the DevSecOps lifecycle. The full version of this DevSecOps tool list, with statistics, is available on Zenodo (https://doi.org/10.5281/zenodo.16932278) and can help practitioners identify and select appropriate tools for the DevSecOps lifecycle.

Table 8. 1 - Revised list of DevSecOps tools

Tool Function/Theme (35)	Tool/Code (230)
T01-Monitoring and alerting tools	New Relic, Pager Duty, Datadog, Grafana, Tripwire, Nagios, Suricata, Ganglia, Kibana, Pingdom, Graphite, Sensu, Cacti, StatsD, Falco, AppDynamics, Prometheus, BigPanda, DynaTrace, Elastalert, Alertlogic, Opsgenie, Digital.ai App Protection
T02-SAST tools	SonarQube, Snyk, Veracode, Bandit, Checkmarx, Fortify SCA, Coverity, Docker bench, Contrast security, Git leaks, Shiftleft, Codacy, Tfsec, Kiuwan , Flawfinder, Spotbugs, Graudit, PMD, Terrascan, Frama-C
T03-Container orchestration and security tools	Kubernetes, Docker, Docker Enterprise Edition, Amazon ECS, Amazon EKS, Azure AKS, Google GKE, Helm, Openshift, Aqua Security, Clair, Twistlock, Notary
T04-Configuration automation tools	Ansible, Puppet, Chef, Saltstack, Terraform, Consul, AWS Cloud Formation
T05-Continuous integration tools (Build automation tools)	Jenkins, Github actions, GitLab CI, Circle CI, Travis CI, Codefresh, Bamboo, Maven, AWS CodeBuild, Azure DevOps
T06-Continuous testing tools (Test automation tools)	Gautntlt, Selenium, BDD security, Sauce Labs, Topaz, Appium, Cucumber, ParaSoft, Tricentis Tosca, Digital.ai Continuous Testing, Squash TM, Jmeter, JUnit
T07-Continuous deployment tools (Deployment automation tools)	Spinnaker, GoCD, ArgoCD, Digital.ai Deploy, IBM UrbanCode Deploy, AWS CodeDeploy, Octopus Deploy, Pulumi, Azure DevOps, Harness, Tekton, OpsMx, Flux
T08-Cloud and security tools	Microsoft Azure, Google Cloud Platform, AWS, OpenStack, Google Firebase, Cloud Foundry, Heroku, ThreatModeler Cloud Edition, Trend Micro Cloud One, AWS Security service, AppScan on Cloud, Cloud Custodian
T09-Logging tools	Splunk, Logstash, Elasticsearch, PaperTrail, Loggly, SumoLogic, Graylog
T10-Threat modelling tools	IriusRisk, ThreatModeler, ThreadFix, Threagile, Threatspec, ThreatPlaybook, OWASP Threat Dragon, Pytm, Microsoft Threat Modeling
T11-Compliance and auditing tools	OpenSCAP, Inspec, Findsecbugs, ESLint security, ServerSpec, OSSAudit, KubeAudit
T12-Verson control system tools	Git, GitHub, GitLab's SCM, Subversion, Mercurial, Bitbucket, ISPW
T13-Collaboration tools	Slack, Mattermost, Teams, Confluence, Miro, Mural, HipChat, Trello
T14-DAST tools	OWASP ZAP, HCL appscan, Fortify Webinspect, Radamsa, FuzzDB, Nikto
T15-Software composition analysis tools	OWASP Dependency Check, Black Duck, Dependency Track, Nexus IQ, Whitesource, Anchore
T16-Vulnerability scanning tools	Retire.js, Arachni, Brakeman, Acutenix, Checkov, Trivy
T17-Access control tools	HashiCorp's Vault, Boundary, CyberArk Conjur, Keycloak
T18-Secrets management tools	Git Secrets, Docker Secrets, Blackbox, Talisman, TruffleHog
T19-Package management tools	Docker Hub, NMP, Artifactory, Yarn, NuGet, Nexus
T20-Enterprise agile planning tools	Jira, Jira Align, Digital.ai Agility, TargetProcess, Planview, Rally
T21-Vulnerability management tools	Rapid7 Nexpose, Stethoscope, HackerOne, Defect Dojo, ArcherySec
T22-Firewall and network security tools	Snort, Nmap, Signal Sciences
T23-RASP tools	Fortify Application Defender, Imperva RASP, OWASP Open RASP, Hdiv
T24-Database management tools	Liquibase, Delphix, Flyway , Toad

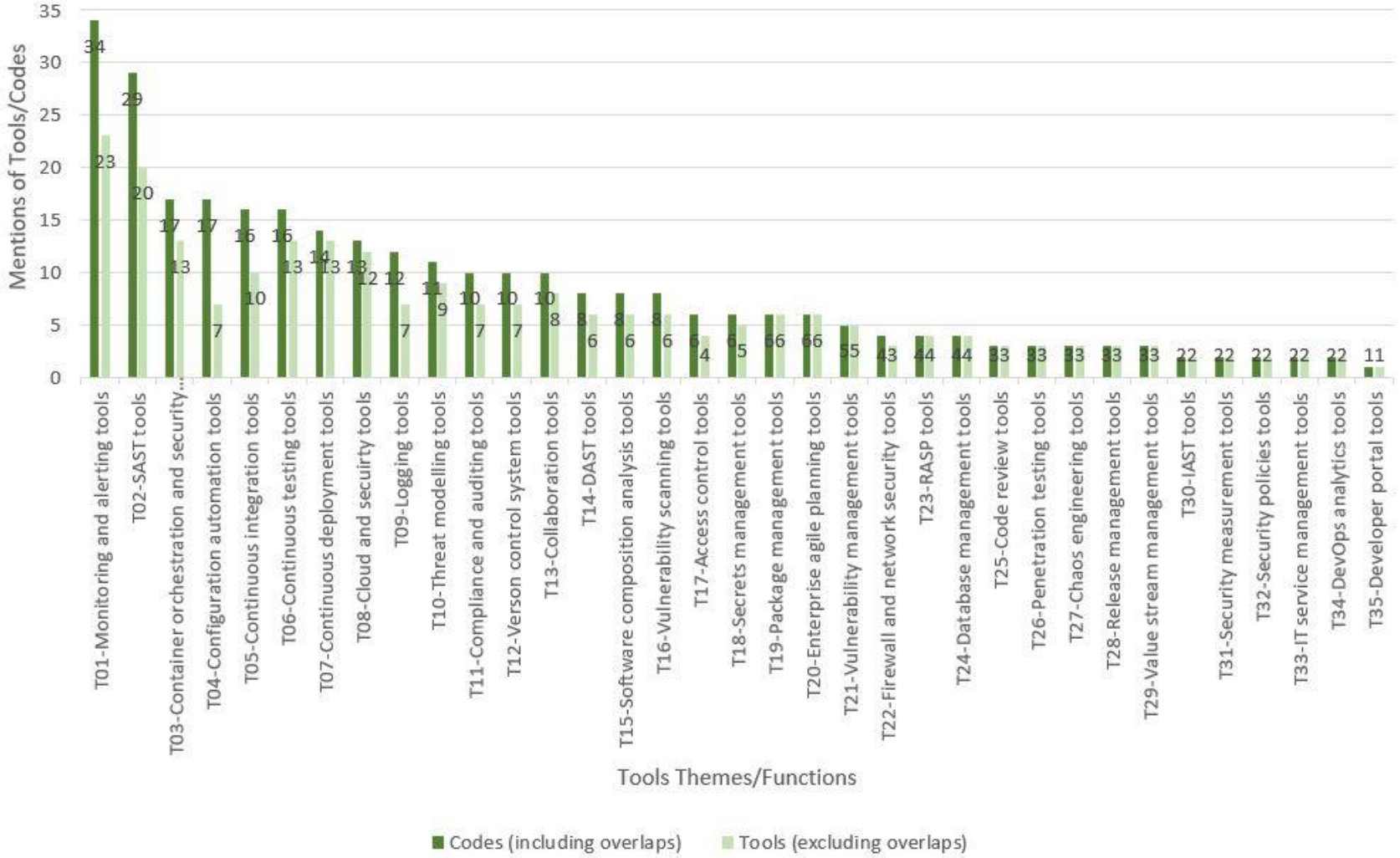
T25-Code review tools	Gerrit, Crucible, DevSkim
T26-Penetration testing tools	BurpSuite, Metasploit, Astra
T27-Chaos engineering tools	Chaos Monkey, Chaos Mesh, Chaos Kube
T28-Release management tools	AWS CodePipeline, UrbanCode Release, Digital.ai Release
T29-Value stream management tools	Planview Viz, Plutora, Digital.ai VSM
T30-IAST tools	Synopsys Seeker, OWASP Benchmark
T31-Secuirty measurement tools	NIST SP 800-55, ISO/IEC 27004
T32-Security policies tools	Open Policy Agent, Kyverno
T33-IT service management tools	ServiceNow, BMC ITSM
T34-DevOps analytics tools	Digital.ai Intelligence, SAFe DevOps Health Radar
T35-Developer portal tool	Backstage

Tools' prioritisation was based on usage rather than perceived importance. As shown in **Figure 8. 8**, the mentions of tools/codes within their themes/functions were counted, and the 35 identified tool themes/functions were ranked from high to low.

A higher number of "mentions of codes (including overlaps)" within a theme/function indicates greater usage of this tool type, supporting the greater importance of a particular functionality. In contrast, the number of "mentions of tools (excluding overlaps)" reveals the extent of tool support for each functionality. This indicates there are several specific tools for each functionality. The higher the count, the more tools support a functionality and its importance.

Same with the challenges, practices, and metrics, the tools' identifiers were also renumbered, according to priority rankings. For example, "T01 – Monitoring and alerting tools" is the most mentioned in the three tool lists, with the highest number of codes (34, including overlaps) and specific tools (23, excluding overlaps). Thus, "T01 – Monitoring and alerting tools" is the top tool type for DevSecOps, with the highest usage and the most tool support.

Figure 8. 8 - Mentions and rankings of DevSecOps tools



8.1.3 Summary of Dissent Analysis

A dissent analysis was conducted to discuss participants' agreements and disagreements after each round of the Delphi survey. In this research, the Coefficient of variation (CV) was calculated to evaluate the degree of dissent at the individual level, thereby analysing consensus or dissent for each participant regardless of position or role, to verify whether participants with similar roles would have more consensual opinions, or if individual variations are the primary factor.

In addition to the CV, the consensus and dissent among four participant roles (i.e., Academic, Industrial, Managerial, and Technical) were compared and analysed to address sub-question 3.2: Do experts disagree on prioritisation *due to their different roles, e.g., academic, industrial, technical, and managerial?*

Based on the results of the dissent analysis in three iterations of the Delphi-AHP study, it can be concluded that **participants with similar roles hold more consistent opinions on the prioritisation of DevSecOps challenges, practices, and metrics; participants with different roles have significantly dissenting opinions.**

According to the results reported previously in **Figure 8. 7**, it can be concluded that **academia and industry in SE have significantly dissenting opinions on the identification and prioritisation of DevSecOps tools, especially regarding classification methods**.

The next section addresses Research Question Four, i.e., DevSecOps in the context of Global Software Engineering (GSE).

8.2 RQ4 – Adoption of DevSecOps in GSE Contexts

(RQ4: What are the experts' opinions on DevSecOps in GSE contexts?

Sub-question 4.1: How is DevSecOps different between local and global settings?

Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?)

To address RQ4 and its associated sub-questions, a set of multiple-choice and open-ended questions was used to survey participants' opinions on differences between DevSecOps in local and GSE contexts. Each survey round included a multiple-choice question with four options: "Extremely Different", "Slightly Different", "Not Different", and "Uncertain", to address Sub-question 4.1 about "How is DevSecOps different in local and GSE contexts?". Additionally, an open-ended question at the end of each survey round was included to collect participants' responses on "What is different?" for Sub-question 4.2.

Table 8. 2 summarises the results of three survey rounds to answer Sub-question 4.1. For the aspects of challenges, practices, and metrics, the "Slightly Different" option consistently accounts for the highest proportions: 44.44%, 35.71%, and 73.33%, respectively. As discussed earlier, although no such survey round was conducted for DevSecOps tools, they can be managed as a subset of practices. Hence, a solid conclusion can be drawn that **DevSecOps is seen and practised slightly differently between local and GSE contexts**.

Table 8. 2 - Summary of opinions on DevSecOps in GSE

"How is DevSecOps different between local and global settings?"				
	Not Different	Slightly Different	Extremely Different	Uncertain
DevSecOps Challenges	33.33%	44.44%	16.67%	5.56%
DevSecOps Practices	28.57%	35.71%	14.29%	21.43%
DevSecOps Metrics	0%	73.33%	6.67%	20%
DevSecOps Tools	N/A	N/A	N/A	N/A

To answer "What is different?" for Sub-question 4.2, an open-ended question was posed to collect participants' responses. The Thematic Analysis (TA) method was employed for data analysis (Braun & Clarke, 2006). Results showed no specific differences in practices, metrics, or tools. However, five global DevSecOps challenges were derived:

- CG1 – Amplified challenges of collaboration, communication and coordination due to remote work.
- CG02 – Challenges of data residency and management due to more inconsistent policies, regulations, and laws worldwide.
- CG03 – Magnified different understanding of DevSecOps culture in global settings.

- CG04 – Magnified challenges of risks and threats in global settings.
- CG05 – Different business focus and levels of importance in global settings.

Furthermore, it can be concluded that:

- **The differences between local and global DevSecOps primarily lie in Organisation, People, and Culture.**
- **DevSecOps challenges will be magnified as software development ecosystems shift from local to global settings, where security requirements ought to be as critical as functional requirements.**

8.3 Updated Conceptual Framework – DevSecOps CPTM Model (Version 2.0)

In this section, the updated conceptual framework, "DevSecOps CPTM Model (Version 2.0)," is presented based on findings from three rounds of the Delphi survey. This section also provides the model design, upgrades, and a scenario to guide navigation of the model.

8.3.1 Model Design and Upgrade

As presented in Chapter 3, the MLR study identifies five main aspects of the DevSecOps research (i.e., Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement), collects related codes and themes of each aspect, and generates a conceptual framework named "DevSecOps CPTM Model (Version 1.0)" by integrating the themes of the latter four aspects (see **Figure 3. 8**, in Chapter 3, on Page 57).

Jabareen (2009) defines the conceptual framework as "a network or a plane of interlinked concepts that together provide a comprehensive understanding of a phenomenon or phenomena. The concepts that constitute a conceptual framework support one another, articulate their respective phenomena, and establish a framework-specific philosophy".

The process and methods for creating the model are fully presented in Chapter 3. The DevSecOps CPTM Model (Version 1.0) is the main contribution of the MLR study. It provides a breakdown

and a broad overview of the DevSecOps topic, from which researchers and practitioners can select an area of focus to enhance their knowledge or practice. It also serves as a theoretical basis for the entire research.

Nonetheless, the initial model (Version 1.0) had gaps that needed to be filled, e.g., the absence of global aspects and the gap between identified tools and practices. Hence, a subsequent Delphi-AHP study was conducted to evaluate and prioritise the identified challenges, practices, and metrics, collect new findings for these aspects, and synthesise opinions on global settings. The gap between DevSecOps tools and practices was also closed by merging three tool lists. Ultimately, the model has been upgraded to Version 2.0.

Table 8. 3 reviews the design of the DevSecOps CPTM Model, including four elements of the model, four categories of identified themes, and ten phases of the model. The ten phases of the Gartner DevSecOps model (MacDonald & Head, 2016) are integrated into the DevSecOps CPTM Model, and all identified themes of the four elements have been allocated to these lifecycle phases, as listed in **Table 8. 4**.

Table 8. 3 - Design of the DevSecOps CPTM Model

Four Elements of the CPTM Model	
Challenges	The obstacles and uphill tasks encountered when adopting DevSecOps require ongoing efforts to overcome. "Problems". "Issues" and "Concerns" are categorised as fitting the "Challenges" aspect. Challenges' identifiers are re-numbered according to overall rankings. For example, C01 represents the most significant challenge. CG01-05 are five newly identified DevSecOps challenges in the GSE context.
Practices	DevOps and security activities are suited for DevSecOps. "Activities", "Approaches", "Solutions", and "Strategies" are categorised as fitting the "Practices" aspect. Practices' identifiers are renumbered according to overall rankings. For example, P01 represents the most important practice.
Tools	Specific tools and technical approaches that are used for DevSecOps practices. They could be part of a subset of DevSecOps practices, particularly those related to technology. Tools' identifiers are renumbered according to overall rankings. For example, T01 represents the most frequently mentioned type of tools.
Metrics	The means to track progress, facilitate decision-making, and improve performance of DevSecOps practices by measuring implementation, effectiveness, efficiency, and impact. Metrics' identifiers are renumbered according to overall rankings. For example, M01 represents the most important metric.
Four Categories of Themes	
Organisation, People and Culture (OPC)	The category encompasses themes related to organisational structure, people management, and cultural strategies, including breaking silos, collaboration, communication, sharing, training, and recruiting. The "OPC" category is shaded in YELLOW in the model.
Process Capabilities (PC)	The category encompasses themes related to the capabilities of the DevSecOps process, including the integration of security, security-left, continuous activities, risk management, and a faster lifecycle. The "PC"

	category is shaded in BLUE in the model.
Technology	The category encompasses themes related to technological approaches, software, and hardware tools, including automation, cloud, containerisation, testing techniques, and associated tools. The "Technology" category is shaded in GREEN in the model.
Business	The category encompasses themes related to business benefits, customers, quality of product and service, e.g., increasing value, higher quality, fewer impacts to users, etc. The reason for adding this category was that the MLR, especially the GL results, showed a business perspective on DevSecOps. The "Business" category is shaded in RED in the model.
Ten Phases of the DevSecOps Model by Gartner (MacDonald & Head, 2016)	
Plan	The phase involves setting project objectives, identifying security requirements, planning security measures, defining metrics and policies, preparing organisations/teams, selecting technologies/tools, and developing budgets.
Create	The phase involves executing the plan, preparing security practices, and setting up security tools.
Verify	The phase is to conduct security practices by using appropriate (automated) tools and technologies, such as security tests (SAST, DAST, IAST) and software composition analysis (SCA).
Preproduction	The next phase involves incorporating additional security tests, including chaos engineering and red team drilling.
Release	The phase involves signing the software, preparing it for release, and integrating it into the production environment. This includes reviewing configuration, infrastructure, network bandwidth, compliance, and other relevant factors.
Prevent	The phase is to protect the runtime environment architecture.
Detect	The phase involves continuously monitoring and scanning the runtime environment architecture.
Respond	The phase is to address the vulnerabilities detected in the previous phase.
Predict	The phase involves analysing vulnerabilities to identify their causes.
Adapt	The phase aims to enhance security processes and re-plan the DevSecOps lifecycle, drawing on lessons learned from previous phases.

Table 8. 4 - Identified elements mapped to ten phases by Gartner

Phase	Challenge	Practice	Tool	Metric	Phase	Challenge	Practice	Tool	Metric
1-Plan	C02, CG03	C11	N/A	M03	**3-Verify**	C31	P25	T34	N/A
							P32	T14	M16
	C08, CG02	P35, P52	T32	N/A			P33	T30	M16
							P46	T02	M16
	C10	P05, P16	N/A	M02			P47	T25	M12
	C22	P44	N/A	M02			P56	T15	M19,20
	C07, CG04	P43	T31	M11, M15	**4-Preprod**	C23, C24	P20	T31	M04
							P24	T22	M05
	C11	P19	N/A	N/A			P50	T26	N/A
		P41	T34	N/A		C34	P57	T27	N/A
	C18	P23	T20	N/A	**5-Release**	C14	P36, P54	T11	M12
	C21	P22, P28, P29	N/A	M09, M10, M14		C28	P30	T28	M01
						C29	P01	T33	M01
					6-Prevent	C17	P13, P40	T18	N/A
	C26	P21	T31	M13			P39	T17	N/A

		P55	T10	M17,18				T24	N/A
	C27	P48	T31	M04			P08	N/A	N/A
	C03, C13, CG05	P07	T29	M01		C33	P51	T23	N/A
							P63	T08	
						C35	P51	T23	N/A
							P60	T03	
					7-Detect	C33	P37, P63	T08	M19, M20
	C16	P17	T13	M03			P49	T01	
	C30	P03	T29	M01			P51	T23	
2-Create	C01	P04, P10	N/A	N/A		C35	P60	T03	
	C04, C05, C06	P06, P14, P15	N/A	N/A			P49	T01	
			N/A	N/A			P51	T23	
			N/A	N/A	**8-Respond**	C33	P37, P63	T08	N/A
	C09, CG01	P12, P18, P34	T13	M03			P38	T21	M11, M15
							P53	T09	
	C12	P02, P09	N/A	N/A		C35	P60	T03	N/A
	C19	P31	N/A	N/A			P38	T21	M11, M15
		P42	N/A	N/A			P53	T09	
		P62	T02	M16	**9-Predict**	C15	P26	T12	M07
			T14				P27	T19	
			T23				P45	T35	
			T30				P61	T04	
			T03	N/A		C25	P53	T09	N/A
			T15	M19,20			P58	T35	N/A
			T16	M11,15	**10-Adapt**	This phase is conducted to improve security processes and re-plan the DevSecOps lifecycle, based on the lessons learned from all the previous phases, so any practice/tool/metric can be adapted during this phase.			
	C20, C31, C32	P59	T04	N/A					
			T06	M06					
			T05	M10					
			T07						

8.3.2 DevSecOps CPTM Model (Version 2.0)

Based on the findings from the MLR study (Chapter 3) and three iterations of the Delphi-AHP study (Chapters 5, 6, 7, and 8), the DevSecOps CPTM Model has been upgraded to Version 2.0, as depicted in **Figure 8. 9**. The lossless image of the DevSecOps CPTM Model (Version 2.0) is available at zenodo.org (https://doi.org/10.5281/zenodo.16932278).

Figure 8. 9 - DevSecOps CPTM Model (Version 2.0)

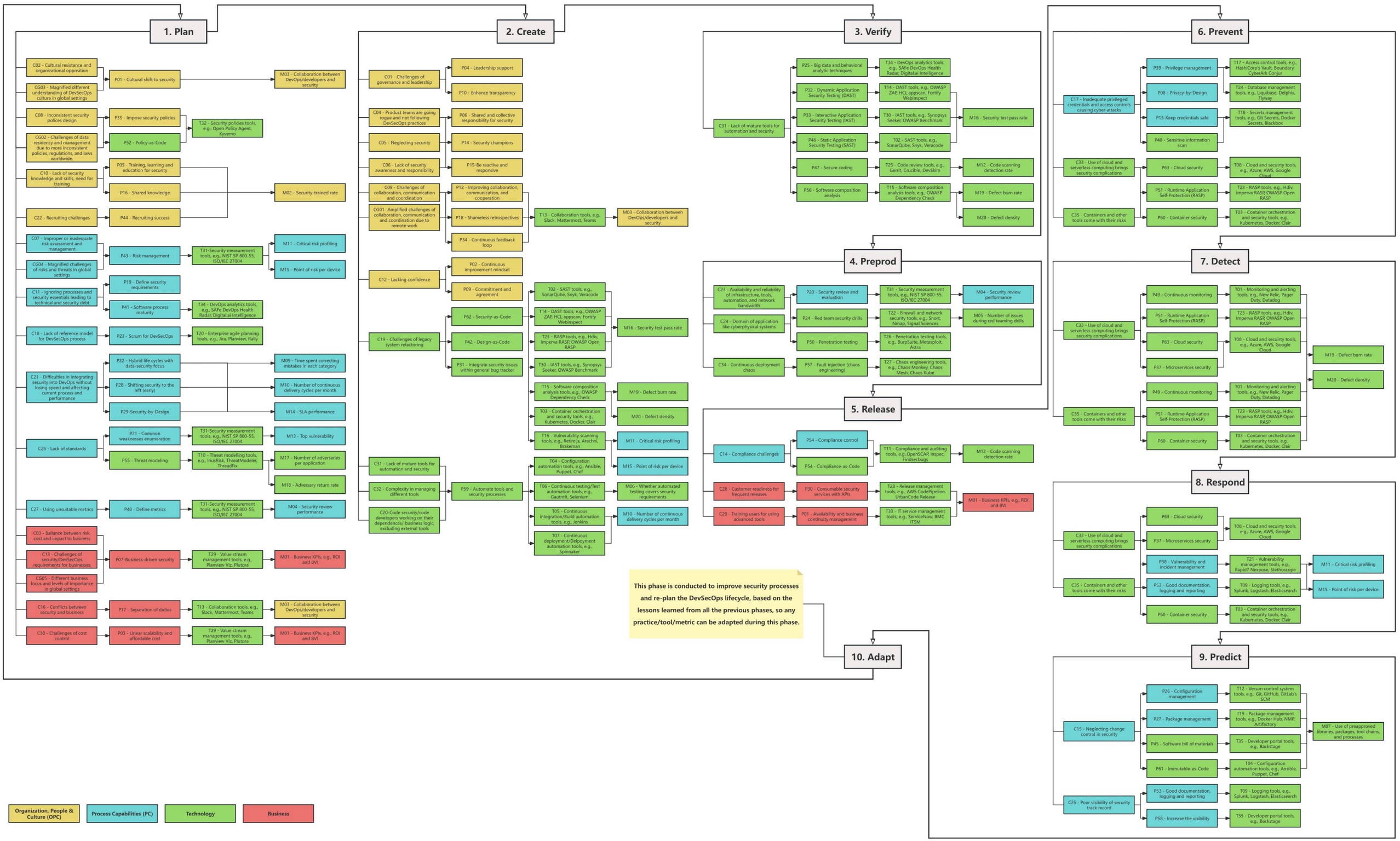


Table 8. 5 compares the DevSecOps CPTM Model between Version 1.0 (in **Figure 3. 8**, Chapter 3, Page 57) and Version 2.0 (in **Figure 8. 9**). Version 2.0 provides a more comprehensive coverage including 35 challenges, 63 practices, 20 metrics, 230 DevSecOps tools across 35 types/functions, and five newly identified challenges of DevSecOps in GSE.

Table 8. 5 - Comparison of the DevSecOps CPTM Model between Versions 1.0 and 2.0

	Version 1.0	**Version 2.0**
Four Elements	28 DevSecOps Challenges. 60 DevSecOps Practices. 71 DevSecOps Tools across 18 types/functions. 20 DevSecOps Metrics.	35 DevSecOps Challenges. 63 DevSecOps Practices. 230 DevSecOps Tools across 35 types/functions. 20 DevSecOps Metrics.
Four Categories	Organisation, People and Culture (OPC). Process Capabilities (PC). Technology. Business.	Organisation, People and Culture (OPC). Process Capabilities (PC). Technology. Business.
Ten Phases	Plan, Create, Verify, Preproduce, Release, Prevent, Detect, Respond, Predict, and Adapt.	Plan, Create, Verify, Preproduce, Release, Prevent, Detect, Respond, Predict, and Adapt.
GSE Contexts	Not provided.	5 DevSecOps Challenges in the GSE context
Prioritisation	Not provided.	Identified elements are ranked by DevSecOps experts, using new identifiers to indicate the prioritisation.

The prioritisation of the identified elements is a new feature added in Version 2.0. In the model, each element has an identifier, which is a combination of letters and numbers. The letter (C, P, T, or M) indicates the type of this element (Challenges, Practices, Tools, and Metrics). The number in the identifier denotes the priority ranking. For instance, C01, P01, M01, and T01 represent the most important Challenge, Practice, Metric, and Tool type; C02, P02, M02, and T02, the second most important; and so on.

According to Gartner's DevSecOps model (MacDonald & Head, 2016), the DevSecOps lifecycle has ten phases: Plan, Create, Verify, Preproduce, Release, Prevent, Detect, Respond, Predict, and Adapt (see **Figure 3. 9** in Chapter 3, on Page 58). The reason for integrating the findings from

this research into those phases is that Gartner's DevSecOps model has been widely accepted by the SE industry and academia, and appears to be one of the most popular reference models for DevSecOps adoption (Prates & Pereira, 2025).

The ten phases form a loop that starts with planning and ends with adapting/re-planning, where each iteration is completed, and the next is improved. A change of the DevSecOps CPTM Model between Version 2.0 and Version 1.0 (not reflected in **Table 8. 5**) is that Version 2.0 has no specific items allocated under the "Adapt" phase, which is conducted to improve security processes and re-plan the DevSecOps lifecycle, based on the lessons learned from all the previous phases, so any practice/tool/metric can be adapted during this phase.

Within each lifecycle phase, four columns correspond to the four elements of the DevSecOps CPTM Model: Challenges, Practices, Tools, and Metrics. The connecting lines demonstrate the relationships between those four elements. It outlines the practices that can be applied to overcome the corresponding challenges, the tools that can support DevSecOps practices, and the metrics that can be used to measure the performance of these practices and tools.

In the CPTM Model, one challenge may correspond to multiple practices, one practice may correspond to multiple challenges, and not every practice has its corresponding tools and metrics. For example, some practices, especially many in the "Organisation, People and Culture" category (yellow in the model), do not require support from tools and technologies. Besides, additional business costs for monitoring and collecting data for measurement must be considered, so that it is not necessary to define a metric for each practice. This is why the model includes 63 DevSecOps practices, but only 20 metrics and 35 tool types. It is also worth noting that a few elements appear to be cross-cutting themes across categories, and their categories potentially differ from those in the thematic analysis. For example, tools are categorised under the "Technology" category (green colour), but some may also appear in other categories within the model to match their corresponding practices.

The distribution of the four categories (i.e., "Organisation, People and Culture" (OPC), "Process Capabilities" (PC), "Technology", and "Business") across ten phases can be analysed by examining the four colours in the model.

As shown in **Figure 8. 9**, many DevSecOps challenges and practices in the "OPC" (yellow colour)

and “PC” (blue colour) categories occur in the “Plan” and “Create” phases, revealing that many challenges can arise at the beginning of the DevSecOps process. In light of this, a set of corresponding practices, tools, and metrics should be planned and developed as early as possible, reflecting the spirit of DevSecOps – shifting security to the left (Rajapakse et al., 2022). Hence, a definite plan and thorough execution are the key to the success of DevSecOps programs.

Notable, too, all the business-related challenges, practices, and metrics (red colour) appear in the “Plan” and “Release” phases, reflecting the fact that the business considerations are also important for releasing the product, not only planning, therefore the organisations who adopt DevSecOps should be concerned with their business performance at the beginning (Plan), the middle (Release) and the end (Adapt/Replan) of the lifecycle.

Compared with the previous three categories, technology-related challenges, practices, metrics, and tools (green colour) are distributed across the phases “Verify”, “Preproduction”, “Prevent”, “Detect”, “Respond”, and “Predict”. This reveals that DevSecOps implementation relies primarily on technological enablers and tools.

8.3.3 Navigation for the DevSecOps CPTM Model with A Scenario

There are many applications of DevOps, but DevSecOps are few because security is an elaborate and time-consuming process (Nagasundari et al., 2025). Singh (2025) illustrated how to integrate threat modelling in DevSecOps with a real-world case. Whereas Nagasundari et al. (2025) and Ramesh et al. (2024) navigated their DevSecOps-related frameworks with fictitious scenarios.

This subsection provides a fictitious scenario named “Global Retailer E-Commerce Platform Security Shift” to navigate and illustrate how organisations might use the DevSecOps CPTM Model (Version 2.0) in **Figure 8. 9**. This scenario was created by the doctoral researcher and his supervisors, drawing on the use cases by Singh (2025), Ramesh et al. (2024), and Nagasundari et al. (2025). The DevSecOps CPTM Model (Version 2.0) is a complex, multi-layered model comprising hundreds of concepts/items across four elements, four categories, and ten phases. Using a scenario can help illustrate potential strategies for implementing the model.

Cordova-Pozo and Rouwette (2023) define scenario or scenario planning as “a description of a future situation and the course of events which allows one to move forward from the actual to the

future situation". Scenarios aim to address uncertainty, such as changing systems, lacking information, or unpredictable states, so they should be useful, plausible, and have the power to break down paradigms (Godet, 2000).

Context:

A multinational retail organisation operates a massive e-commerce platform serving millions of customers in over 40 countries. They have a mature DevOps process that supports daily deployments to multiple regional data centres, powering online sales, in-store pickups, online payments, order tracking, inventory changes, return services, and personalised shopping features. A central InfoSec team manages security after releases, often leading to late-stage vulnerability discoveries and costly emergency patches. Hence, the retail organisation decided to integrate security into their original DevOps process and evolve to DevSecOps.

Current State (DevOps):

- Pipeline: GitLab CI/CD with Kubernetes deployment to multiple regional data centres.
- Testing: Automated unit, integration, and UI tests; no security scans in the CI/CD flow.
- Security: The central security team performs quarterly penetration tests.
- Risk: Sensitive customer data, such as payment details, is at risk from supply chain vulnerabilities and misconfigured cloud services.

Trigger for Change:

- During Black Friday, a cyber attacker exploited a vulnerable third-party checkout library and stole partial payment data in one region.
- Regulatory agencies demanded proof of compliance with PCI DSS (Payment Card Industry Data Security Standard) and GDPR (General Data Protection Regulation) within 90 days.
- The organisation's executive leadership announced a complete shift from DevOps to DevSecOps to embed security across all development stages, without compromising productivity or disrupting current processes.

DevSecOps Transformation by Referring to the CPTM Model

The retail organisation aimed to identify likely DevSecOps challenges, apply recommended DevSecOps practices to address those challenges, and select mature tools that can effectively support DevSecOps practices. They also aimed to define sets of appropriate metrics to evaluate the transformation from DevOps to DevSecOps. Hence, the organisation decided to utilise the CPTM Model (Version 2.0) as a reference framework for their DevSecOps transformation.

Ten Phases of DevSecOps Lifecycle:

The model decomposes the DevSecOps lifecycle into ten phases: Plan, Create, Verify, Preproduce, Release, Prevent, Detect, Respond, Predict, and Adapt. These phases are derived from Gartner's DevSecOps model (MacDonald & Head, 2016), which is inspired by the classic DevOps lifecycle (JirehTech, 2016) with eight phases: Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor. Therefore, they are particularly well-suited for the transition from DevOps to DevSecOps. The ten phases form a loop that starts with planning and ends with adapting/re-planning, where each iteration is completed, and the next is improved.

High Priority First (HPF):

The organisation should perform the ten phases in sequence, but choose and employ elements under each phase that align with their current state and needs. In other words, the DevSecOps CPTM Model (Version 2.0) is not equivalent to a phase-based Work Breakdown Structure (WBS), which establishes scheduling relationships between tasks (PMI, 2024). The organisation does not need to apply each practice, tool, or metric to their DevSecOps implementation as a mandatory step, because they may not face all the challenges that are listed in this model.

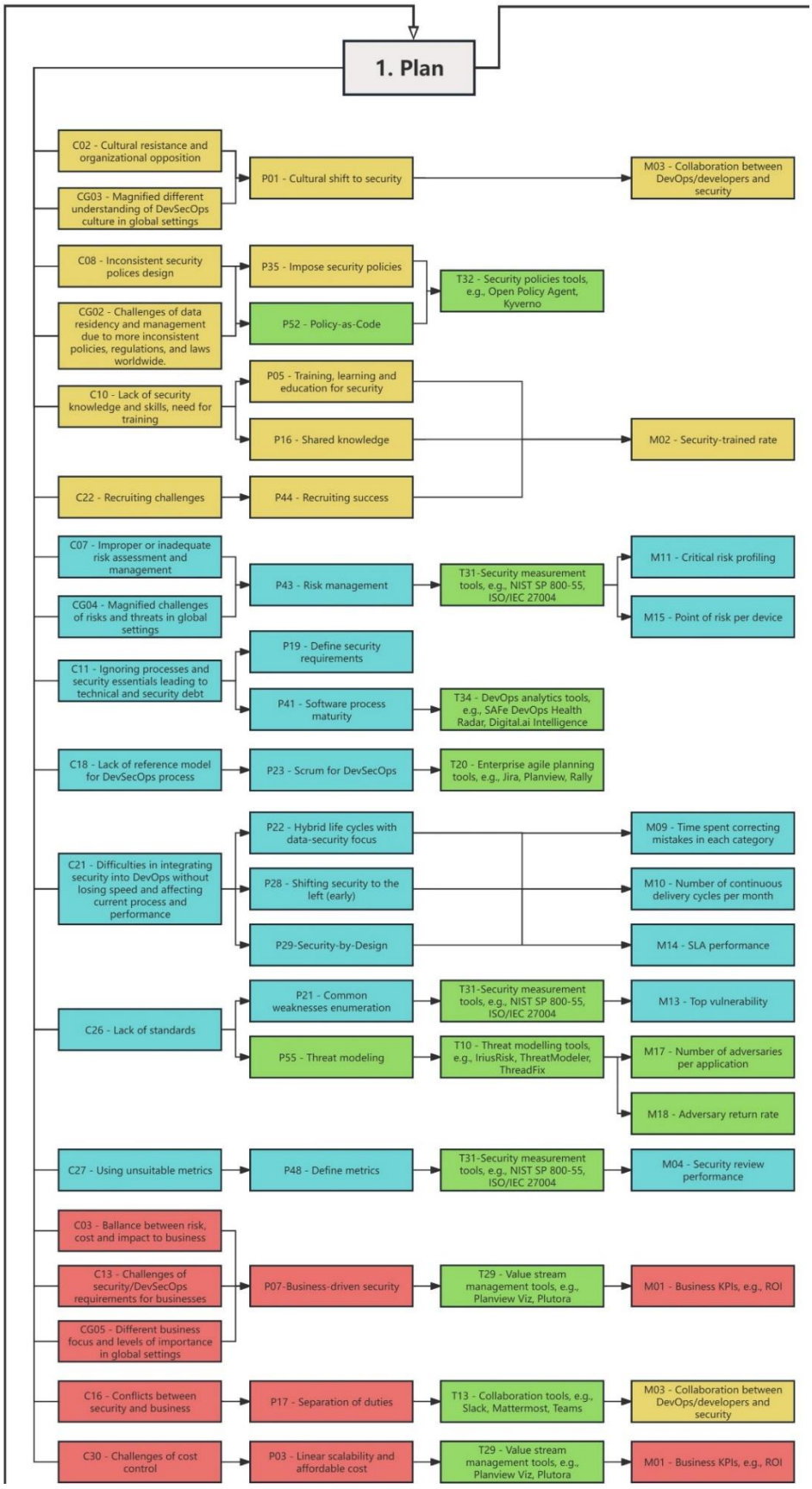
Prioritisation is a key feature of the DevSecOps CPTM Model (Version 2.0), which allows the organisation to select and apply appropriate DevSecOps challenges, practices, tools, or metrics (identifier represents the priority ranking), depending on their realistic needs and following the principle of High Priority First (HPF), meaning that items that are considered the most important and urgent should be addressed before others.

Plan Phase:

As depicted in **Figure 8. 10**, the DevSecOps CPTM Model starts with the "Plan" phase, which requires a comprehensive consideration of DevSecOps challenges, practices, tools, and metrics

within the categories of “Organisation, People and Culture” (OPC), “Process Capabilities” (PC), “Technology”, and “Business”.

Figure 8. 10 - DevSecOps CPTM Model (Version 2.0) – Plan



In particular, many OPC-related challenges (yellow colour) should be addressed as early as possible, e.g., “C02 – Cultural resistance and organisational opposition”, “C08 – Inconsistent security polices design”, and “C10 – Lack of security knowledge and skills, need for training”.

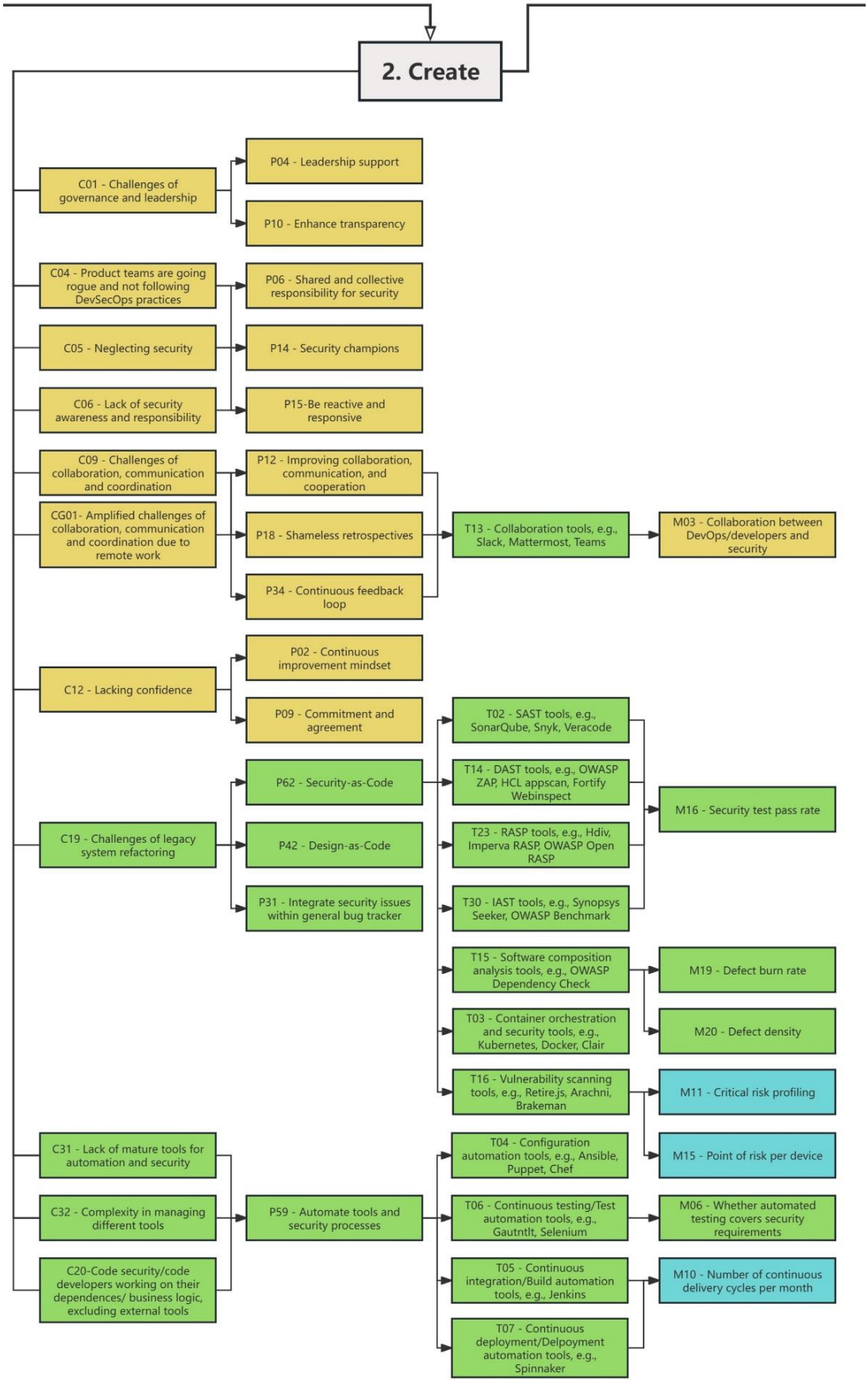
This retail organisation operates a multinational e-commerce platform that involves cross-border payments and logistics. Thus, some global challenges, such as “CG02 – Challenges of data residency and management due to more inconsistent policies, regulations, and laws worldwide” and “CG03 – Magnified different understanding of DevSecOps culture in global settings”, should also be taken into consideration.

Once challenges are identified, the corresponding practices, tools, and metrics can be selected from the model. For example, they can employ “P35 – Impose security policies” and “P52 – Policy-as-Code” to address “C08” and “CG02” and use “T32 – Security policies tools, e.g. Open Policy Agent and Kyverno” to support both practices; they can employ “P05 – Training, learning and education for security” and “P16 – Shared knowledge” to address “C10” and use “M02 – Security-trained rate” to measure the performance of “P16”. The model can also guide the organisation on how to plan its DevSecOps process (blue colour, e.g., risks, standards, metrics, and threats modelling) and the business strategy (red colour, e.g., business focus and budgets).

Create Phase:

Figure 8. 11 depicts the “Create” phase, which involves executing the plan, preparing security processes and practices, and setting up security tools.

Figure 8. 11 - DevSecOps CPTM Model (Version 2.0) – Create



At the beginning of the "Create" phase, the team may still face some OPC-related challenges (yellow colour), e.g., "C01 – Challenges of governance and leadership", "C06 – Lack of security awareness and responsibility", and "C09 – Challenges of collaboration, communication, and coordination".

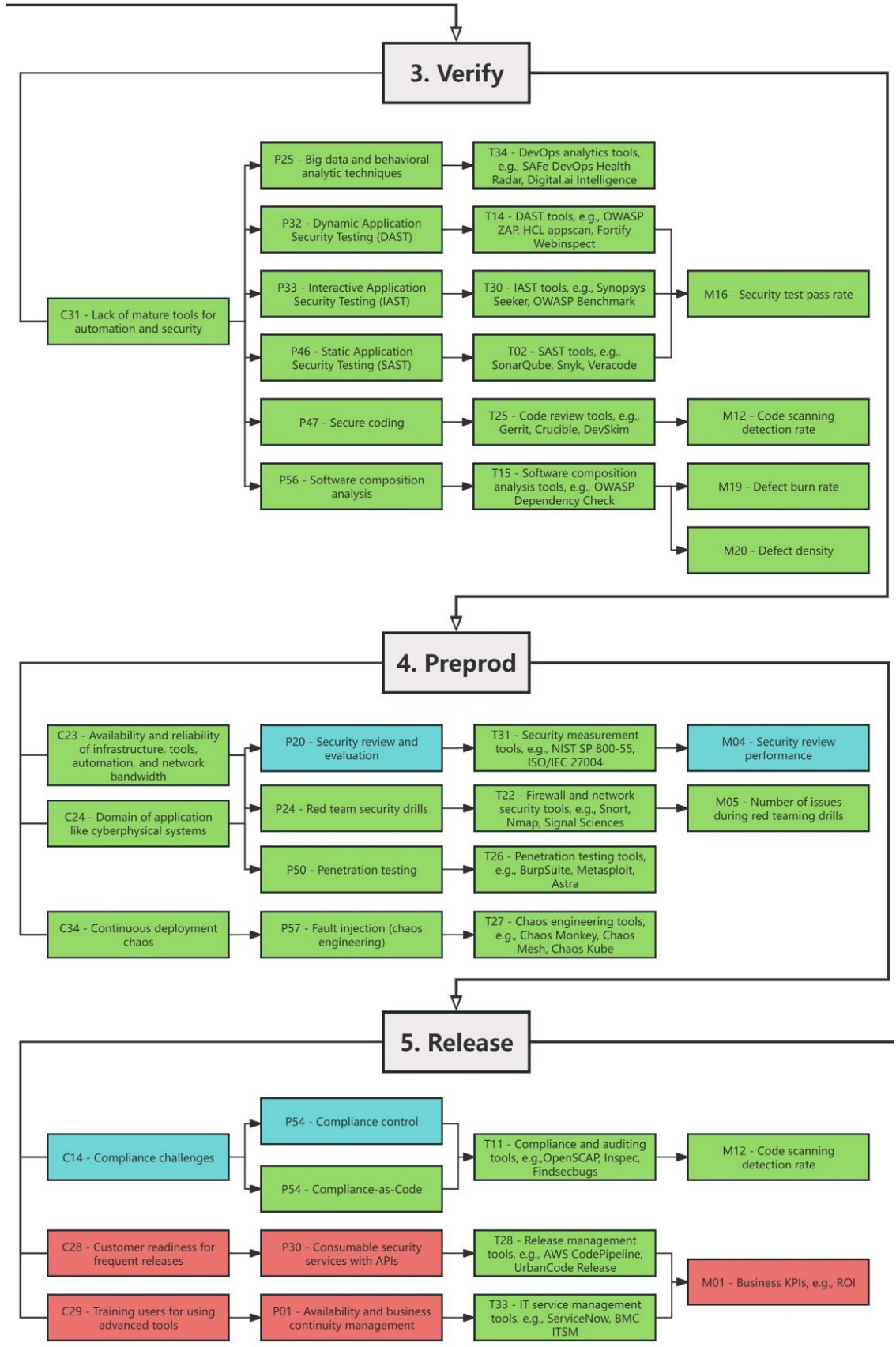
In addition, their globally distributed setup may face "CG01 – Amplified challenges of collaboration, communication, and coordination due to remote work". Hence, the organisation can address these OPC-related challenges with corresponding OPC-related practices, e.g., "P04 – Leadership support" and "P10 – Enhance transparency" for "C01"; "P06 – Shared and collective responsibility for security", "P14 – Security champions", and "P15 – Be reactive and responsive" for "C06"; "P12 – Improving collaboration, communication, and cooperation", "P18 – Shameless retrospectives", and "P34 – Continuous feedback loop" for "C09" and "CG01". "T13 – Collaboration tools" and "M03 – Collaboration between DevOps/developers and security" are tools and metrics for those practices.

To address the technology-related challenges (green colour) such as "C19 – Challenges of legacy system refactoring", "C31 – Lack of mature tools for automation and security" and "C32 – Complexity in managing different tools", some selected tools for automation and security ("P62 – Security-as-Code") need to be set up during the "Create" phase, e.g., SAST, DAST, IAST, RASP, SCA, etc. Given the retail organisation's existing mature DevOps pipeline and automated testing integration, they can skip setting up CI/CD and the test automation in this phase.

Phases of Verify, Preproduce, and Release:

As demonstrated in **Figure 8. 12**, the "Verify" phase implements the security practices and technologies established during the "Create" phase, including SAST, DAST, IAST, RASP, and SCA.

Figure 8. 12 - DevSecOps CPTM Model (Version 2.0) – Verify, Preproduce, and Release



The following "Preproduce" phase includes further security tests, such as fault injection, penetration test, and red team drilling. Both phases in the model involve numerous technology-related security testing practices and tools (green colour), allowing the company to select as needed.

Additionally, as a by-product of the DevSecOps CPTM Model (Version 2.0), a list of DevSecOps tools is presented to provide more best-of-breed tools across the lifecycle (see **Table 8. 1**, or zenodo.org: https://doi.org/10.5281/zenodo.16932278).

The "Release" phase reviews the configuration, infrastructure, network bandwidth, and compliance, ensures the software is ready to be released, and builds it into the production environment. Several business-related challenges, practices, and metrics (red colour) emerge during the "Release" phase, indicating that business considerations are equally important for product release as they are for planning.

These three phases are located in the middle of the lifecycle, preceding and following the product release, and provide the optimal timing for measuring and evaluating DevSecOps performance by defining and utilising appropriate metrics. As discussed above, measuring the performance of each practice at every step of the lifecycle is unnecessary, as monitoring and collecting data for this purpose can increase costs.

Phases of Prevent, Detect, Respond, and Predict:

Figure 8. 13 and **Figure 8. 14** depict four phases that are intimately connected in sequence. The "Prevent" phase protects the runtime environment architecture, the "Prevent" phase continuously monitors and scans the runtime environment architecture, the "Respond" phase addresses the vulnerabilities detected in the previous phases, and the "Predict" phase finally analyses the vulnerabilities to identify the causes. For example, "P49 – Continuous monitoring", "P51 – RASP", "P60 – Container security", and "P63 – Cloud security" can be employed to protect, monitor, and scan the runtime environment architecture. "P08 – Privacy-by-Design", "P13 – Keep credentials safe", "P39 – Privilege management", and "P40 - Sensitive information scan" are for the challenges of access controls.

Figure 8. 13 - DevSecOps CPTM Model (Version 2.0) – Prevent and Detect

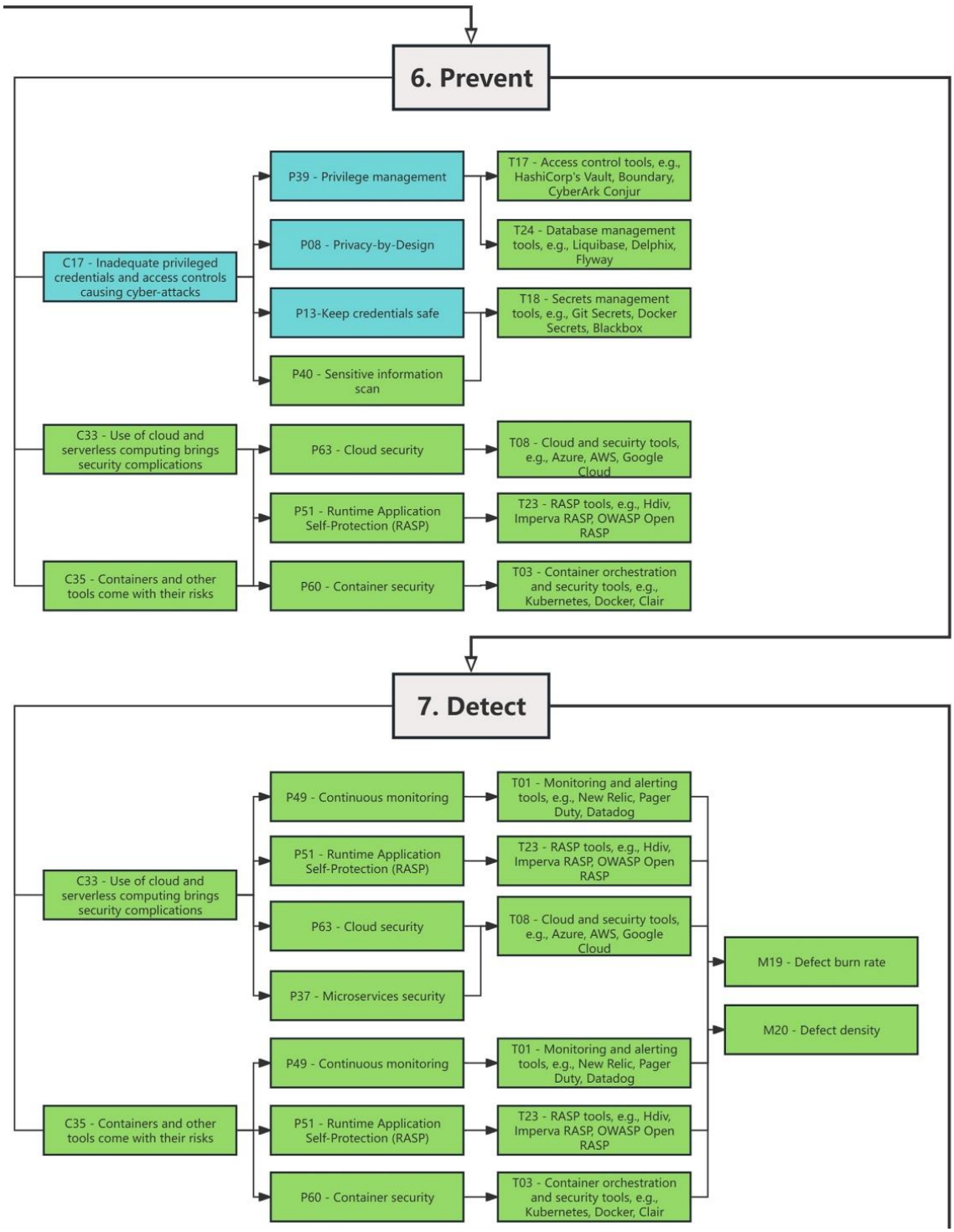
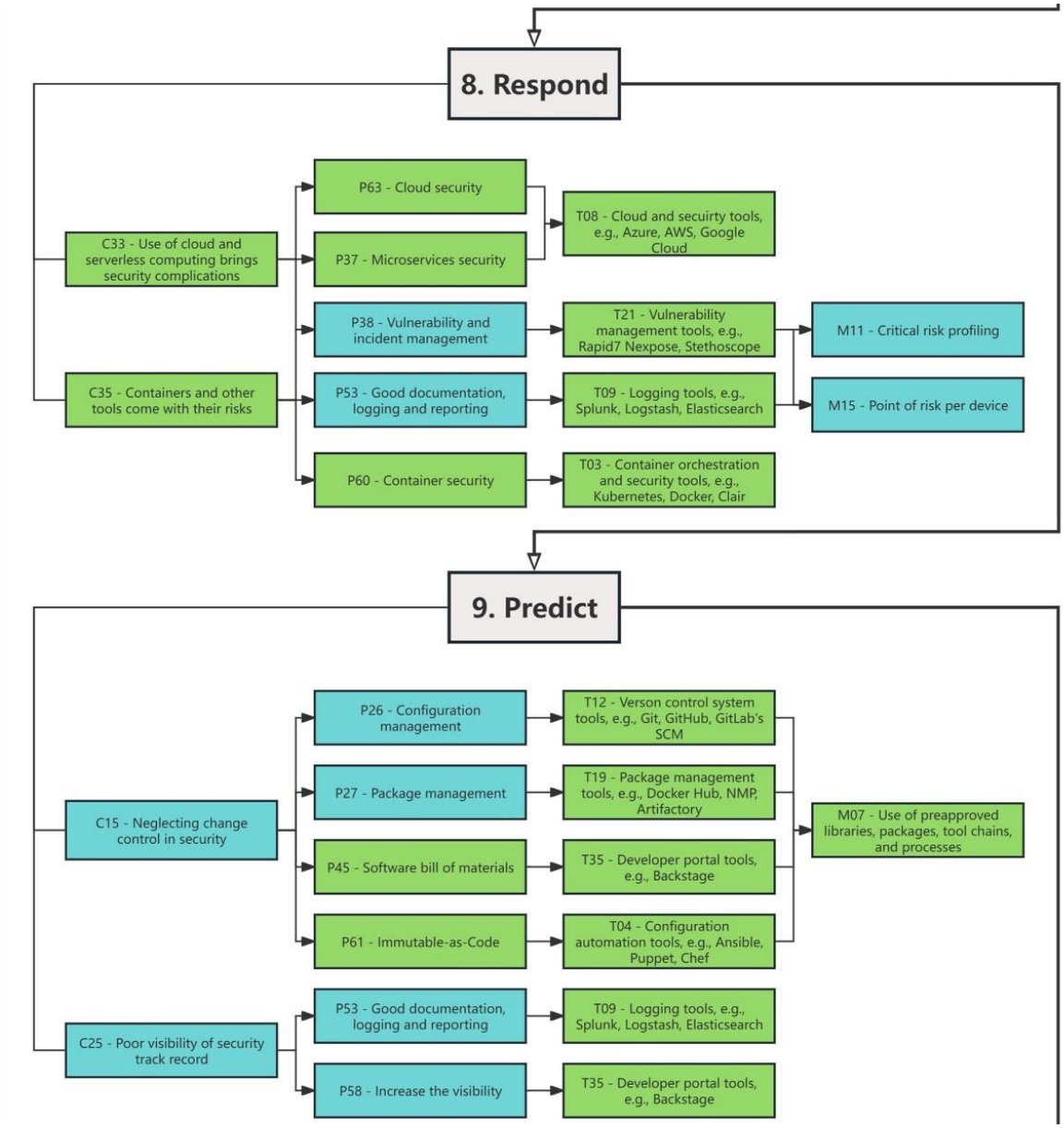


Figure 8. 14 - DevSecOps CPTM Model (Version 2.0) – Respond, and Predict



Adapt Phase and Re-Plan:

The “Adapt” phase is conducted to improve the security process and re-plan the DevSecOps lifecycle, based on lessons learned from all previous phases in the CPTM Model, allowing any practice, tool, or metric to be adapted during this phase. The retail organisation can select multiple metrics during the “Plan” phase to assess DevSecOps performance within this loop and then resume the “Plan” phase to re-plan the subsequent lifecycle.

8.3.4 Further Applications in Digital Transformation

In addition to the migration from DevOps to DevSecOps, the DevSecOps CPTM Model (Version 2.0) can be applied to further contexts, such as security-oriented digital transformations and large-scale agile transformations.

Vial (2019) defines digital transformation (DT) as "a process that aims to improve an entity by triggering significant changes to its properties through combinations of information, computing, communication, and connectivity technologies". Van Veldhoven and Vanthienen (2023) present eight DT guidelines, including digital strategy, business agility, innovation and ambidexterity, modern organisational structure, digital culture, top management support, adequate IT infrastructure, and digital skills. Each guideline is expanded with several best practices of how to implement it in practice. The DevSecOps CPTM Model (Version 2.0) can be applied to digital transformations, and many of its components can map to those DT guidelines and practices.

Digital Strategy:

Digital strategy is the starting point of DT, aiming to plan the organisational change and outline the vision and roadmap that the company is pursuing (Van Veldhoven & Vanthienen, 2023). Hence, deciding to adopt DevSecOps can be regarded as a digital strategy.

Business Agility:

Business agility refers to the ability to change swiftly, encompassing associated practices such as agile development methods, empowering employees, collaborative work, utilising cloud services, and outsourcing (Van Veldhoven & Vanthienen, 2023). The CPTM Model (Version 2.0) provides several DevSecOps practices that correspond, e.g., "P23 – Scrum for DevSecOps", "P09 – Commitment and agreement", "P12 – Improving collaboration, communication and cooperation", "P63 – Cloud security", and "P30 – Consumable security services with APIs".

Innovation and Ambidexterity:

Innovation and ambidexterity can be defined as the combination of digital and physical components to create novel products or services, and embedding them in wider sociotechnical environments (Van Veldhoven & Vanthienen, 2023). In this case, the DevSecOps CPTM Model itself can be seen as an innovation and a source of ambidexterity for a DevSecOps-driven digital

transformation.

Modern Organisational Structure:

The modern organisational structure is characterised by a focus on openness, collaboration, and agility (Van Veldhoven & Vanthienen, 2023). It matches or partly matches the "Organisation, People and Culture" (OPC) category in the DevSecOps CPTM Model.

Digital Culture:

Digital culture refers to creativity, equality, flexibility, openness, a willingness to learn, and digital savviness (Van Veldhoven & Vanthienen, 2023). It also matches the OPC category in the model.

Top Management Support:

Top management support refers to the support and determination of senior management and is considered a key success factor for DT (Van Veldhoven & Vanthienen, 2023). The DevSecOps CPTM Model (Version 2.0) has a perfectly matched practice, "P04 – Leadership support", which is the fourth most important out of 63 practices of DevSecOps.

Adequate IT Infrastructure:

IT infrastructure is the backbone of any digitalisation effort, encompassing various associated practices, including operational backbone, digital services platform, data usage, support for novel technologies, legacy systems, and cybersecurity (Van Veldhoven & Vanthienen, 2023). Cybersecurity is highlighted here. By comparison, the DevSecOps CPTM Model (Version 2.0) has some counterparts such as "P26 – Configuration management", "P22 – Hybrid life cycles with data-security focus", "P25 – Big data and behavioural analytic techniques", and "P31 – Integrate security issues within general bug tracker". By nature, this security-oriented model encompasses various security practices and tools that can be utilised to safeguard infrastructure and protect data privacy against cyberattacks.

Digital Skills:

Digital skills encompass digital strategies, digital technologies, data sciences, and cybersecurity skills, along with associated practices such as recruiting digital talent and continuous education (Van Veldhoven & Vanthienen, 2023). The DevSecOps CPTM Model (Version 2.0) has several

corresponding practices, such as “P44 – Recruiting success”, “P05 – Training, learning and education for security”, and “P16 – Shared knowledge”.

Large-scale Agile Transformation:

Specifically, the DevSecOps CPTM Model (Version 2.0) can also be well-suited for large-scale agile transformations that involve adopting DevSecOps practices. Senapathi and Strode (2025) present a qualitative case study of a large-scale retail organisation that underwent a digital transformation from waterfall to agile methods and sustained that change for six years. By interviewing nine employees of the retail organisation, nine challenges and eight practices were identified. Although this agile transformation case is in a completely different context from the DevSecOps transformation scenario mentioned above, there are still some commonalities between the two transformations worth discussing.

Table 8. 6 and **Table 8. 7** show the correspondences between the identified challenges and practices in the work of Senapathi and Strode (2025) and some of the identified challenges and practices in this research. By comparing them with hiding attributes (e.g., “Agile”, “DevSecOps”, “Security”, and “Risk”), it can be seen that all the challenges and practices correspond, particularly in the aspects of Business, Organisation, People, and Culture. Coincidentally, the matched DevSecOps challenges and practices are those with relatively high rankings (i.e., great importance) in this research.

Table 8. 6 - Comparison of challenges in agile and DevSecOps transformations

Agile Challenges in (Senapathi & Strode, 2025)	**DevSecOps Challenges** in this research
Disparate levels of maturity	C10-Lack of security knowledge and skills, need for training
Resistance to change	C02-Cultural resistance and organizational opposition
Competing priorities	C03-Ballance between risk, cost and impact to business; C16-Conflicts between security and business
Lack of team empowerment	C01-Challenges of governance and leadership
Dependencies between teams	C09-Challenges of collaboration, communication and coordination
Onboarding people	C10-Lack of security knowledge and skills, need for training; C22-Recruiting challenges
Lack of role clarity	C22-Recruiting challenges
Staff turnover	C22-Recruiting challenges
Incompatible funding model	C30-Challenges of cost control

Table 8. 7 - Comparison of practices in agile and DevSecOps transformations

Agile Practices in (Senapathi & Strode, 2025)	**DevSecOps Practices** in this research
Have a clear reason for implementing agile principles	P11-Cultural shift to security
Continuously focus on raising agile awareness	P02-Continuous improvement mindset; P06-Shared and collective responsibility for security
Continuously adjust the business operating model	P04-Leadership support; P09-Commitment and agreement
Change roles	P05-Training, learning and education for security; P16-Shared knowledge
Change the funding model	P03-Linear scalability and affordable cost
Make work visible across the whole organisation	P10-Enhance transparency; P58-Increase the visibility
Support teamwork behaviours	P12-Improving collaboration, communication and cooperation
Monitor; Measure; Feedback; Act	P49-Continuous monitoring; P48-Define metrics; P34-Continuous feedback loop; P15-Be reactive and responsive

As discussed previously in Section 2.2.2, Chapter 2 (Page 23), DevOps, as a crucial part of agile product delivery competency, is hybridised with scaling agile frameworks, such as Disciplined DevOps in DAD (PMI, 2024) and DevOps series in SAFe (SAFe, 2025), while the security focus is not sufficiently addressed. Although those frameworks guide organisations through scaling agile development with DevOps practices and also provide sets of strategies for security requirements (PMI, 2025; SAFe, 2025), they have not presented any reference model for DevSecOps adoption. In light of this, the DevSecOps CPTM Model (Version 2.0) can serve as a well-matched supplement to these industrial frameworks, providing an illustrative guideline for adopting DevSecOps in large-scale or globally distributed settings.

8.4 Chapter Summary

In this chapter, the identification and prioritisation of DevSecOps challenges, practices, tools, and metrics, as well as the findings on the differences between DevSecOps in local and GSE contexts, are summarised and discussed to address the two research questions and associated sub-questions of the Delphi-AHP study.

Based on those findings, the DevSecOps CPTM Model (Version 2.0) is presented. It identifies a broad range of DevSecOps elements in a global software development ecosystem. For practitioners, this conceptual framework guides the adoption of relevant software engineering practices and tools within a collaborative SE approach that involves diverse stakeholders and is mutually beneficial to all within one's software development ecosystem. For researchers, it provides a foundation for future research on new and emerging phenomena (new approaches and technologies) impacting the software engineering approach, such as DevSecOps, collaborative SE, large-scale software development, and global software development.

Next, the final chapter of this thesis concludes the doctoral research, revisits the motivation, outlines the contributions, implications, and limitations, and finally points out future directions.

9 Chapter 9: Conclusion

This chapter concludes the doctoral research by revisiting the motivation, outlining contributions, discussing implications, identifying threats to validity, and suggesting future directions.

9.1 Motivation Revisited

This section revisits the motivation for this PhD thesis by reviewing its research aim, objectives, questions, and findings to examine the overall achievement of the research.

9.1.1 Research Motivation

The motivation for this research was to gain a thorough understanding of DevSecOps by developing an empirically grounded conceptual framework that represents the global DevSecOps approach and helps to identify the adoption of relevant structures (teams, roles) and practices based on the collaborative software engineering method.

9.1.2 Research Aim

The aim of this research was to provide an in-depth understanding of DevSecOps and its adoption in GSE by developing an empirically grounded conceptual framework.

9.1.3 Research Objectives

The specific objectives of the PhD research were established as follows:

For the theoretical foundation:

- To explore the current state of DevSecOps in the existing white and grey literature, on which to base this research.
- To explore the adoption of DevSecOps in GSE from the existing white and grey literature.
- To establish a conceptual framework of DevSecOps based on the existing literature.

For the validation and refinement:

- To validate and refine the conceptual framework through an empirical investigation.

- To investigate differences between DevSecOps in local and global settings and further upgrade the drafted conceptual framework into a global version that could guide practitioners adopting the DevSecOps approach to support software engineering practices in a GSE setting.
- To investigate whether a consensus or dissent exists on the DevSecOps approach between the SE industry and academia.

9.1.4 Research Design and Implementation

To address these objectives and achieve the overall research aim, this doctoral research was divided into two stages:

- For the theoretical foundation, in Stage One, a Multivocal Literature Review (MLR) study was conducted to identify existing research and practical trends and establish a DevSecOps conceptual framework, which served as a theoretical basis for further research.
- For the validation and refinement of the DevSecOps conceptual framework, in Stage Two, a combined Delphi-AHP study was conducted for an empirical investigation to validate, refine, and upgrade the conceptual framework.

9.1.4.1 Research Questions

The following were two sets of research questions, based on which the decision was made to conduct the MLR study in the first stage of this PhD study, followed by adopting the Delphi-AHP approach in the second stage.

Research/Review Questions of the MLR study:

- ***RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect, and their links) in the existing white and grey literature?***

  Sub-question 1.1: What aspects of DevSecOps can be found in the existing white and grey literature?

  Sub-question 1.2: What themes do these aspects contain?

  Sub-question 1.3: How do the identified aspects and themes link to each other?

- ***RQ2: How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?***

Research Questions of the Delphi-AHP Study:

- ***RQ3: How do the experts prioritise the identified challenges, practices, tools, and metrics of DevSecOps?*** *(Challenges, practices, tools, and metrics are four aspects of DevSecOps, which have been identified by the MLR. They are also four elements of the conceptual model.)*

  Sub-question 3.1: What additional DevSecOps challenges, practices, tools, and metrics could be collected from the experts?

  Sub-question 3.2: Do experts have dissent on the prioritisation due to their different roles (e.g., academic, industrial, technical, and managerial)?

- ***RQ4: What are the experts' opinions on DevSecOps in GSE contexts?***

  Sub-question 4.1: How is DevSecOps different between local and global settings?

  Sub-question 4.2: What are the additional challenges, practices, tools, and metrics when DevSecOps comes to a global setting?

9.1.4.2 Findings of MLR Study (Stage One)

As reported in Chapter 3, the MLR study identified five major aspects of DevSecOps (Definitions, Challenges, Practices, Tools/Technologies, and Metrics/Measurement), collected related codes and themes of each aspect, and developed the initial conceptual framework named "DevSecOps CPTM Model (Version 1.0)" by integrating the themes of the latter four aspects. Additionally, an unexplored area relating to the adoption of DevSecOps in the GSE context was identified.

9.1.4.3 Findings of Delphi-AHP Study (Stage Two)

Subsequently, three iterations of the Delphi-AHP study and the findings have been reported in Chapters 5, 6, and 7. This empirical investigation evaluated and prioritised the elements (challenges, practices, tools, and metrics) of the DevSecOps conceptual framework, and included new items for each element. It also investigated the adoption of DevSecOps in GSE and identified that DevSecOps is perceived (slightly) differently in the local and global contexts. In addition, the empirical investigation showed that dissenting opinions on DevSecOps exist between SE academia and industry. Chapter 8 summarises and discusses the findings and presents the DevSecOps Challenge-Practice-Tool-Metric (CPTM) Model (Version 2.0).

9.2 Research Contributions

This research has made several key contributions that are categorised into three domains proposed by McGrath and Brinberg (1983): Conceptual domain, Methodological domain, and Substantive domain.

9.2.1 Contributions in the Conceptual Domain

According to McGrath and Brinberg (1983), the conceptual domain contains elements that are concepts, and relations between elements that are essentially conceptual models about patterns of concepts.

This research presents a conceptual framework (i.e., the DevSecOps CPTM Model) that identifies four elements of DevSecOps (i.e., challenges, practices, tools, and metrics), prioritises the identified elements, and depicts the relationship between those elements. In addition, the DevSecOps CPTM Model (Version 2.0) incorporates the global dimension of DevSecOps.

This model reflects the current state of DevSecOps, captures past and current knowledge and experience, and provides a broad landscape with a prioritised breakdown of DevSecOps elements. Hence, the model can be adopted by researchers to guide future DevSecOps studies. It can also serve as a helpful roadmap for practitioners to improve their DevSecOps theory and practices.

9.2.2 Contributions in the Methodological Domain

McGrath and Brinberg (1983) define the methodological domain as containing elements that are methods, instruments, or techniques for making observations, and relations that are structures or comparison models for comparing sets of observations.

This research validates its innovative research design and confirms its relevance and practicality for investigating research gaps within the SE discipline. Combining SLR/MLR studies with surveys and decision-making approaches is a classic research methodology, which has been employed by many academic papers on DevOps and DevSecOps topics, for example, (Akbar et al., 2020; Akbar, Rafi, et al., 2022; Akbar, Smolander, et al., 2022; Greene, 2020; Khan & Shameem, 2020; Rafi et al., 2020; Zhou et al., 2023; Zohaib, Alsanad, & Abdullah Alhogail, 2024).

Compared to SLR, MLR is advocated as a valuable approach to investigate practice-driven topics in the SE domain (Garousi, Felderer, & Mäntylä, 2019). DevSecOps is an emerging and rapidly evolving field primarily driven by industry needs (Allspaw & Hammond, 2009). A reliable DevSecOps conceptual framework should be derived from both researcher- and practitioner-oriented sources to realise this phenomenon. The MLR study in this PhD research has enabled the integration of practical reality with academic research and has demonstrated that MLR is a valuable tool for investigating the SE domain.

Although combining Delphi with AHP remains a novel approach in empirical studies, this research has demonstrated the feasibility of the hybrid-method approach and summarised its appropriateness. Incorporating the experience from this research into lessons learned from the existing literature, it can be summarised that the combination of Delphi and AHP is suitable for the research, which:

- Has limited expert availability or involves geographically dispersed experts, but still pursues credible results.
- Validates sets of findings or outcomes of review studies on a controversial subject with limited information.
- Address a complex problem or a layered system that can be hierarchically structured.
- Tolerates a long period of surveying time and research duration.

In addition, several limitations of pairing Delphi with AHP are identified, and corresponding recommendations are provided, offering a helpful reference for researchers who intend to use similar research methods:

- The quality of results and the efficiency of responses in the first Delphi round may be affected by participants' unfamiliarity with the AHP method and confusion over the questionnaire format. In general, this matter can be continually improved as the Delphi process goes on, so no specific solution is needed.
- Using AHP to build consensus and compromise with tiny groups (fewer than 5 participants) may lead to inconsistencies in comparisons. In contrast, with relatively large groups (over 30 participants), it may result in overly compromised outcomes and undermine the merit of

this method. Although there is no published evidence, experience from this research supports that AHP is the most suitable for small to medium-sized groups (e.g., 8 to 20 participants).

- Comparing a large number of observations within an AHP comparison matrix (over 15) may lead to high inconsistency in AHP comparisons and recency bias among participants. A corresponding solution could be to include additional AHP layers to divert observations and to reduce the volume of each AHP comparison matrix.

9.2.3 Contributions in the Substantive Domain

McGrath and Brinberg (1983) define the substantive domain as containing elements that are events (behaviours in temporal/spatial/situational contexts) and relations that are phenomena (patterns of relations among events). Accordingly, the following contributions are grouped in the substantive domain:

- A Multivocal Literature Review (MLR) protocol is developed and published to ensure MLR's transparency, reproducibility, and objectivity.
- A Multivocal Literature Review (MLR) study on DevSecOps is conducted and published to consolidate, update, and add value to the extant literature.
- This research identifies and prioritises a set of challenges, practices, tools, and metrics of DevSecOps, and depicts their relations by developing a novel conceptual framework, i.e., the DevSecOps CPTM Model, which is built based on the MLR study (Version 1.0), and has been validated and upgraded through the empirical Delphi-AHP study (Version 2.0).
- A comprehensive list of DevSecOps tools is presented, enabling practitioners to select those needed throughout the DevSecOps lifecycle.
- This research addresses the research gap between DevSecOps and Global Software Engineering (GSE) by investigating differences in DevSecOps across local and global settings.
- This research investigates the consensus and dissent on DevSecOps between the SE industry and academia.

9.3 Research Implications

Based on the above contributions, this research can provide implications for both researchers and practitioners working in DevSecOps.

9.3.1 Implications for Researchers

For researchers, this research provides a systematic, state-of-the-art overview of DevSecOps, conducted through an MLR study encompassing both white and grey literature. It identifies the main aspects of DevSecOps studies in the existing literature, i.e., Definition, Challenges, Practices, Tools/Technologies, and Metrics/Measurement. Of these, challenges and practices seem to be of most concern for researchers. Hence, the findings of the MLR study could help researchers to see a body of knowledge and choose research directions in this area.

Moreover, this research employs a combined Delphi-AHP method to validate the MLR findings and create a prioritisation-based conceptual framework covering identified challenges, practices, tools, and metrics within the DevSecOps lifecycle, so that researchers can learn about the detailed implementation and the existing experience of the DevSecOps process, and further consider the most significant aspects of DevSecOps in their future research. With DevSecOps spanning multiple stages of the lifecycle, the proposed framework also enables researchers to identify areas of focus, themes, and lifecycle stages that require further investigation

In addition, the research reveals a scarcity of literature on adopting DevSecOps in GSE contexts and highlights the subtle differences between DevSecOps in local and global settings. This finding helps researchers avoid unnecessary work in this area and identifies a potential research gap.

Methodologically, this research has demonstrated the feasibility of combining the Delphi survey with the AHP approach. Its validity, appropriateness, benefits, limitations, and recommendations have also been summarised to assist researchers in conducting similar empirical studies.

9.3.2 Implications for Practitioners

This research also provides practitioners adopting the DevSecOps paradigm with knowledge and experience. For example, practitioners can refer to the proposed conceptual framework as a roadmap during the execution of DevSecOps projects, as it depicts what practices can be adopted to address corresponding DevSecOps challenges, what structures and tools can be used to support

DevSecOps practices, and what metrics can be applied to measure the performance of those practices and tools. This framework also covers various categories (i.e., OPC, PC, Technology, and Business) to guide DevSecOps teams in considering work items for different roles and from different perspectives, and to identify areas of weakness that could benefit from increased attention. The prioritisation of identified DevSecOps challenges, practices, tools, and metrics will help practitioners identify the most critical options.

In addition, researchers and practitioners have different emphases and strengths in DevSecOps that complement one another. Researchers have summarised the first decade of DevSecOps development, striving to develop frameworks that highlight significant challenges, best practices, relevant tools, and their connections. Simultaneously, the industry is committed to developing and applying more pertinent tools and metrics that can be adopted by DevSecOps teams. The cooperation between academia and industry is expected to be strengthened to achieve unity in DevSecOps work from a variety of perspectives.

9.4 Research Limitations

This section discusses threats to validity in both stages of the PhD research, i.e., the MLR study and the Delphi-AHP study. Limitations of the MLR include study selection bias, subjectivity in quality assessment, data extraction bias, the trustworthiness of the synthesis, and the construction of the search string. The Delphi-AHP study faces several limitations, including participants' unfamiliarity with the methods, inconsistent AHP results, and recency bias during the survey.

9.4.1 Limitations of the MLR Study

The MLR study faces several threats to validity, including study selection bias, subjectivity in quality assessment, data extraction bias, issues with the trustworthiness of data synthesis, and limitations in the construction of the search string.

9.4.1.1 Bias of Study Selection, Quality Assessment, and Data Extraction

Study inclusion/exclusion bias, quality assessment subjectivity, and data extraction bias are the common threats to validity in SE secondary studies (Ampatzoglou et al., 2019). In this MLR study, the PhD researcher identified white and grey literature individually. To mitigate these threats, the

MLR protocol was developed and discussed with supervisors to ensure that the inclusion and exclusion criteria, study quality assessment procedure, and the data extraction form were appropriate and relevant. This MLR protocol was updated over the research timeline to ensure it remained relevant.

Nonetheless, even when guided by the MLR protocol, it cannot be guaranteed that all relevant literature has been included for review and data collection. In light of this, the snowballing technique was adopted to identify other valuable papers that the search strings would not have identified in the selected databases.

9.4.1.2 Trustworthiness of Data Synthesis

The coding and theming were done by the PhD researcher and reviewed with his supervisors. The reflexive TA approach was adopted for data synthesis in this study, which does not require multiple coders working independently to measure agreement between coders (inter-rater reliability) (Braun & Clarke, 2021). However, the trustworthiness of data synthesis remains a threat that needs to be recognised and assessed. Inevitable biases arising from the researcher's subconscious preferences might undermine the trustworthiness of the synthesis.

In the MLR study, the researcher had certain preconceived notions in this topic, e.g., the elements of DevOps/DevSecOps (Capabilities, Cultural Enablers, and Technological Enablers) (Smeds, Nybom, & Porres, 2015) and the CAMS (Culture, Automation, Measurement, and Sharing) model (Humble & Molesky, 2011), that might have influenced the coding and theming. The existing literature and preconceived notions of DevSecOps may have influenced the identification of the elements of the model.

To ensure the trustworthiness of the findings, TA tasks were reviewed using Braun's checklist (Braun & Clarke, 2021), and the four components of trustworthiness (credibility, confirmability, dependability, and transferability) were assessed (Cruzes & Dyba, 2011a).

- Credibility is significantly concerned with the quality of selected primary studies. A quality assessment was therefore conducted on the selected white and grey literature. Another concern for achieving credibility is the suitability of text segments, particularly long segments, such as definitions of DevSecOps.

- Confirmability focuses on agreement among researchers. Both supervisors of the PhD researcher are experienced scholars. Their recognition can be the key factor of confirmability.
- Dependability refers to the stability of findings. The review results were compared with the findings of other secondary studies on the same topic.
- Transferability is the extent to which the findings can be transferred to other settings. The empirical investigation, using Delphi-AHP methods in the second stage of this research, has validated the transferability of the conceptual framework, i.e., the artefact developed from the MLR findings.

It is important to distinguish between bias and subjectivity, especially in this research, when performing the reflexive TA, which is based on a relativist ontology and a subjective epistemology. Subjectivity, which is caused by researchers' knowledge, experiences, roles, and backgrounds, may affect data collection, theme analysis, and interpretation. However, it should be considered a strength in knowledge production rather than a threat to credibility (Braun & Clarke, 2021).

9.4.1.3 Construction of Search String

Inappropriately constructed search strings can yield redundant or insufficient search results (Ampatzoglou et al., 2019). This MLR study identified a scarcity of primary studies on DevSecOps adoption in global settings. Some terminologies specific to GSE and DevSecOps might have been missed when determining the search strings. Although the search strings were modified multiple times with additional keywords, the results remained unchanged. Hence, it can be confidently stated that the existing white and grey literature lacks a global dimension of DevSecOps. The results of the subsequent empirical investigation, i.e., the Delphi-AHP study, support this claim.

9.4.2 Limitations of the Delphi-AHP Study

The Delphi-AHP study is subject to several limitations, including participants' unfamiliarity with the methods, inconsistency in AHP results, and participants' recency bias in the survey.

9.4.2.1 Participants' Unfamiliarity with Methods

The first round of the Delphi survey often has limitations, notably that the quality of the results and the efficiency of responses may be affected by participants' confusion about the questionnaire

format and unfamiliarity with the research method. In particular, this research employed a combination of Delphi and AHP, a complex approach.

To mitigate the issue, participants were provided with a detailed explanation of the research design and the AHP approach before the first survey round. In general, this matter could have been continually mitigated as the Delphi process progressed.

In addition, the minimum number of AHP comparisons was used rather than completing all pairwise comparisons in the following survey rounds. Once participants had completed these comparisons, the researcher calculated the remaining comparison judgments. Minimising the number of AHP comparisons eliminated redundant questions, reduced unnecessary workloads and survey durations for participants, and mitigated inconsistencies caused by redundant pairwise comparisons (Mu & Pereyra-Rojas, 2017).

According to participants' feedback, they gradually adapted to the survey, and response efficiency improved significantly in the second and third survey rounds.

9.4.2.2 Inconsistency of AHP Results

The Consistency Ratio (CR) of AHP comparison matrices was calculated to evaluate and ensure consistency. Saaty (1982) states that the "reasonable" value of CR is lower than 0.2, and the "acceptable" value is lower than 0.1. Saaty (1982) suggests that the comparison process should be refined and repeated if its CR exceeds 0.2. In this research, a threshold of 0.1 was adopted as the "acceptable" consistency value, with occasional CR values between 0.1 and 0.2 tolerated.

In the Round Two survey, for the evaluation of DevSecOps practices, some high inconsistencies (CRs between 0.1 and 0.2) emerged. It could be speculated that having a large number of observations within a single AHP matrix, possibly any more than 15, to be compared, could be the underlying cause of high inconsistencies in AHP comparisons.

This research employed a combination of the AHP method and Delphi survey, allowing multiple survey rounds for participants and the researcher to refine and repeat the AHP comparison process if an unacceptable CR occurred. In the following survey round, the AHP structure and questionnaire were revised by adding an AHP layer of sub-categories to reduce the volume of observations in a single AHP comparison matrix (not more than 10). The results showed that the

issue of high inconsistencies was resolved.

9.4.2.3 Participants' Recency Bias

In the Round Two survey evaluating DevSecOps practices, another limitation emerged: many participants displayed recency bias when rating and ranking practices, reacting more heavily to the latter-ordered practices than to the former-ordered ones.

Aren (2019) defines recency bias as the tendency to regard newly presented information as more important than earlier information without justification and to assign inconsistent significance to recent items when making decisions. Fudenberg and Levine (2014) state that recency bias can result from limited short-term memory for past observations.

Thus, similar to the issue of inconsistencies, recency bias was also caused by the large volume of observations in the AHP comparison matrices. The same solution was applied to the following survey round, and the results showed that the recency bias was effectively mitigated.

9.5 Future Work

In future work, case studies or observations can be conducted with multiple global software vendors, focusing on the cloud deployment of their software products, to validate the efficacy of the proposed conceptual framework in various GSE contexts.

Furthermore, the State of DevOps Report by Puppet (2023) highlights a recent industry trend: many organisations no longer use the term 'DevOps' because they have internalised its culture and are focusing more on engineering and technology. Hence, mature DevOps organisations tend to use 'Platform Engineering', which is "the discipline of designing and building self-service capabilities to minimise cognitive load for developers and to enable fast flow software delivery." Their latest report (Puppet, 2025) highlights that the most significant DevOps trend in 2024 was security and anticipates that "Platform Engineering will remain at the centre of the security and compliance conversation" to achieve productivity in harmony with security.

Thus, a potential future direction may be to extend the DevSecOps topic by covering security aspects in Platform Engineering if the research trend shifts from DevOps to Platform Engineering.

In fact, two categories in the DevSecOps CPTM Model (Version 2.0), i.e., “Process Capabilities” and “Technologies”, match Platform Engineering to a certain extent, only without using this terminology, as it arose recently.

A further area warranting attention for cybersecurity professionals is the rapidly developing field of Artificial Intelligence (AI) (Chakrabarty et al., 2023). AI-driven security approaches, particularly those leveraging machine learning (ML) or deep learning (DL), are contributing to DevSecOps practices and tools, such as automated security testing, continuous monitoring, and operations (Fu, Pasuksmit, & Tantithamthavorn, 2025). The intersection of DevSecOps and AI-driven security techniques may create opportunities for a revolution in SE security, which warrants systematic study as a future direction. A sensible starting point may be the AI-featured tools and AI-related practices identified in the proposed conceptual framework.

The DevSecOps CPTM Model (Version 2.0) provides a comprehensive understanding of DevSecOps and articulates its current implementation. As DevSecOps continues to mature, the model can be continuously updated to further versions (e.g., 3.0, 4.0, etc.) by incorporating new findings from subsequent case studies and new knowledge in Platform Engineering and AI fields.

References

Aengenheyster, S., Cuhls, K., Gerhold, L., Heiskanen-Schüttler, M., Huck, J., & Muszynska, M. (2017). Real-time Delphi in practice: A comparative analysis of existing software-based tools. *Technological Forecasting and Social Change*, *118*, 15-27. https://doi.org/10.1016/j.techfore.2017.01.023

Ahmed, Z., & Francis, S. C. (2019). *Integrating security with devsecops: Techniques and challenges* 2019 International Conference on Digitization, Sharjah, United Arab Emirates.

Akbar, M. A., Mahmood, S., Shafiq, M., Alsanad, A., Alsanad, A. A.-A., & Gumaei, A. (2020). Identification and prioritization of devops success factors using Fuzzy-AHP approach. *Soft Computing*, *27*(4), 1907–1931. https://doi.org/10.1007/s00500-020-05150-w

Akbar, M. A., Rafi, S., Alsanad, A. A., Qadri, S. F., Alsanad, A., & Alothaim, A. (2022). Toward successful devops: A decision-making framework. *IEEE Access*, *10*, 51343–51362. https://doi.org/10.1109/access.2022.3174094

Akbar, M. A., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*, *147*(1), 1-21. https://doi.org/10.1016/j.infsof.2022.106894

Alharahsheh, H. H., & Pius, A. (2020). A Review of key paradigms: positivism VS interpretivism. *Global Academic Journal of Humanities and Social Sciences*, *2*(3), 39-43. https://doi.org/10.36348/gajhss.2020.v02i03.001

Ali, Y., Butt, M., sabir, M., Mumtaz, U., & Salman, A. (2017). Selection of suitable site in Pakistan for wind power plant installation using Analytic Hierarchy Process (AHP). *Journal of Control and Decision*, *5*(2), 117–128. https://doi.org/10.1080/23307706.2017.1346490

Allspaw, J., & Hammond, P. (2009). *10+ Deploys Per Day: Dev and Ops Cooperation at Flickr*. Retrieved 10/10/2025 from https://tech-talks.code-maven.com/ten-plus-deploys-per-day.html

Alsaqaf, W., Daneval, M., & Anish, P. R. (2021). *Analyzing SAFe Practices with Respect* Product-Focused Software Process Improvement: 22nd International Conference, PROFES 2021, Turin, Italy.

Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., & Chatzigeorgiou, A. (2019). Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Information and Software Technology*, *106*(1), 201-230. https://doi.org/10.1016/j.infsof.2018.10.006

Aren, S. (2019). Evaluation of psychological biases in the financial framework. *Journal of Social, Humanities and Administrative Sciences*, *2*(1), 1-25.

Arnold, V., Collier, P. A., Leech, S. A., & Sutton, S. G. (2000). The effect of experience and complexity on order and recency bias in decision making by professional accountants. *Accounting & Finance*, *40*(2), 109-134. https://doi.org/10.1111/1467-629X.00039

Bass, L., Weber, I. M., & Zhu, L. (2015). *DevOps: A software architect's perspective*. Addison-Wesley.

Bastiaansen, C. A. J., & Wilderom, C. P. M. (2021). Agile and generic work values of British vs Indian it workers: A culture-clash case. *Journal of Strategy and Management*, *15*(3), 353–376. https://doi.org/10.1108/jsma-03-2021-0071

Beecham, S., Clear, T., Lal, R., & Noll, J. (2021). Do scaling agile frameworks address risk in global software development? An empirical study. *Journal of Systems and Software*, *171*(110823). https://doi.org/10.1109/icse-seip52600.2021.00037

Beiderbeck, D., Frevel, N., von der Gracht, H. A., Schmidt, S. L., & Schweitzer, V. M. (2021). Preparing,

conducting, and analyzing delphi surveys: Cross-disciplinary practices, new directions, and advancements. *MethodsX*, *101401*. https://doi.org/10.1016/j.mex.2021.101401

Betts, D. (2022). *3 essential steps to enable security in DevOps*. Retrieved 23/4/2025 from https://www.gartner.com/en/documents/4261699

Biedenbach, T., & Müller, R. (2011). Paradigms in project management research: Examples from 15 years of IRNOP conferences. *International Journal of Managing Projects in Business*, *4*(1), 82–104. https://doi.org/10.1108/17538371111096908

Bouzon, M., Govindan, K., Rodriguez, C. M. T., & Campos, L. M. S. (2016). Identification and analysis of reverse logistics barriers using Fuzzy Delphi method and AHP. *Resources, Conservation and Recycling*, *108*, 182–197. https://doi.org/10.1016/j.resconrec.2015.05.021

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Braun, V., & Clarke, V. (2020). *Thematic analysis: a reflexive approach*. https://www.psych.auckland.ac.nz/en/about/thematic-analysis.html

Braun, V., & Clarke, V. (2021). One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qualitative Research in Psychology*, *18*(3), 328-352. https://doi.org/10.1080/14780887.2020.1769238

Brunelli, M. (2015). *Introduction to the analytic hierarchy process*. Springer International Publishing.

Burns, A. A., McDermid, J., & Dobson, J. (1992). On the meaning of safety and security. *The Computer Journal*, *35*(1), 3-15. https://doi.org/10.1093/comjnl/35.1.3

Caniglia, A., Dentamaro, V., Galantucci, S., & Impedovo, D. (2025). FOBICS: Assessing project security level through a metrics framework that evaluates DevSecOps performance. *Information and Software Technology*, *178*, 107605. https://doi.org/10.1016/j.infsof.2024.107605

Carter, K. (2017). Francois Raynaud on devsecops. *IEEE Software*, *34*(5), 93-96. https://doi.org/10.1109/ms.2017.3571578

Carturan, S. B., & Goya, D. H. (2019). *A systems-of-systems security framework for requirements definition in cloud environment* Proceedings of the 13th European Conference on Software Architecture, New York, NY.

Chakrabarty, A., Hanley, M., Daugherty, R., & O'Shea, B. (2023). *Redefining the next decade of cybersecurity: AI-powered security built to empower developers [Plenary Presentation SEC2732M]* (GitHubUniverse, Issue. https://reg.githubuniverse.com/flow/github/universe23/sessioncatalog/page/sessioncatalog/session/1689094392389001bUiL

Cico, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2021). Exploring the intersection between software industry and software engineering education: A systematic mapping of Software Engineering Trends. *Journal of Systems and Software*, *172*. https://doi.org/10.1016/j.jss.2020.110736

Conchuir, E. O., Ågerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: Where are the benefits? *CACM*, *52*(8), 127-131. https://doi.org/10.1145/1536616.1536648

Constantino, K., Zhou, S., Souza, M., Figueiredo, E., & Kästner, C. (2020). *Understanding collaborative software development: An interview study* Proceedings of the 15th international conference on global software engineering,

Cordova-Pozo, K., & Rouwette, E. A. J. A. (2023). Types of scenario planning and their effectiveness: A review of reviews. *Futures*, *149*, 103153. https://doi.org/10.1016/j.futures.2023.103153

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87-114.

https://doi.org/10.1017/S0140525X01003922

Cruzes, D. S., & Dyba, T. (2010). *Synthesizing evidence in software engineering research* Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Bolzano-Bozen, Italy. http://dx.doi.org/10.1145/1852786.1852788

Cruzes, D. S., & Dyba, T. (2011a). Recommended steps for thematic synthesis in software engineering. *2011 International Symposium on Empirical Software Engineering and Measurement*, 275-284. https://doi.org/10.1109/ESEM.2011.36

Cruzes, D. S., & Dyba, T. (2011b). Research synthesis in software engineering: A tertiary study. *Information and Software Technology*, *53*(5), 440-455. https://doi.org/10.1016/j.infsof.2011.01.004

Dajani, J. S., Sincoff, M. Z., & Talley, W. K. (1979). Stability and agreement criteria for the termination of Delphi studies. *Technological Forecasting and Social Change*, *13*(1), 83-90. https://doi.org/10.1016/0040-1625(79)90007-6

Davis, J., & Daniels, K. (2016). *Effective devops: Building a culture of collaboration, affinity, and tooling at scale*. O'Reilly.

Di Zio, S., & Maretti, M. (2013). Acceptability of energy sources using an integration of the delphi method and the analytic hierarchy process. *Quality and Quantity*, *48*(6), 2973–2991. https://doi.org/10.1007/s11135-013-9935-0

Diel, E., Marczak, S., & Cruzes, D. S. (2016). *Communication challenges and strategies in distributed DevOps* 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), Orange County, CA, USA. Orange County, CA, USA

Digital.ai. (2025). *DevSecOps Tools Periodic Table*. Retrieved 31/5/2025 from https://digital.ai/learn/devsecops-periodic-table/

DoD, U. S. D. o. D. (2025). *The State of DevSecOps within the Department of Defense*. https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsStateOf.pdf

Donohoe, H., Stellefson, M., & Tennant, B. (2012). Advantages and limitations of the e-delphi technique. *American Journal of Health Education*, *43*(1), 38–46. https://doi.org/10.1080/19325037.2012.10599216

Dyck, A., Penners, R., & Lichter, H. (2015). *Towards definitions for release engineering and devops* 2015 IEEE/ACM 3rd International Workshop on Release Engineering, Florence, Italy. http://dx.doi.org/10.1109/RELENG.2015.10

Edmundson, C., & Hartman, K. (2022). *SANS 2022 DevSecOps survey: Creating a culture to significantly improve your organization's security posture*. https://www.sans.org/white-papers/sans-2022-devsecops-survey-creating-culture-improve-organization-security/

Everitt, B. S. (2006). *The Cambridge Dictionary of Statistics*. Cambridge University Press.

Farace, D. J., & Schopfel, J. (2010). *Grey literature in library and information studies*. De Gruyter Saury.

Felderer, M., & Carver, J. C. (2017). Guidelines for systematic mapping studies in security engineering. In *Empirical Research for Software Security* (pp. 47-68). CRC Press.

Fernandez, G. P., & Brito, A. (2019). *Secure container orchestration in the cloud: Policies and implementation* Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus. https://doi.org/10.1145/3297280.3297296

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, *123*, 176-189. https://doi.org/10.1016/j.jss.2015.06.063

Franzago, M., Ruscio, D. D., Malavolta, I., & Muccini, H. (2018). Collaborative Model-Driven Software Engineering: A Classification Framework and a Research Map. *IEEE Transactions on Software Engineering*, *44*(12), 1146-1175. https://doi.org/10.1109/TSE.2017.2755039

Fu, M., Pasuksmit, J., & Tantithamthavorn, C. (2025). Ai for devsecops: A landscape and future opportunities. *ACM Transactions on Software Engineering and Methodology*, *34*(4), 1-61.

Fudenberg, D., & Levine, D. K. (2014). Learning with recency bias. *Proceedings of the National Academy of Sciences*, *111*, 10826-10829. http://www.dklevine.com/papers/recency.pdf

Gall, M., & Pigni, F. (2022). Taking DevOps mainstream: a critical review and conceptual framework. *European Journal of Information Systems*, *31*(5), 548-567. https://doi.org/10.1080/0960085X.2021.1997100

Garousi, V., Felderer, M., & Mäntylä, M. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, *106*, 101-121. https://doi.org/10.1016/j.infsof.2018.09.006

Gobet, F., & Clarkson, G. (2004). Chunks in expert memory: Evidence for the magical number four … or is it two? *Memory*, *12*(6), 732-747. https://doi.org/10.1080/09658210344000530

Godet, M. (2000). The Art of Scenarios and Strategic Planning: Tools and Pitfalls. *Technological Forecasting and Social Change*, *65*(1), 3-22. https://doi.org/10.1016/S0040-1625(99)00120-1

Grande, R., Vizcaino, A., & Garcia, F. O. (2024). Is it worth adopting DevOps practices in global software engineering? Possible challenges and benefits. *Computer Standards and Interfaces*, *87*(1). https://doi.org/10.1016/j.csi.2023.103767

Greene, B. (2020). *Developing effective it governance for devops teams: A qualitative delphi study* (Publication Number 2456883129) Capella University]. http://ezproxy.aut.ac.nz/login?url=https://www.proquest.com/dissertations-theses/developing-effective-governance-devops-teams/docview/2456883129/se-2

Guba, E., & Lincoln, Y. (1994). Competing paradigms in qualitative research. *Handbook of Qualitative Research*.

Gupta, R. K., Venkatachalapathy, M., & Jeberla, F. K. (2019). *Challenges in adopting continuous delivery and DevOps in a globally distributed product team: A case study of a healthcare organization* 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), Montreal, QC, Canada. http://dx.doi.org/10.1109/ICGSE.2019.00020

Hasson, F., Keeney, S., & McKenna, H. (2025). Revisiting the Delphi technique - Research thinking and practice: A discussion paper. *International Journal of Nursing Studies*, *168*, 105119. https://doi.org/10.1016/j.ijnurstu.2025.105119

Hemon-Hildgen, A., & Rowe, F. (2022). Conceptualising and defining DevOps: a review for understanding, not a framework for practitioners. *European Journal of Information Systems*, *31*(5), 568-574. https://doi.org/10.1080/0960085X.2022.2100061

Hemon-Hildgen, A., Rowe, F., & Monnier-Senicourt, L. (2020). Orchestrating automation and sharing in DevOps teams: a revelatory case of job satisfaction factors, risk and work conditions. *European Journal of Information Systems*, *29*(5), 474-499. https://doi.org/10.1080/0960085X.2020.1782276

Ho, L.-W., Lie, T.-T., Leong, P. T., & Clear, T. (2018). Developing offshore wind farm siting criteria by using an international Delphi Method. *Energy Policy*, *113*, 53–67. https://doi.org/10.1016/j.enpol.2017.10.049

Hoda, R. (2021). Socio-technical grounded theory for software engineering. *IEEE Transactions on Software Engineering*, *48*(10), 3808-3832. https://doi.org/10.1109/TSE.2021.3106280

Humble, J., & Molesky, J. (2011). DevOps: a software revolution in the making? *CutterIT Journal*, *24*(8), 6-24.

Humphrey, W. S. (1988). *The software engineering process: definition and scope* Proceedings of the 4th international software process workshop on Representing and enacting the software process,

Hussain, W., Clear, T., & MacDonell, S. (2017). *Emerging trends for global DevOps: A New Zealand perspective* 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), Buenos Aires, Argentina. http://dx.doi.org/10.1109/icgse.2017.16

Huttermann, M. (2012). *DevOps for Developers* (1st ed.). Apress. https://doi.org/10.1007/978-1-4302-4570-4

Ishak, A., Asfriyati, & Akmaliah, V. (2019). *Analytical hierarchy process and PROMETHEE as decision making tool: A Review* IOP Conference Series: Materials Science and Engineering, Medan City North Sumatera, Indonesia. https://doi.org/10.1088/1757-899x/505/1/012085

Ishizaka, A., & Lusti, M. (2004). An expert module to improve the consistency of AHP matrices. *International Transactions in Operational Research*, *11*(1), 97-105. https://doi.org/10.1111/j.1475-3995.2004.00443.x

Jabareen, Y. (2009). Building a Conceptual Framework: Philosophy, Definitions, and Procedure. *International Journal of Qualitative Methods*, *8*(4), 49-62. https://doi.org/10.1177/160940690900800406

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). *What is DevOps?: A systematic mapping study on definitions and practices* Proceedings of the Scientific Workshop Proceedings of XP2016, New York, NY, USA. http://dx.doi.org/10.1145/2962695.2962707

Jalali, S., Gencel, C., & Šmite, D. (2010). *Trust dynamics in global software engineering* Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, New York, NY, USA. http://dx.doi.org/10.1145/1852786.1852817

Jiang, H. (2021). *Cybersecurity Domain Map ver 3.0*. Retrieved 23/4/2025 from https://www.linkedin.com/pulse/cybersecurity-domain-map-ver-30-henry-jiang/

JirehTech. (2016). *What is DevOps*. Retrieved 13/12/2024 from http://www.jirehtechconsulting.com/what-is-devops/

Khan, A. A., & Shameem, M. (2020). Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process. *Journal of Software: Evolution and Process*, *32*(10). https://doi.org/10.1002/smr.2263

Kim, J., Kim, C., Kim, G., Kim, I., Abbas, Q., & Lee, J. (2022). Probabilistic tunnel collapse risk evaluation model using Analytical Hierarchy Process (AHP) and delphi survey technique. *Tunnelling and Underground Space Technology*, *120*(104262). https://doi.org/10.1016/j.tust.2021.104262

Kim, M., Jang, Y.-C., & Lee, S. (2013). Application of delphi-AHP methods to select the priorities of WEEE for recycling in a waste management decision-making tool. *Journal of Environmental Management*, *128*, 941–948. https://doi.org/10.1016/j.jenvman.2013.06.049

Kitchenham, B. (2007). *Guidelines for performing systematic literature reviews in software engineering* [EBSE Technical Report](EBSE-2007-01).

Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004). *Evidence-based software engineering* Proceedings of the 26th International Conference on Software Engineering, Edinburgh, UK. http://dx.doi.org/10.1109/ICSE.2004.1317449

Kozłowski, A., Kaliszewski, A., Dąbrowski, J., & Klimek, H. (2021). Virtual network sampling method using linkedin. *MethodsX*, *8*(101393). https://doi.org/10.1016/j.mex.2021.101393

Lal, R., & Clear, T. (2021). *Three levels of agile planning in a software vendor environment* Proceedings of the 2021 Australasian Conference on Information Systems, https://aisel.aisnet.org/acis2021/48/

Lilja, K. K., Laakso, K., & Palomäki, J. (2011). *Using the delphi method* Proceedings of PICMET '11: Technology Management in the Energy Smart World (PICMET), Portland, OR, USA.

Line, M. B., & Rostad, L. (2006). *Safety vs. Security?* Proceedings of the 8th International Conference on

Probabilistic Safety Assessment and Management,

Loo, R. (2002). The delphi method: A powerful tool for strategic management. *Policing: International Journal of Police Strategy & Management*, *27*(4), 762-769. https://doi.org/10.1108/13639510210450677

Loukides, M. (2012). *What is devops?*

Luz, W. P., Pinto, G., & Bonifácio, R. (2018). *Building a collaborative culture: a grounded theory of well succeeded devops adoption in practice* Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Oulu, Finland. https://doi.org/10.1145/3239235.3240299

MacDonald, N., & Head, I. (2016). *Devsecops: How to seamlessly integrate security into devops*. Retrieved 23/4/2025 from https://https://www.gartner.com/en/documents/3463417

Macharis, C., Springael, J., De Brucker, K., & Verbeke, A. (2004). Promethee and AHP: The design of operational synergies in multicriteria analysis. *European Journal of Operational Research*, *153*(2), 307–317. https://doi.org/10.1016/s0377-2217(03)00153-x

Mao, R., Zhang, H., Dai, Q., Huang, H., Rong, G., Shen, H., Chen, L., & Lu, K. (2020). *Preliminary findings about DevSecOps from grey literature* 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS), Macau, China.

McGrath, J. E., & Brinberg, D. (1983). External validity and the research process: A comment on the calder/lynch dialogue. *Journal of Consumer Research*, *10*(1), 115–124. http://www.jstor.org/stable/2488862

Md Arof, A. (2015). The application of a combined Delphi-AHP method in maritime transport research: A review. *Asian Social Science*, *11*(23). https://doi.org/10.5539/ass.v11n23p73

Mezak, S. (2018). *The Origins of DevOps: What's in a Name?* Retrieved 10/10/2025 from https://devops.com/the-origins-of-devops-whats-in-a-name/

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, *63*(2), 81.

Mohan, V., & Othmane, L. B. (2016). *Secdevops: Is it a marketing buzzword? - mapping research on security in devops* 2016 11th International Conference on Availability, Reliability and Security, Salzburg, Austria. http://dx.doi.org/10.1109/ares.2016.92

Mohan, V., Othmane, L. B., & Kres, A. (2018). *Bp: Security concerns and best practices for automation of software deployment processes: An industrial case study* 2018 IEEE Cybersecurity Development (SecDev),

Monin, D. J., & Dewe, P. J. (1994). *Skills in an environment of turbulence* Proceedings of the 1994 Computer Personnel Research Conference on Reinventing IS : Managing Information Technology in Changing Organizations Managing Information Technology in Changing Organizations - SIGCPR '94, New York, NY, USA. https://doi.org/10.1145/186281.186325

Morales, J. A., Scanlon, T. P., Volkmann, A., Yankel, J., & Yasar, H. (2020). *Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment* Proceedings of the 15th International Conference on Availability, Reliability and Security, New York, NY, USA. http://dx.doi.org/10.1145/3407023.3409186

Mu, E., & Pereyra-Rojas, M. (2017). *Practical Decision Making: An Introduction to the analytic hierarchy process (AHP). UsingSuper Decisions V2*. Springer International Publishing.

Myrbakken, H., & Colomo-Palacios, R. (2017). *DevSecOps: A multivocal literature review* Software Process Improvement and Capability Determination, http://dx.doi.org/10.1007/978-3-319-67383-7_2

Nagasundari, S., Manja, P., Mathur, P., & Honnavalli, P. B. (2025). Extensive Review of Threat Models for DevSecOps. *IEEE Access*, *13*, 45252-45271. https://doi.org/10.1109/ACCESS.2025.3547932

NCCoE. (2025). *NIST Special Publication 1800-44A Secure Software Development, Security, and Operations (DevSecOps) Practices (Initial Public Draft) Document Version*. Retrieved 31/08 from https://www.nccoe.nist.gov/projects/secure-software-development-security-and-operations-devsecops-practices#maincontent

Nisha, T., & Khandebharad, A. (2022). Migration from DevOps to DevSecOps: A complete migration framework, challenges, and evaluation. *International Journal of Cloud Applications and Computing (IJCAC)*, *12*(1), 1-15. https://doi.org/10.4018/IJCAC.2022010102

Noorani, N., Zamani, A., Alenezi, M., Shameem, M., & Singh, P. (2022). Factor prioritization for effectively implementing DevOps in software development organizations: A SWOT-AHP approach. *Axioms*, *11*(10). https://doi.org/10.3390/axioms11100498

Omoronyia, I., Ferguson, J., Roper, M., & Wood, M. (2010). A review of awareness in distributed collaborative software engineering. *Software: Practice and Experience*, *40*(12), 1107-1133. https://doi.org/10.1002/spe.1005

Ononiwu, M., Azonuche, T. I., Imoh, P. O., & Enyejo, J. O. (2023). Exploring SAFe Framework Adoption for Autism-Centered Remote Engineering with Secure CI/CD and Containerized Microservices Deployment. *International Journal of Scientific Research in Science and Technology*, *10*(6), 606-626. https://doi.org/10.32628/IJSRST

Packeer Mohamed, S. F., Baharom, F., Deraman, A., Tarawneh, O., & Yusof, Y. (2022). Software process assessment and certification: Application of the analytic hierarchy process for priority determination. *International Journal of the Analytic Hierarchy Process*, *14*(3). https://doi.org/10.13033/ijahp.v14i3.870

PMI. (2024). *Disciplined DevOps*. Retrieved 09/09/2025 from https://www.pmi.org/Microsites/Disciplined%20Agile/Process/Disciplined%20Devops

PMI. (2025). *Security Strategies for DevOps*. Retrieved 09/09/2025 from https://www.pmi.org/disciplined-agile/process/security/devops-strategies

Powell, C. (2003). The Delphi technique: myths and realities. *Journal of Advanced Nursing*, *41*(4), 376-382. https://doi.org/10.1046/j.1365-2648.2003.02537.x

Prates, L., Faustino, J., Silva, M., & Pereira, R. (2019). *DevSecOps metrics* Information Systems: Research, Development, Applications, Education, Cham. http://dx.doi.org/10.1007/978-3-030-29608-7_7

Prates, L., & Pereira, R. (2025). DevSecOps practices and tools. *International Journal of Information Security*, *24*(1), 1-25.

Puppet. (2023). *State of DevOps report 2023*. https://www.puppet.com/success/resources/state-of-platform-engineering

Puppet. (2025). *2024 State of DevOps Report: The Evolution of Platform Engineering*. https://www.puppet.com/resources/state-of-platform-engineering

Qualtrics. (2024). *Qualtrics XM - experience management software*. Retrieved 17/12/2024 from https://www.qualtrics.com/

Rafi, S., Yu, W., Akbar, M. A., Alsanad, A., & Gumaei, A. (2020). Prioritization based taxonomy of DevOps security challenges using PROMETHEE. *IEEE Access*, *8*, 105426-105446. https://doi.org/10.1109/ACCESS.2020.2998819

Ragab, N., Ahmed, A., & AlHashmi, S. (2015, 2015//). *Software Engineering for Security as a Non-functional Requirement* Intelligent Data Analysis and Applications. Proceedings of the Second Euro-China Conference on Intelligent Data Analysis and Applications, ECC 2015., Cham.

https://doi.org/10.1007/978-3-319-21206-7_29

Rahman, A. A. U., & Williams, L. (2016). *Software security in devops: Synthesizing practitioners' perceptions and practices* Proceedings of the International Workshop on Continuous Software Evolution and Delivery, New York, NY, USA. http://dx.doi.org/10.1145/2896941.2896946

Rajapakse, R., & Szabo, C. (2024). Towards Multi-Class Socio-Technical Congruence: Assessing Coordination in Collaborative Software Development Settings. *Available at SSRN 4824803*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4824803

Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and Software Technology*, *141*(1), 1-22. https://doi.org/10.1016/j.infsof.2021.106700

Ramesh, B., Dutta, R., Salian, P., & Anand, G. (2024). End-to-End AI-Driven DevSecOps: A Framework for Risk-Aware Testing, Monitoring, and Lifecycle Optimization. *International Journal of Emerging Research in Engineering and Technology*, *5*(2), 157-164. https://doi.org/10.63282/3050-922X.IJERET-V5I2P116

Rho, S. (2006). Delphi Technique: Professional insight to predict the future. *Korea Research Institute for Human Settlements*(299), 53-62.

Rowe, F. (2014). What literature review is not: Diversity, boundaries and recommendations. *European Journal of Information Systems*, *23*(3), 241-255. https://doi.org/10.1057/ejis.2014.7

Saaty, T. L. (1982). The analytic hierarchy process: A new approach to deal with fuzziness in architecture. *Architect. Sci. Rev.*, *25*(3), 64–69.

Saaty, T. L. (2013). The modern science of multicriteria decision making and its practical applications: The AHP/ANP approach. *Operations Research*, *61*(5), 1101–1118. https://doi.org/10.1287/opre.2013.1197

SAFe. (2025). *SAFe DevOps series*. Retrieved 09/09/2025 from https://framework.scaledagile.com/devops/

Sanchez-Gordon, M., & Colomo-Palacios, R. (2020). *Security as culture: A systematic literature review of DevSecOps* Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, New York, NY, USA.

Schopfel, J. (2010). *Towards a prague definition of grey literature*. https://greynet.org/images/GL12_S1S,_Sch_pfel.pdf

Sebastian, I. M., Ross, J. W., Beath, C., Mocker, M., Moloney, K. G., & Fonstad, N. O. (2020). How big old companies navigate digital transformation. In *Strategic Information Management* (5th ed., pp. 133-150). Routledge. https://doi.org/10.4324/9780429286797-6

Senapathi, M., & Strode, D. E. (2025). An exploratory study of sustaining organisational agility. *Information and Software Technology*, *187*, 107842. https://doi.org/10.1016/j.infsof.2025.107842

Singh, B. (2025). Integrating Threat Modeling In Devsecops For Enhanced Application Security. *Available at SSRN*, *5267976*.

Smeds, J., Nybom, K., & Porres, I. (2015). Devops: A definition and perceived adoption impediments. *Agile Processes in Software Engineering and Extreme Programming*, *212*, 166-177. https://doi.org/10.1007/978-3-319-18612-2_14

Smite, D., Christensen, E. L., Tell, P., & Russo, D. (2023). The Future Workplace: Characterizing the Spectrum of Hybrid Work Arrangements for Software Teams. *IEEE Software*, *40*(2), 34-41. https://doi.org/10.1109/MS.2022.3230289

Soni, M. (2015). *End to end automation on cloud with build pipeline: The case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery* 2015 IEEE International Conference on Cloud Computing in Emerging Markets, Bangalore, India.

Steinmacher, I., Chaves, A. P., & Gerosa, M. A. (2010). *Awareness Support in Global Software Development: A Systematic Review Based on the 3C Collaboration Model* Collaboration and Technology, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-15714-1_15

SuperDecisions. (2023). *Homepage Super Decisions*. Retrieved 23/4/2025 from https://superdecisions.com/

Tamburri, D. A., Razo-Zapata, I. S., Fernández, H., & Tedeschi, C. (2012). *Simulating awareness in global software engineering: A comparative analysis of scrum and agile service networks* 2012 4th International Workshop on Principles of Engineering Service-Oriented Systems, Zurich, Switzerland.

Van Veldhoven, Z., & Vanthienen, J. (2023). Best practices for digital transformation based on a systematic literature review. *Digital Transformation and Society*, *2*(2), 104-128. https://doi.org/10.1108/dts-11-2022-0057

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems*, *28*(2), 118-144. https://doi.org/10.1016/j.jsis.2019.01.003

Vizcaíno, A., García, F., Piattini, M., & Beecham, S. (2016). A validated ontology for global software development. *Computer Standards and Interfaces*, *46*, 66-78. https://doi.org/10.1016/j.csi.2016.02.004

Von der Gracht, H. A. (2012). Consensus measurement in delphi studies. *Technological Forecasting and Social Change*, *79*(8), 1525–1536. https://doi.org/10.1016/j.techfore.2012.04.013

Warth, J., von der Gracht, H. A., & Darkow, I.-L. (2013). A dissent-based approach for multi-stakeholder scenario development — the future of Electric Drive Vehicles. *Technological Forecasting and Social Change*, *80*(4), 566–583. https://doi.org/10.1016/j.techfore.2012.04.005

Washizaki, H., eds. (2024). *Guide to the software engineering body of knowledge (SWEBOK Guide), Version 4.0*. IEEE Computer Society. www.swebok.org

Weiser, O., Kalman, Y. M., Kent, C., & Ravid, J. (2022). 65 competencies: Which ones should your data analytics experts have? *Communications of the ACM*, *65*(3), 58–66. https://doi.org/10.1145/3467018

Whitehead, J., Mistrik, I., Grundy, J., & van der Hoek, A. (2010). Collaborative Software Engineering: Concepts and Techniques. In *Collaborative Software Engineering* (pp. 1-30). Springer. https://doi.org/10.1007/978-3-642-10294-3

Wiedemann, A., Wiesche, M., Gewald, H., & Krcmar, H. (2023). Integrating development and operations teams: A control approach for DevOps. *Information and Organization*, *33*(3), 100474. https://doi.org/10.1016/j.infoandorg.2023.100474

Wohlin, C., Kalinowski, M., Felizardo, K. R., & Mendes, E. (2022). Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. *Information and Software Technology*, *147*(1), 1-12. https://doi.org/10.1016/j.infsof.2022.106908

Xu, Y. (2025). Applications of Modern Software Engineering Methods in Cloud Service Projects-An Example of Charging Piles. *Applied and Computational Engineering*, *146*, 76-82. https://doi.org/10.54254/2755-2721/2025.TJ22234

Zaydi, M., & Nassereddine, B. (2020). DevSecOps practices for an agile and secure it service management. *Journal of Management Information and Decision Sciences*, *23*(2), 134-149. https://www.abacademies.org/articles/DevSecOps-practices-for-an-agile-and-secure-it-service-management-1532-5806-23-2-186.pdf

Zhao, P., Md Ali, Z., & Ahmad, Y. (2023). Developing indicators for sustainable urban regeneration in

historic urban areas: Delphi Method and Analytic Hierarchy process (AHP). *Sustainable Cities and Society*, *99*(104990). https://doi.org/10.1016/j.scs.2023.104990

Zhao, X., Clear, T., & Lal, R. (2024a). *Identifying the primary dimensions of DevSecOps: a multi-vocal literature review* https://doi.org/10.5281/zenodo.10668696

Zhao, X., Clear, T., & Lal, R. (2024b). Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. *Journal of Systems and Software*, *214*(112063). https://doi.org/10.1016/j.jss.2024.112063

Zhou, X., Mao, R., Zhang, H., Dai, Q., Huang, H., Shen, H., Li, J., & Rong, G. (2023). Revisit security in the era of devops: an evidence-based inquiry into DevSecOps industry. *IET Software*, *17*(4), 435–454. https://doi.org/10.1049/sfw2.12132

Zohaib, M., Alsanad, A., & Abdullah Alhogail, A. (2024). Prioritizing DevOps implementation guidelines for sustainable software projects. *IEEE Access*, *12*, 71109–71130. https://doi.org/10.1109/access.2024.3402832

Appendices

Appendix A. Ethics Approval

Auckland University of Technology Ethics Committee (AUTEC)

14 September 2023

Ramesh Lal

Faculty of Design and Creative Technologies

Dear Ramesh

Re Ethics Application: **23/225 Representing global DevSecOps to usefully support software engineering practice**

Thank you for your responses to AUTEC's conditions. Your ethics application has been approved for three years until 14 September 2026.

Standard Conditions of Approval

1. The research is to be undertaken in accordance with the Auckland University of Technology Code of Conduct for Research and as approved by AUTEC.
2. All public facing documents must have the AUTEC approval number and be of a high standard of spelling and grammar. Dates on the Information Sheet(s) and Consent Form(s) must be consistent.
3. Any amendments to the project must be approved by AUTEC prior to being implemented.
4. A progress report is due annually on the anniversary of the approval date.
5. A final report is due at the expiration of the approval period, or, upon completion of project.
6. Any serious or adverse events must be reported to AUTEC, this includes unforeseen issues that might affect continued ethical acceptability of the project.
7. AUTEC grants ethical approval only. You are responsible for obtaining management permission for access from any institution or organisation at which your research is being conducted and you need to meet all ethical, legal, public health, and locality obligations or requirements for the jurisdictions in which the research is being undertaken.

The application number and title need to be referenced on all correspondence related to this project.

All forms are available online http://www.aut.ac.nz/research/researchethics

For any enquiries, please contact ethics@aut.ac.nz

(This is a computer-generated letter for which no signature is required)

The AUTEC Secretariat

Auckland University of Technology Ethics Committee

Cc: gavin.zhao@autuni.ac.nz; tony.clear@aut.ac.nz

Appendix B. Research Tools

Appendix B.1. MLR Protocol

1. Background

DevOps has become a trending technology term and gained popularity in the software industry and academia. It refers to improving the performance of software development operations by incorporating the Software Development (Dev) team and IT Operations (Ops) team in one process (Hussain, Clear, & MacDonell, 2017). Security appears as a key non-functional requirement of software development but is often ignored and devalued in DevOps programs, because of regarding security as an inhibitor to the high velocity required in DevOps (Mohan & Othmane, 2016). Growing importance of security includes privacy to users in larger scale systems, the rise of Software as a Service (SaaS), globally distributed systems, and how incorporating security conflicts with rapid delivery cycles. Specially, the use of containers, cloud and serverless computing brings increasing security complications.

To build security in DevOps, the term "DevSecOps" has been created as a security-oriented variant of DevOps. It aims to integrate security into DevOps processes without losing speed and quality, to effectively reduce risks and address security issues, by promoting the collaboration amongst security, development and operations teams (Zaydi & Nassereddine, 2019).

This study aims to review, document and analyse the current state of DevSecOps; and investigate the application of DevSecOps in the Global Software Engineering (GSE) context. DevSecOps is an increasingly topical concept in both of the academic and industrial settings, so the voices from academia and industry are equally essential. Hence, a Multi-vocal Literature Review (MLR) was conducted by executing a dual-track strategy covering published and unpublished literature, in order to identify recent research and practical trends and to find out opportunities for further research.

2. Research Questions

We will review white and grey literature to identify recent research and practical trends of DevSecOps, aiming to: (a) observe, document and analyse the state of art of DevSecOps; and (b) investigate the application of DevSecOps in the Global Software Engineering (GSE) context. Regarding the research objectives, research questions were posed

- RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect and their links) in the existing (white and grey) literature?
  - Sub-question 1.1: What aspects of DevSecOps can be found in the existing (white and grey) literature?
  - Sub-question 1.2: What themes do these aspects contain?
  - Sub-question 1.3: How do the identified aspects and themes link to each other?
- RQ2. How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?

3. Research Methodology

A Systematic Literature Review (SLR) is a form of secondary study to identify, evaluate, analyse and interpret all of the possible existing literature relevant to particular research questions, and to synthesise these available researches in a fair manner (Kitchenham & Charters, 2007). However, a normal SLR mainly uses formally peer-reviewed and commercially published literature (e.g., journal and conference papers) and is not quite adequate for this research, because we find that there are only a limited number of academic papers available relevant to this topic, after a quick pre-searching process. Therefore, a Multi-vocal Literature Review (MLR) needs to be conducted. MLR is a special form of SLR which does not only use formally peer-reviewed and commercially published literature (called White Literature) but also includes the Grey Literature (any unpublished work such as technical reports, news, websites, blog posts, white papers, speeches, videos, etc) (Garousi, Felderer, & Mäntylä, 2019). Another important reason for conducting MLR is that the investigation of DevOps should contain both of the researcher-oriented and practitioner-oriented sources. Software engineering practitioners outside of academia constantly produce all kinds of grey literature based on their practical experience. Researchers should not ignore these valuable sources of knowledge and information during the process of literature review.

In this case, a protocol for a multi-vocal literature review of DevOps and its security has been developed, based on the guidelines for performing SLR in SE (Kitchenham & Charters, 2007) and guidelines for conducting MLR in SE (Garousi, Felderer, & Mäntylä, 2019).

4. Search Strategy

An exhaustive search strategy for the multi-vocal literature review is presented in this section, after having consulted the guidelines for both SLR and MLR in SE. Some subsections will describe separately, if there are differences between searching white literature and grey literature.

4.1. Source to be searched

4.1.1 Source to be searched for White Literature

- Automatic searching in well-known digital databases: ACM digital library, IEEE Xplore and Scopus.
- Searching in Google Scholar and the digital library of AUT University.
- Snowballing technique will be conducted on selected literature if necessary, so that more relevant studies can be included.

4.1.2 Source to be searched for Grey Literature

- General web searching engine i.e. Google.
- Source with high credibility, such as books, magazines, specialised databases, government reports, white papers, method creators and consultants' websites and case studies.
- Source with medium credibility, such as technical reports, news, Q/A sites (like StackOverflow), Wiki articles, blog posts, presentations and videos.
- Contacting individuals or organisations for un-published work or specialised databases of theses, via multiple methods, such as direct requests, emails and social media.

- Like snowballing in WL searching, backlinks can be navigated either forward or backward to find more relevant information.

4.2. Search Strings

To address RQ1, Search String 1 = *(DevOps AND (security OR secure OR safe)) OR SecDevOps OR DevSecOps*. After applying Search String 1 in all search sources, we find that the results do not include any studies involving global DevSecOps. To address RQ2, we will use an additional Search String 2 = *(devops AND (security OR secure OR safe) OR secdevops OR devsecops) AND ("global software engineering" OR "global software development" OR gse OR gsd OR "globally distributed" OR "distributed software development" OR "distributed software engineering" OR "multi-site" OR "multi-nation" OR "transnational" OR "remote work")*. Search strings might vary according to the differences between databases, because of their acceptability of Boolean operators.

In addition, some limitations need to be preset on the searches. Strings will be searched within Metadata (title, abstract and keywords); the written language should be set to be English; the publication year should be between 2011 and 2021 (as this decade was the period within which we deemed DevOps to have become common and DevSecOps was first mentioned in 2012); and books (chapters), posters and abstracts would be excluded. All results sort by relevance so that searches can finish when the relevance is extremely weak. Based on the search results, the research questions and search terms would be refined; further, search strings would possibly be re-formulated. Meanwhile, a list of key papers will be generated to ensure the reliability and relevancy of search results (Jalali & Wohlin, 2010).

4.3. Study Selection

4.3.1. Study selection criteria

Study selection criteria are defined to ensure that selected studies provide direct evidence about the research questions, including both inclusion and exclusion criteria (Kitchenham & Charters, 2007). There are only selection criteria for White Literature being listed here, because in practice, inclusion and exclusion criteria for Grey Literature usually overlap and can be integrated with study quality assessment (Section 4.4) (Garousi, Felderer, & Mäntylä, 2019). Also, a large piece of GL has no accurate information so that selection should be made according to specific circumstance. Hence, the following inclusion and exclusion criteria apply to all WL, and have some reference value for selecting GL, although not entirely appropriate.

- Inclusion criteria:

  a) The study mentions one or more primary aspects related to the topic of DevSecOps, e.g., definition, challenges, practices/activities/solutions, tools/technologies, metrics/measurement, and global applications;

  b) The study is written in English;

  c) The study is published from 2012;

  d) The study has a clearly stated methodology/research design;

  e) The study has credible source;

- Exclusion criteria:

  a) The study does not have a full-text;

  b) The study is external to the subject area of computer science and software engineering;

  c) The study does not have a rigorous research method to prove the correctness of findings;

  d) Duplicate studies;

  e) Secondary studies (However, we can use them to help validate and complement our results based on primary studies).

4.3.2. Study selection process

- Selection process for White Literature:

  a) Firstly, the defined search strings are applied to search full-text papers in digital databases, i.e., ACM digital library, IEEE Xplore and Scopus database.

  b) Second, inclusion and exclusion criteria are applied to select papers quickly, based on their titles and abstracts.

  c) Duplicates will be removed.

  d) The full-texts of papers need to be read, if difficult to determine inclusion or exclusion based on titles and abstracts.

  e) Snowballing technique can be performed based on the reference lists of included papers, so that more relevant papers can be found.

  f) Finally, stop searching and selecting studies, when data exhaustion.

- Selection process for Grey Literature:

  a) Firstly, web searching engine (Google) is used to search defined strings, then some relevant information (websites, news, white books, blogs, etc) will be found.

  b) Second, all relevant information need be filtered and determined inclusion or exclusion, by browsing and reading them in detail.

  c) Duplicates will be removed.

  d) Backlinks can be navigated either forward or backward to search more relevant information.

  e) Individuals or organisations of GL studies can be contacted for un-published work or specialised databases, if necessary.

  f) Finally, stop searching and selecting studies, when theoretical saturation, effort bounded or evidence exhaustion.

- After performing the selection process for WL and GL, all selected WL studies and GL studies will be combined together and ready for further processing.

4.4. Study Quality Assessment

Study quality assessment is a necessary procedure to provide more detailed inclusion/exclusion criteria; to determine the valid extent of sources; to assess importance of studies; and to minimise bias (Kitchenham & Charters, 2007). The table shows the study quality assessment checklist adapted from MLR guidelines of Garousi et al. (2019). Garousi et al. (2019) only presented quality assessment checklist of grey literature, here is an extension including both white and grey literature. The first 14 questions were grouped into 6 categories and would be answered YES/NO, so the criteria would be marked 0/1. "Literature Type" would be marked on a scale from 0 to 4. Out of 18 points (14+4), the borderline was set as 11 (60% of 18).

Criteria	Questions
Authority of the producer (Measure = 0 or 1)	• Is the author or the publishing organisation reputable? • Has the author published other work in the field? • Does the author have expertise in the area?
Methodology (Measure = 0 or 1)	• Does the work have a clearly stated aim? • Does the work have a stated methodology? • Does work have authoritative and contemporary references? • Are any limits clearly stated?
Objectivity (Measure = 0 or 1)	• Does the work provide objective statements or credible findings? • Is there vested interest? E.g., a tool comparison by authors that are working for particular tool vendor. • Is the conclusion supported by the data?
Publication Date (Measure = 0 or 1)	• Does the work have a clearly stated date?
Novelty (Measure = 0 or 1)	• Does the work have a novel idea or something unique? • Does the work strengthen or refute a current position?
Impact (Measure = 0 or 1)	• For WL, is the author's work cited often? / For GL, is the source viewed/shared/discussed often?
Literature Type (Measure = 0 to 4)	• WL: peer-reviewed academic papers (Measure = 4). • WL: PhD/Master thesis (Measure = 3). • GL with high credibility, such as books, magazines, specialised databases, white papers, method creators and consultants' websites and case studies (Measure = 2). • GL with medium credibility, such as technical reports, news, Q/A sites, blogs, presentations and videos (Measure = 1). • GL with low credibility, like ideas/opinions/thoughts/commentaries without evidences (Measure = 0).

4.5. Data extraction

Data extraction phase is to extract all relevant information from the selected papers. The following data should be extracted from each selected study:

- Paper information: paper id, title, authors, publication year, sources, etc.
- Key data items (e.g., for RQ1 and S1, data like definitions, terms, meanings etc. should be excerpted as key data items).
- Assessing paper by using exclusion/inclusion criteria
- Quality assessment score of the study.

- Context of study: study types.
- Qualitative data extraction

A data extraction form (available on final page) will be designed to accurately record all this information. Most contents of data extraction forms for MLR are similar to those in the SLR guidelines of Kitchenham and Charters (2007). Besides, Garousi et al. (2019) suggest that explicit traceable links between the extracted data and primary sources should be added in data extraction form, because it is helpful to deal with some GL sources without standardised structure.

Kitchenham & Charters (2007) recommend that data extraction process should be performed independently by at least two researchers (One is data extractor, another is data checker). For single researchers such as PhD students, it is necessary to use some checking techniques.

Moreover, during the process of data extraction, multiple publications of the same data should not be included because duplicate reports would cause heavy bias to influence the results. The right way is to use the most completed version (Kitchenham & Charters, 2007). For example, if having a PhD 1st or conference paper, its subsequent journal version would be more highly rated as most thoroughly reviewed (in theory depending on standing of the journal).

4.6. Data synthesis

Data synthesis is a procedure to collate and summarise the results of the included primary studies (Kitchenham & Charters, 2007). In this research data synthesis will be conducted by using Thematic Analysis (TA), which is a method for identifying, analyzing and reporting themes with data, combining qualitative (text segments, codes, themes) and quantitative (frequency statistics) evidences (Braun & Clarke, 2006). Flexibility is a key advantage of TA method, enabling researchers to provide a wide range of analytic options. Compared to other methods, TA is said to be relatively easy to learn and perform, thereby being accessible to inexperienced researchers. Thus, TA is one of the most frequently used methods for data synthesis in SE, 2/3 of the systematic reviews in SE employed TA to synthesise the data from primary studies. The key distinction between TA and another classic synthesis method - Grounded Theory (GT), is that GT uses an ongoing process to code data throughout data collection (Cruzes & Dyba, 2011), while TA is applied after data collection. Another distinction is that GT aims to create a new theory, but TA is used to capture themes and summarise key features based on existing frameworks. Therefore, the latter is a more appropriate choice for this study. Considering the above, TA is selected as the synthesis method.

Braun and Clarke (2021) specify three types of TA: Coding reliability, Codebook, and Reflexive. We are using reflexive TA, which fully embraces qualitative research values and researchers' subjective skills, thereby fitting an experiential (e.g., critical realist, contextualist) and critical (e.g., relativist, constructionist) framing of language, data and meaning. In reflexive TA, analysis is a situated interpretative reflexive process and can be conducted inductively or deductively; coding is open and organic without a coding framework; themes are the final outcome of data coding and iterative theme development. In comparison with 'reflexive TA' (informed by interpretivism, which advocates a relativist ontology and subjective epistemology (Alharahsheh & Pius, 2020)), 'coding reliability TA' is concerned with objective and unbiased coding (informed by positivism, which aims to achieve an objective reality from pure data without human interpretation (Alharahsheh & Pius, 2020)); 'codebook TA' uses their developed hybrid variant of

a structured codebook or coding framework with coding reliability approaches (informed by pragmatism, which is driven by pragmatic demands around pre-determined information needs (Braun & Clarke, 2021)). Except for 'coding reliability TA', agreement between researchers and inter-rater reliability are not required as measures of quality for 'codebook TA' and 'reflexive TA'. Braun and Clarke (2021) rather critically stress that it is "illogical, incoherent and ultimately meaningless" to require coding reliability and bias suppression in reflexive TA, "because meaning and knowledge are understood as situated and contextual, and researchers' subjectivity is conceptualised as a resource for knowledge production, which inevitably sculpts the knowledge produced, rather than a must-be-contained threat to credibility."

Cruzes and Dyba (2006) present four levels of interpretation and abstraction in TA: Text, Code, Themes, and Model. According to their recommended steps, data will be firstly extracted in initial reading, iteratively in consultation with PhD supervisors, and large amounts of raw data will be labeled as codes; subsequently overlap should be reduced and all the codes will be translated into themes; then themes will be further classified into categories; finally, a conceptual model would be built. TA can be performed either by manual methods or using a software programme. We will analyse manually because initial text segments and codes capturing concepts (e.g., the definitions) are so long that they are difficult to fit to software, which favors short and descriptive codes. The TA process will be performed manually, by searching for concepts and highlighting segments of text on included papers, extracting data to word documents, writing notes for potential themes, and making tables for numeric counts.

4.7. Combination of WL and GL

Finally, all the processed data, codes and themes from the white and grey literature will be combined, analysed, reported and discussed, to answer the research questions.

5. Evaluation, Validation and Amendment

The draft of this review protocol will be presented to PhD supervisors (Dr Ramesh Lal & Assoc. Prof. Tony Clear) for evaluation and criticism. According to the SLR guidelines of Kitchenham & Charters (2007), the protocol can be approved to be validated if following conditions are checked to meet:

- The search strings are derived from the research questions.
- The study selection criteria and process for WL and GL are appropriate.
- The data extraction procedure is appropriate to address the research questions.
- The data synthesis procedure is appropriate to answer the research questions.

In addition, there may be some further amendments of this MLR protocol when executing the procedures in new situations. Some necessary changes can make up for deficiencies and improve the current version of this protocol. Each revision of the MLR protocol will be recorded timely and the protocol will be updated accordingly.

6. Dissemination

The final phase of an MLR is dissemination (Reporting the review). This review will be reported in a technical report or in a section of the PhD thesis.

References

Alharahsheh, H. H., & Pius, A. (2020), A review of key paradigms: positivism vs interpretivism, Global Academic Journal of Humanities and Social Sciences 2 (3), 39–43. doi:10.36348/gajhss.2020.v02i03.001.

Braun, V., & Clarke, V. (2006), Using thematic analysis in psychology, *Qualitative Research in Psychology 3* (2), 77–101. doi:10.1191/1478088706qp063oa.

Braun, V., & Clarke, V. (2021), One size fits all? what counts as quality practice in (reflexive) thematic analysis?, Qualitative Research in Psychology 18 (3), 328–352. doi:10.1080/14780887.2020.1769238.

Cruzes, D. S. & Dyba, T. (2011), Recommended steps for thematic synthesis in software engineering, *2011 International Symposium on Empirical Software Engineering and Measurement*, 275–284. doi:10.1109/ESEM.2011.36.

Cruzes, D. S. & Dyba, T. (2011), Research synthesis in software engineering: A tertiary study, Information and Software Technology 53 (5),440–455. doi:10.1016/j.infsof.2011.01.004.

Garousi, V., Felderer, M., & Mäntylä, M. V. (2019), Guidelines for Including Grey Literature and Conducting Multivocal Literature Reviews in Software Engineering, *Information and Software Technology*, *106*, 101–121，doi: 10.1016/j.infsof.2018.09.006.

Gupta, R. K., Venkatachalapathy, M., & Jeberla, F. K. (2019), Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization, *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, Montreal, QC, Canada, 30-34, doi: 10.1109/ICGSE.2019.00020.

Hussain, W., Clear, T., & MacDonell, S. (2017), Emerging Trends for Global DevOps: A New Zealand Perspective, *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*, Buenos Aires, 21-30, doi: 10.1109/ICGSE.2017.16.

Jalali, S., & Wohlin, C. (2010), Agile Practices in Global Software Engineering - A Systematic Map, *2010 International Conference on Global Software Engineering*, 45-54, doi: 10.1109/ICGSE.2010.14

Kitchenham, B., & Charters, S. (2007), Guidelines for Performing Systematic Literature Reviews in Software Engineering, doi: 10.1.1.117.471

Mohan, V., & Othmane, L. B. (2016), SecDevOps: Is It a Marketing Buzzword? - Mapping Research on Security in DevOps, *2016 11th International Conference on Availability, Reliability and Security (ARES),* Salzburg, 542-547, doi: 10.1109/ARES.2016.92.

Rütz, M. (2019), Devops: A Systematic Literature Review, *Seminar Paper*.

Souag, A., Mazo, R., Salinesi, C., and Comyn-Wattiau, I. (2016), Reusable knowledge in security requirements engineering: a systematic mapping study. *Requirements Engineering*, 21:251–283.

Stray, V., Moe, N. B., & Aasheim, A. (2019), Dependency Management in Large-Scale Agile: A Case Study of DevOps Teams, *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 7007-7016, ISBN: 978-0-9981331-2-6, https://hdl.handle.net/10125/60137.

Wilde, N., Eddy, B., Patel, K., Cooper, N., Gamboa, V., Mishra, B., & Shah, K. (2016), Security for DevOps Deployment Processes: Defenses, Risks, Research Directions. *International Journal of Software Engineering & Applications (IJSEA)*, 7(6), 1-16, doi: 10.5121/ijsea.2016.7601.

Zaydi, M., & Nassereddine, B. (2019). DevSecOps practices for an agile and secure it service management. *Journal of Management Information and Decision Sciences*, 22(4), 527-540.

Appendix B.1.1. Search Record

- **ACM Digital Library:**

Search String 1:

[[All: devops] AND [[All: security] OR [All: secure] OR [All: safe]]] OR [All: secdevops] OR [All: devsecops]

Filters: English, Research articles (including proceedings and journals)

Search results: 416 results

(ACM has reported hundreds of results for this search, including many false positives. We could sort results by relevance and finish searching when the relevance is extremely weak, i.e. there are no searching terms in the topic and abstract of papers.)

Search String 2:

[[[All: devops] AND [[All: security] OR [All: secure] OR [All: safe]]] OR [All: secdevops] OR [All: devsecops]] AND [[All: "global software engineering"] OR [All: "global software development"] OR [All: gse] OR [All: gsd] OR [All: "globally distributed"] OR [All: "distributed software development"] OR [All: "distributed software engineering"] OR [All: "multi-site"] OR [All: "multi-nation"] OR [All: "transnational"] OR [All: "remote work"]]

Filters: English, Research articles (including proceedings and journals)

Search results: 97 results

- **IEEE Xplore:**

Search String 1:

(DevOps AND (security OR secure OR safe)) OR SecDevOps OR DevSecOps

Filters: English, Conferences, Journals

Search results: 100 results

Search String 2:

(devops AND (security OR secure OR safe) OR secdevops OR devsecops) AND ("global software engineering" OR "global software development" OR gse OR gsd OR "globally distributed" OR "distributed software development" OR "distributed software engineering" OR "multi-site" OR "multi-nation" OR "transnational" OR "remote work")

Filters: English, Conferences, Journals

Search results: 27 results

- **Scopus:**

Search String 1:

devops AND (security OR secure OR safe) OR secdevops OR devsecops

Filters: English, Conference papers, Articles, CS, Engineering

Search results: 176 results

Search String 2:

(devops AND (security OR secure OR safe) OR secdevops OR devsecops) AND ("global software engineering" OR "global software development" OR gse OR gsd OR "globally distributed" OR "distributed software development" OR "distributed software engineering" OR "multi-site" OR "multi-nation" OR "transnational" OR "remote work")

Filters: English, Conference papers, Articles, CS, Engineering

Search results: 92 results

Appendix B.1.2. Data Extraction Form

Phase1: Paper Selection	Your Response	Comments
Paper Information		
Paper id		ID of paper, including search No. and databases, e.g. S1_IEEE_01, S2_ACM_02.
Paper title		Title of paper, short version suffices
Authors		Authors' names of paper
Publication year		Year of Publication
Key data items		e.g. for RQ1, data like secure, security, SecDevOps, etc. should be excerpted as key data items
Date researcher analysed this paper		When researcher completed this form
Exclusion/Inclusion Criteria (Some criteria have been pre-performed during searching papers, e.g. language, years and subject area.)		
Ex (a): Is the study external to the terms of Search 1 (DevOps security & SecDevOps/DevSecOps)?		Paper needs to focus on DevOps and its security aspects (for RQ1)
Ex (b): Is the study external to the terms of Search 2 (DevOps & security & GSE/GSD)?		Paper also needs to focus on DevOps security and GSE/GSD topic (for RQ1)
Ex (c): Is the study based on personal opinion?		We reject papers without rigorous methodology or research design to prove

		the correctness of findings
Ex (d): Is this a repeated study?		We only include the key study (most comprehensive) because repeated study would bias results
In (a): Is RQ addressed?		Which research question is addressed by the paper
In (b): Is the study from acceptable source		Include: conference and journal papers; Exclude: books (chapters), posters and abstracts.
Quality assessment score of study		To assess and grade the quality
Decision		
Decision status: Accept/Reject/Waiting for Full paper/Don't Know		Define decision status. "Don't know" status will go to peer review and arbitration
Decision based on: Title/Abstract/Introduction/ Conclusion/Methods/Whole Paper/Peer Review/Arbitration		At what point did researcher make decision.
Context of Study		
Type of papers		Indicate type: solution proposals, philosophical papers, evaluation research, validation research, opinion papers, and personal experience papers.
For Empirical Studies Add		
Type of empirical study methods		Indicate type: experiments, survey, interviews, observation, case study, action research, focus groups, etc
Country/Location		List countries involved in the study
Phase2: Qualitative Data Extraction (Go to Phase 2 if paper has passed all criteria in Phase 1 above.)		
Understanding of DevSecOps (RQ1), including: definitions, challenges, practices, tools, and metrics.		RQ1. What is the state of art of DevSecOps in the existing literature? (List as many as you find)
The relationship between GSE and DevSecOps.		RQ2. How does Global Software Engineering relate to DevSecOps? (List as many as you find)
Additional Data/Follow Up		
Other observations or useful quotes found in paper		Record useful texts or exact quotes which can be used in our report
References found in paper/snowballing (to follow up)		Include more relevant studies based on the references of selected papers.

Appendix B.2. Participant Information Sheet

Participant Information Sheet

Date Information Sheet Produced:

18 August 2023

Project Title

A Delphi study on the Challenges, Practices, Tools, and Metrics of DevSecOps (Security in DevOps), determining their importance and adoption using AHP (Analytic Hierarchy Process).

An Invitation:

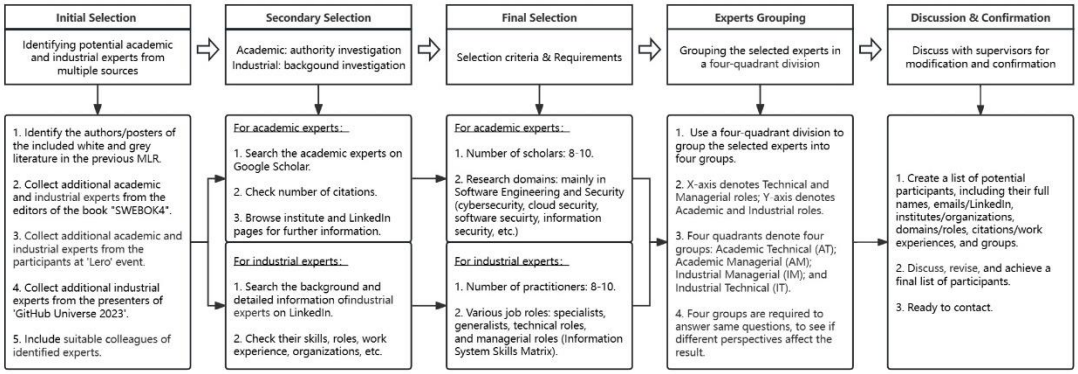
My name is Gavin Zhao. I am currently doing my doctoral research at Auckland University of Technology (AUT), Auckland, New Zealand. An important stage of my research involves a Delphi study on the Challenges, Practices, Tools, and Metrics of DevSecOps (CPTM model), determining their importance and adoption by using the Analytic Hierarchy Process (AHP). I would like to invite you to participate in my research, which will contribute to my PhD qualification. Kindly rest assured that your participation is voluntary, and you may withdraw at any time before the completion of the Delphi survey.

What is the purpose of this research?

We have identified sets of DevSecOps challenges, practices, tools, and metrics through a Multi-vocal Literature Review (MLR) and have built a Challenge-Practice-Tool-Metric (CPTM) model for DevSecOps based on the MLR findings. In addition, the absence of global application of DevSecOps has been identified in the existing white and grey literature. This work is published in the Q1-ranked Journal of Systems and Software: X. Zhao, T. Clear and R. Lal, Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. The Journal of Systems & Software (2024), doi: https://doi.org/10.1016/j.jss.2024.112063. As the following work, the Delphi study aims to validate and refine the main findings (CPTM Model) of the previous MLR. Hence, research objectives are defined to: (1) facilitate an expert discussion on DevSecOps in terms of its Challenges, Practices, Tools/Technologies, and Metrics/Measurement (CPTM); (2) collect experts' opinions on the CPTM model, and refine it; (3) collected experts' opinions on DevSecOps in the Global Software Engineering (GSE) context, and attempt to adapt the CPTM model into a global version. The Analytic Hierarchy Process (AHP) is adopted to obtain the priorities/weights of identified elements in the CPTM model. The main advantage of AHP is that it uses a hierarchical structure to synthesise information quantitatively and logically, so that a complex problem (like the CPTM model with numerous elements in different categories) can be broken down into hierarchies, and the elements in each level can be evaluated by providing better focus during the priority allocation. The successful completion of this Delphi survey is an important part of my doctoral research which will lead to my PhD qualification. The findings of this research may be also used for academic publications and presentations.

How was I identified and why am I being invited to participate in this research?

To match the previous Multi-vocal Literature Review (MLR) and meet research needs, participants/experts are selected from academia and industry respectively, and you have been identified as an expert in the area of study. The expert selection process consists of five parts: Initial Selection, Secondary Selection, Final Selection, Experts Grouping, and Discussion & Confirmation.



Expert Selection Process

The initial selection extracted hundreds of authors/posters' names from included academic papers and grey articles. The frequencies of the extracted names were counted and ranked. We collected additional experts from the editors of the book "Guide to the Software Engineering Body of Knowledge 2024" aka "SWEBOK4", and from the presenters from GitHub Universe 2023. Low response rate and high dropout rate are inevitable problems to a Delphi study, we expected to collect potential participants as many as possible at this stage. The secondary selection was divided into two routes: one was to carried out an authority investigation into the selected academic experts, the other was to investigate the background of the selected industrial experts. For the academic experts, we searched their profiles on Google Scholar, and considered the number of citations as the key factor of authority evaluation. For industrial expert selection, LinkedIn was used to investigate the detailed information.

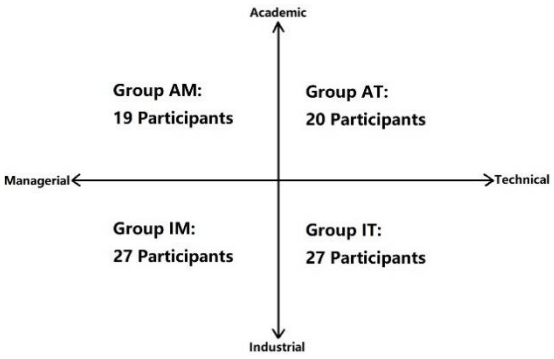
In the final selection, we defined a set of expert selection criteria and requirements, including the number of experts, professional experience, working fields, and domains of expertise. We applied the criteria on the outcome of initial and secondary selection, to group the selected experts. We expected to identify 20 experts, 10 from academia and 10 from industry. However, considering the risks of low response rate and high dropout rate, we expected to collect experts as many as possible at this stage. For academic experts, according to their research focuses and interests provided by Google Scholar, we grouped the research domains of selected scholars into two main subjects, i.e., Software Engineering and Security (including cybersecurity, software security, cloud security, information security, etc), to match the research topic – DevSecOps. For industrial experts, an Information System Skills Matrix presented by Monin and Dewe (1994) divides job roles into four types, namely, specialists, generalists, technical roles, and managerial roles. We identified industrial experts to cover various job roles, to wrestle with the comprehensiveness of this survey by collecting data from different perspectives.

Expert selection criteria

Criteria	Requirements
Number of experts	15-20 experts.

Professional experience	Minimum 5 years.
Field of work	• Academia: 8-10 experts. • Industry: 8-10 experts. • Non-profit organisations (NPO): No formal number required. • Overlapped fields: No formal number required.
Domain of expertise	Academia: Software Engineering and Security. Industry: various job roles in the Information System Skills Matrix (specialist, generalists, technical roles, and managerial roles)

Furthermore, we designed a four-quadrant division to group the selected experts into four groups. X-axis denotes Technical and Managerial roles; Y-axis denotes Academic and Industrial roles. We utilised Technical and Managerial roles from the Information System Skills Matrix (Monin & Dewe,1994), whereas gave up to distinguish between Specialist and Generalist. Redundant groups would reduce the sampling size of each group, and would thereby influence the reliability of data. We also grouped academic experts into Technical and Managerial roles, according to biographies and publications. The four quadrants denote four groups: Academic Technical (AT) – 20 participants; Academic Managerial (AM) – 19 participants; Industrial Managerial (IM) – 27 participants; and Industrial Technical (IT) – 27 participants. The four groups of participants are required to answer same questions, to see if different perspectives affect the result.



Experts Groups

After three rounds of selection and grouping, a list of potential participants was completed (39 academic experts and 54 industrial experts), including their full names, emails/LinkedIn, institutes/organisations, domains/roles, citations/work experiences, and groups. This list was discussed, revised, and finally confirmed within the research team. The Delphi study is conducted anonymously and personal information will be kept confidential at all time.

How do I agree to participate in this research?

Your participation in this research is voluntary (it is your choice) and whether or not you choose to participate will neither advantage nor disadvantage you. You are able to withdraw from the study at any time. If you choose to withdraw from the study, then you will be offered the choice between having any data that is identifiable as belonging to you removed or allowing it to continue to be used. However, once the findings have been produced, removal of your data may not be possible.

What will happen in this research?

The aim of this Delphi study is to determine the importance and adoption of sets of DevSecOps Challenges, Practices, Tools, Metrics captured in a CPTM model, which have been identified from a

previous MLR. Delphi is an iterative process, and there would be three rounds of this Delphi survey. Round 1 is for rating the importance of identified DevSecOps challenges in a AHP format (pairwise comparisons). Based on the result of Round 1, the list of challenges would be revised. Round 2 is to assess the revised challenges and to rate the importance and adoption of the identified DevSecOps practices. Based on the result of Round 2, the list of practices would be revised. Round 3 is to assess the revised practices and a set of identified DevSecOps metrics. Each round would contain 15-30 questions, taking 15-30 minutes.

Survey round	**Estimated time**	**Goal and activities**
Round 1	15 minutes	Rate the importance of DevSecOps challenges in AHP format (pairwise comparisons).
Round 2	30 minutes	Assess the revised challenges (minimise the number of comparisons to reduce time and avoid inconsistency); Rate the importance/adoption of identified DevSecOps practices.
Round 3	30 minutes	Assess the revised practices (minimise the number of comparisons to reduce time and avoid inconsistency); Rate the importance/adoption of identified DevSecOps metrics.

The Analytic Hierarchy Process (AHP) is adopted to obtain the priorities/weights of all elements in the CPTM model. Hence, this research requires the discussions and opinions based on your expertise, knowledge and experience. You will be required to answer numerous pairwise comparison questions to judge the importance and adoption of criteria and sub-criteria, i.e., elements of the CPTM model, by using the fundamental AHP scale of 1-9. Some additional open questions would be included for your opinions on the global application of DevSecOps.

Before conducting the main research, a pilot round of Delphi study has been conducted with several internal and external experts, to refine the survey strategy and questionnaire. The result of the pilot study is satisfactory, so that the feasibility and effectiveness of the methodology is proven. The result also reveals the dissent between participants, that is fully acceptable. Because the research goal is not to achieve a consensus but to see if different roles and perspectives affect the result. Hence, Delphi survey will be stopped once the result reaches a satisfactory stability, instead of consensus. On the other hand, the pilot result reflects certain potential problems, which are common and inevitable to a Delphi study, e.g. the risk of low response rate and high dropout rate during the survey. Therefore, your participation and contributions are greatly appreciated.

What are the discomforts and risks?

There will be no discomforts and risks in the research.

What are the benefits?

Your participation will contribute to the better understanding of the DevSecOps approach, and a refined model of value to practitioners with guidelines for assessing their areas of strength and weakness in their DevSecOps implementation. The research will add value to the existing knowledge of DevSecOps through potential publications and presentation. It will also contribute to my PhD qualification. Therefore, your participation is greatly appreciated.

How will my privacy be protected?

The Delphi study will be conducted in a confidential format, the identity of participants will only be used for creating the expert panel, and will be kept confidential at all time. The survey will be conducted anonymously, so that the research will not analyse the collected data based on individual identification, but on the profession or industry. In 'Active' phase, research data will be stored on Qualtrics, as this data is not sensitive. Consent forms will be stored in OneDrive. In 'Archive' phase, data and consent forms will be securely stored in OneDrive (two different locations respectively) with strict access controls (only researcher and two supervisors have access).

What are the costs of participating in this research?

There is no monetary cost involved. However, it may take 15-30 minutes to complete each round of the survey, and there will be three rounds in total.

What opportunity do I have to consider this invitation?

Please let me know within 14 days from the date of the invitation email. Rest assured that your participation is voluntary, and you may withdraw from the survey at any time before the completion of data collection, without being disadvantaged in any way.

Will I receive feedback on the results of this research?

Yes. We conduct a Delphi study which has multiple rounds of survey and runs iteratively. We will give each round's result back to you, so that you can answer the next round based on previous results. You will receive the summary of findings (a few pages) when the research is completed. You will have option to receive the publications written based on this research. You will have option to be acknowledged by name for the contribution to the research in any research outputs.

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the Project Supervisor, Dr Ramesh Lal, ramesh.lal@aut.ac.nz, +64 9219999 ext 6323.

Concerns regarding the conduct of the research should be notified to the Executive Secretary of AUTEC, ethics@aut.ac.nz, +64 9219999 ext 6038.

Whom do I contact for further information about this research?

Please keep this Information Sheet and a copy of the Consent Form for your future reference. You are also able to contact the research team as follows:

Researcher Contact Details:

Gavin Zhao, gavin.zhao@autuni.ac.nz

Project Supervisors Contact Details:

Dr Ramesh Lal, ramesh.lal@aut.ac.nz ; Associate Prof. Tony Clear, tony.clear@aut.ac.nz

Approved by the Auckland University of Technology Ethics Committee on *18 August 2023*, AUTEC Reference number *23/225*.

Appendix B.3. Consent Form

Consent Form

Project title: A Delphi study on the Challenges, Practices, Tools, and Metrics of DevSecOps (Development, Security & Operation), determining their importance and adoption using AHP (Analytic Hierarchy Process)

Project Supervisors: Dr Ramesh Lal, Assoc. Prof. Tony Clear

Researcher: Gavin Zhao

- ○ I have read and understood the information provided about this research project in the Information Sheet dated 18 August 2023.
- ○ I have had an opportunity to ask questions and to have them answered.
- ○ I understand that taking part in this study is voluntary (my choice) and that I may withdraw from the study at any time without being disadvantaged in any way.
- ○ I understand that if I withdraw from the study then I will be offered the choice between having any data that is identifiable as belonging to me removed or allowing it to continue to be used. However, once the findings have been produced, removal of my data may not be possible.
- ○ I agree to take part in this research.
- ○ I wish to receive a summary of the research findings (please tick one): Yes○ No○
- ○ I wish to be acknowledged for my contribution in any research outputs (please tick one): Yes○ No○

Participant's signature: .....

Participant's name: .....

Participant's Contact Details (if appropriate):

.....

.....

.....

Date:

Approved by the Auckland University of Technology Ethics Committee on 18 August 2023, AUTEC Reference number 23/225.

Note: The Participant should retain a copy of this form.

Appendix B.4. Participant Invitation Letter

Dear potential participants,

My name is Gavin Zhao. I found your contact from your publication or institution. I am currently pursuing my Doctoral research at Auckland University of Technology (AUT), New Zealand, under the supervision of Dr Ramesh Lal and Ass. Prof. Tony Clear. My research topic is DevSecOps (Security in DevOps) and its adoption in global software engineering. An important stage of my research involves a Delphi study on the Challenges, Practices, Tools, and Metrics of DevSecOps, determining their importance and adoption using AHP (Analytic Hierarchy Process). I would like to invite you to participate in my research, to be one of the expert panelists for this Delphi study. Our previous related work can be found at: X. Zhao, T. Clear and R. Lal, Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. The Journal of Systems & Software (2024), doi: https://doi.org/10.1016/j.jss.2024.112063.

Your participation will contribute to the better understanding of the DevSecOps approach, and a refined model of value to practitioners with guidelines for assessing their areas of strength and weakness in their DevSecOps implementation. The research will add value to the existing knowledge of DevSecOps through potential publications and presentations. It will also contribute to my PhD qualification. Thus, your participation is greatly appreciated. Kindly rest assured that your participation is voluntary, and you may withdraw at any time before the completion of the survey.

Delphi is an iterative process and this survey consists of three rounds. Each round of survey would contain 15-30 questions, take 15-30 minutes. The result of each round will be shared with participants and will be also used for designing the following round.

I attached with this email a Participant Information Sheet which describes this invitation and my research, and an Explanation of Terms which defines all technical terms in the research. Please also find a Consent Form for your action, if you can participate in my research. After completing and signing it, please return the form to us, then you will receive a survey link.

Thank you for considering this invitation. I greatly appreciate your time and look forward to your response. Please kindly reply this email within 14 days, to inform me of your decision.

Yours sincerely,
Gavin Zhao
Software Engineering Research Centre (SERC)
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology (AUT), Auckland, New Zealand

Appendix B.5. Initial Potential Experts Selection from MLR

WL Authors (Personal details are available online to the public.)			
Tier	**Name & Email**	**Institute & Country**	**Citations (in Google Scholar)**
Tier 1 (Freq > 3)	Rahman (arahman@ncsu.edu)	North Carolina State University (USA)	1626
	Williams (laurie_williams@ncsu.edu)	North Carolina State University (USA)	21608
	Jaatun (martin.g.jaatun@sintef.no)	SINTEF (Norway)	2921
	Rak (massimiliano.rak@unicampania.it)	University of Campania Studies Luigi Vanvitelli (Italy)	3462
Tier 2 (Freq = 3)	Akbar (azeem.akbar@ymail.com)	Nanjing University of Aeronautics and Astronautics (China)	2191
	Alsanad (aasanad@ksu.edu.sa)	King Saud University (Saudi Arabia)	238
	Benedictis (debenedictis@unina.it)	University of Napoli Federico II (Italy)	1103
	Casola (valentina.casola@unina.it)	University of Napoli Federico II (Italy)	2318
	Iturbe (eider.iturbeg@tecnalia.com)	Fundaci´on Tecnalia Research & Innovation (Spain)	NA
	Rios (erkuden.rios@tecnalia.com)	Fundaci´on Tecnalia Research & Innovation (Spain)	518
	Joosen (wouter.joosen@cs.kuleuven.be)	imec-DistriNet, KU Leuven (Belgium)	15002
	Scandariato (riccardo.scandariato@cse.gu.se)	University of Gothenburg (Sweden)	3462
	Tuma (katja.tuma@cse.gu.se)	University of Gothenburg (Sweden)	372
	Wagner (torrey.wagner@afit.edu)	Air Force Institute of Technology (USA)	175
Tier 3 (Freq = 2)	Bass (lenbass@cmu.edu)	Carnegie Mellon University (USA)	25639
	Morales (jamorales@sei.cmu.edu)	Carnegie Mellon University (USA)	34
	Volkmann (amvolkmann@sei.cmu.edu)	Carnegie Mellon University (USA)	NA
	Yasar (hyasar@sei.cmu.edu)	Carnegie Mellon University (USA)	127
	Jamshidi (pjamshid@cse.sc.edu)	University of South Carolina (USA)	6368
	Parnin (cjparnin@ncsu.edu)	North Carolina State University (USA)	5331
	Cruzes (danielac@sintef.no)	SINTEF (Norway)	4401
	Ferry (nicolas.ferry@sintef.no)	SINTEF (Norway)	1331
	Nguyen (phu.nguyen@sintef.no)	SINTEF (Norway)	590
	Colomo-palacios (colomo-palacios@hiof.no)	Østfold University College (Norway)	8844
	Dullmann	University of Stuttgart (Germany)	152
	Paule	University of Stuttgart (Germany)	NA
	Hoorn (andre.van.hoorn@uni-hamburg.de)	University of Hamburg (Germany)	3086
	Shen (haifeng.shen@scu.edu.au)	Southern Cross University	2099
	Fernandez (gabriel.pereira_fernandez@tu-dresden.de)	Technische Universität (Germany)	13
	Luna (jluna@cs.tu-darmstadt.de)	Technische Universität (Germany)	1079
	Mohan (vaishnavi.mohan@stud.tu-darmstadt.de)	Technische Universität (Germany)	NA
	Othmane (lotfi.ben.othmane@sit.fraunhofer.de)	Fraunhofer SIT (Germany)	1593
	Schork (sebastian.schork@cas.de)	CAS Software AG* (Germany)	12

	Huang (mg1832003@smail.nju.edu.cn)	Nanjing University (China)	138
	Rafi (saeem112@yahoo.com)	Chongqing University of Post and Telecommunication (China)	138
	Yu (wuyu@cqupt.edu.cn)	Chongqing University of Post and Telecommunication (China)	699
	Gupta (rajeevkumar.gupta@siemenshealthineers.com)	Siemens Healthcare Pvt Limited (India) (Industry)	953
	Jeberla (j.ferosekhan@siemenshealthineers.com)	Siemens Healthcare Pvt Limited (India) (Industry)	NA
	Venkatachalapathy (mekanathan.venkatachalapathy@siemens-healthineers.com)	Siemens Healthcare Pvt Limited (India) (Industry)	NA
	Kumar (aknrgk@gmail.com)	Dayananda Sagar College of Engineering (India)	249
	Goyal (denise.goyal@ufabc.edu.br)	Federal University of ABC (Brazil)	NA
	Litoiu (mlitoiu@yorku.ca)	York University (Canada)	8234
	Gumaei (abdugumaei@gmail.com)	King Saud University (Saudi Arabia)	4120
	Sion (laurens.sion@cs.kuleuven.be)	imec-DistriNet, KU Leuven (Belgium)	548
	Villano (villano@unisannio.it)	University of Sannio (Italy)	2315
	Rajapakse (roshan.rajapakse@adelaide.edu.au)	University of Adelaide (Australia)	553
	Zahedi (mansooreh.zahedi@adelaide.edu.au)	University of Adelaide (Australia)	940
	Babar (ali.babar@adelaide.edu.au)	University of Adelaide (Australia)	12963
Global DevOps	Macarthy (r.w.macarthy@edu.salford.ac.uk)	University of Salford (UK)	23
	Bass (j.bass@salford.ac.uk)	University of Salford (UK)	2425
	Diel (elisa.diel@acad.pucrs.br)	PUCRS (Brazil)	NA
	Marczak (sabrina.marczak@pucrs.br)	PUCRS (Brazil)	2694
	Jiménez (miguel@uvic.ca)	University of Victoria (Canada)	NA
	Müller (hausi@uvic.ca)	University of Victoria (Canada)	11981
	Adams (bram.adams@polymtl.ca)	Polytechnique Montreal (Canada)	10411
	Hussain (whussain@aut.ac.nz)	AUT (New Zealand)	512
	Clear (tclear@aut.ac.nz)	AUT (New Zealand)	3061
	MacDonell (stephen.macdonell@vuw.ac.nz)	VUW (New Zealand)	6281
	Florea (ralucamf@ifi.uio.no)	University of Oslo (Norway)	122
	Stray (stray@ifi.uio.no)	University of Oslo (Norway)	1698
	Villegas (nvillega@icesi.edu.co)	Universidad Icesi (Colombia)	3785
	Tamura (gtamura@icesi.edu.co)	Universidad Icesi (Colombia)	3539
	Kerzazi (n.kerzazi@um5s.net.ma)	Mohammed V University in Rabat (Morocco)	429
GL Authors/Posters (Personal details are available online to the public.)			
Tier	**Name**	**Positions/Roles**	**Experience**
Tier 1 (Over 20 years'	Kirstie Magowan (https://www.linkedin.com/in/kirstie-magowan-cbrm-01466611/)	Business Relationship Manger for NZ Police, IT Service Management Consultant for Verso Solutions, Author	Over 20 years

experience)		for ITIL4	
	Sam Bocetta (https://www.linkedin.com/in/sambocetta/)	Security Analyst, Researcher and Evangelist, Retired Network Engineer for Navy	Over 20 years
	Anastasios Arampatzis (https://www.linkedin.com/in/anastasiosarampatzis/)	Cybersecurity Content Writer for Bora	Over 20 years
	James Darwin (https://www.linkedin.com/in/james-darwin/)	Identity and Access Management (IAM) & Cloud Security Solution Architect, CISSP, CCSP at Okta	Over 20 years
	Pete Cheslock (https://www.linkedin.com/in/petecheslock/)	Head of Growth and Community for AppMap, Principal Cloud Economist for the Duckbill Group,	Over 20 years
	Gilad David Maayan (https://www.linkedin.com/in/giladdavidmaayan/)	Strategic SEO/PPC, Content Writer for Technology Markets	Over 20 years
Tier 2 (10-20 years' experience)	Tj Blogumas (https://www.linkedin.com/in/tj-blogumas-mba/)	Cloud/DevOps Architect at Broadcom Inc	Over 10 years
	Mike Spisak (https://www.linkedin.com/in/mike-spisak/)	Distinguished Engineer, Master Inventor, Architect, CTO of the IBM Security Garage	Over 10 years
	Bojana Dobran (https://www.linkedin.com/in/bojanadobran/)	Enterprise Product Marketing Manager at Global Message Services (GMS)	Over 10 years
	Isaac Eldridge (https://www.linkedin.com/in/isaac-eldridge-989a117a/)	Technical Writer at Outshift by Cisco	Over 10 years
	Kev Zettler (https://www.linkedin.com/in/kev-zettler/)	Full Stack Engineer at King	Over 10 years
	Ethan Miller (https://www.linkedin.com/in/ethan-miller-649206188/)	Developer, IT at American Airlines	Over 10 years
	Lucian Constantin (https://www.linkedin.com/in/lconstantin/)	Cybersecurity Journalist for CSOOnline.com	Over 10 years
	Mark Preston (https://www.linkedin.com/in/mgpreston/)	Software Engineer at self-employed	Over 10 years
Tier 3 (Over 5 years' experience)	Christy Maerz (https://www.linkedin.com/in/christy-maerz/)	Content Marketing Manager at AppDynamics	Over 5 years
	Ilai Bavati (https://www.linkedin.com/in/ilai-bavati-0b1a1418a/)	Content Editor at Agile SEO Israel	Over 5 years
	Marius Rimkus (https://www.linkedin.com/in/mariusrimkus/)	Sales Manager at Cherry Servers	Over 5 years
	Ayush Singh (https://www.linkedin.com/in/ayushsingh08/)	Technical Program Manager at Amazon	Over 5 years

Appendix B.6. Final Potential Experts Selection

(SE – Software Engineering; Sec – Security; M – Managerial; T – Technical; G – Generalist; S – Specialist)

Academic experts identified from included white literature (36-1=35) **(Personal details are available online to the public.)**				
Full Name	**Email/LinkedIn**	**Research Interests**	**Domains**	**Citations**
Len Bass	lenbass@cmu.edu https://www.linkedin.com/in/len-bass-7198169/	software architecture, user interface software, software engineering, security architecture, DevOps	SE, Sec T	25682
Laurie Williams	laurie_williams@ncsu.edu https://www.linkedin.com/in/lauriewilliams/	software security, agile software development, continuous deployment, software reliability, software testing and analysis	SE, Sec M	21648
Wouter Joosen	wouter.joosen@cs.kuleuven.be https://www.linkedin.com/in/wouter-joosen-19279113/	security and privacy of distributed software systems, services and applications	Sec M	15039
M. Ali Babar	ali.babar@adelaide.edu.au https://www.linkedin.com/in/ali-babar-5bb4884/	DevOps, security and privacy, empirical software engineering, software architecture	SE, Sec M	12999
Hausi A. Müller	hausi@uvic.ca https://www.linkedin.com/in/hausi/	quantum computing and engineering, software engineering, adaptive systems, cyber physical systems	SE T	11986
Bram Adams	bram.adams@polymtl.ca https://www.linkedin.com/in/bramadams/	software release engineering, software integration, software build systems, software modularity, software maintenance	SE T	10438
Ricardo Colomo-Palacios	ricardo.colomo@upm.es https://www.linkedin.com/in/rcolomo/	software process, software engineering, management information systems	SE M	8883
Marin Litoiu	mlitoiu@yorku.ca https://www.linkedin.com/in/marin-litoiu-92367a7/	adaptive software systems, cloud computing, performance engineering	SE T	8240
Pooyan Jamshidi	pjamshid@cse.sc.edu https://www.linkedin.com/in/pooyanjamshidi/	machine learning, computer systems, autonomous systems	SE T	6383
Stephen MacDonell	stephen.macdonell@vuw.ac.nz https://www.linkedin.com/in/stephenmacdonell/	software engineering, information systems development methodologies	SE M	6286
Chris Parnin	cjparnin@ncsu.edu https://www.linkedin.com/in/chris-parnin/	software engineering, HCI, empirical studies	SE T	5354
Daniela S. Cruzes	daniela.s.cruzes@ntnu.no https://www.linkedin.com/in/daniela-cruzes-72162315/	empirical software engineering, secure software engineering, agile software development, software quality	SE M	4409
Abdu Gumaei	abdugumaei@gmail.com	software engineering, image	SE	4120

	https://www.linkedin.com/in/abdu-gumaei-78130b90/	processing, computer vision, and machine learning	T	
Norha M. Villegas	nvillega@icesi.edu.co https://www.linkedin.com/in/norha villegas/	software engineering, self-adaptive software systems, dynamic context management, cyber physical and digital twin systems	SE M	3789
Gabriel Tamura	gtamura@icesi.edu.co https://www.linkedin.com/in/gabriel ltamura/	software architecture, self-adaptive software systems, software quality attributes	SE T	3543
Riccardo Scandariato	riccardo.scandariato@tuhh.de https://www.linkedin.com/in/riccar do-scandariato-2512991a9/	security and privacy, software engineering	SE, Sec M	3471
Massimiliano Rak	massimiliano.rak@unicampania.it https://www.linkedin.com/in/massi miliano-rak-1262a5/	cloud computing, service level agreement, performance evaluation, computer security, threat modeling	SE, Sec M	3468
André van Hoorn	andre.van.hoorn@uni-hamburg.de https://www.linkedin.com/in/avanh oorn/	automated software engineering, software architecture, software performance engineering, DevOps, distributed systems	SE T	3101
~~Tony Clear~~ (supervisor, conflict of interest)	tony.clear@aut.ac.nz https://www.linkedin.com/in/tony-clear-797ab6/	global software development, global software engineering, collaborative technologies, computing education research, programming	SE M	3062
Martin Gilje Jaatun	martin.g.jaatun@sintef.no https://www.linkedin.com/in/jaatun /	security, cloud, smart grid	Sec M	2926
Sabrina Marczak	sabrina.marczak@pucrs.br https://www.linkedin.com/in/sabrin a-marczak-b785b35/	software engineering, requirements engineering, global software development, collaborative software development	SE M	2694
Julian Bass	j.bass@salford.ac.uk https://www.linkedin.com/in/julian bass/	agile software development, empirical software engineering	SE M	2425
Valentina Casola	valentina.casola@unina.it https://www.linkedin.com/in/valent ina-casola-56113918/	computer security, security evaluation, cloud security, IoT security, critical protection systems	Sec M	2318
Umberto Villano	villano@unisannio.it	distributed computing, cloud security, performance evaluation	Sec T	2315
Muhammad Azeem Akbar	azeem.akbar@ymail.com https://www.linkedin.com/in/muha mmad-azeem-akbar/	empirical software engineering, machine learning, IoT security, quantum application	SE, Sec T	2191
Haifeng Shen	haifeng.shen@scu.edu.au https://www.linkedin.com/in/haifen g-shen-b6aa6287/?originalSubdomain=au	Software engineering, Artificial intelligence, Human-centred computing, Software and application security, Collaborative and social	SE, M	2099

		computing		
Viktoria Stray	stray@ifi.uio.no https://www.linkedin.com/in/viktoriastray/	large-scale agile, agile software development, coordination, meetings, teamwork	SE M	1698
Akond Rahman	akond@auburn.edu https://www.linkedin.com/in/akondrahman/	DevOps, secure software development	SE T	1626
Lotfi ben Othmane	Lotfi.Benothmane@unt.edu https://www.linkedin.com/in/lotfibenothmane/	security	Sec T	1593
Nicolas Ferry	nicolas.ferry@inria.fr https://www.linkedin.com/in/nicolasferry/	security, cloud	Sec T	1331
Alessandra De Benedictis	alessandra.debenedictis@unina.it https://www.linkedin.com/in/alessandra-de-benedictis-452bb217/	security, computer science	Sec M	1103
Jesus Luna	jluna@deeds.informatik.tu-darmstadt.de https://www.linkedin.com/in/jlunagar/	computer security, security metrics, cloud security, grid security	Sec M	1079
Mansooreh Zahedi	mansooreh.zahedi@unimelb.edu.au https://www.linkedin.com/in/mansoorehzahedi/	empirical software engineering, human aspects of SE, software security, AI & SE	SE M	940
Laurens Sion	laurens.sion@cs.kuleuven.be https://www.linkedin.com/in/laurenssion/	threat modeling, secure software engineering, security by design, privacy by design	Sec T	548
Erkuden Rios	erkuden.rios@tecnalia.com https://www.linkedin.com/in/erkuden-rios-24041a3/	security, Continuous risk, industrial IoT, multi cloud	Sec M	518
Katja Tuma	k.tuma@vu.nl https://www.linkedin.com/in/katjatuma/	security-by-design, threat modeling, risk analysis, empirical software engineering, security compliance	Sec M	372
Academic experts identified from the editors of SWEBOK4 (9) (Personal details are available online to the public.)				
Ali Ouni	ali.ouni@etsmtl.ca https://www.linkedin.com/in/ali-ouni-21097b13/	SBSE, refactoring, software engineering, software maintenance and evolution, artificial intelligence	SE T	5865
Alain April	alain.april@etsmtl.ca https://www.linkedin.com/in/alainapril/	software maintenance, SQA, BPM and cloud/bigdata	SE M	4094
Xin Peng	pengxin@fudan.edu.cn https://www.linkedin.com/in/xin-peng-09641a15/	software engineering	SE T	4045
Nobukazu Yoshioka	nobukazuy@acm.org, nobukazu@engineerable.ai	software engineering, security engineering	SE, Sec M	2328

	https://www.linkedin.com/in/nobukazu-yoshioka-11568525/			
Eda Marchetti	eda.marchetti@isti.cnr.it https://www.linkedin.com/in/eda-marchetti-34873a5/	software testing	SE T	2097
Seiji Munetoh	munetoh@jp.ibm.com https://www.linkedin.com/in/seiji-munetoh-69b5601a/	computer science and engineering	Sec T	2049
Maribel Sanchez-Segura	misanche@inf.uc3m.es https://www.linkedin.com/in/maribelsanchezsegura/	Software engineering, software process improvement, knowledge management	SE M	1486
Francis Bordeleau	francis.bordeleau@etsmtl.ca https://www.linkedin.com/in/francis-bordeleau-b2aa273/	software engineering, model-based engineering, open source	SE T	743
Said Daoudagh	said.daoudagh@isti.cnr.it https://www.linkedin.com/in/said-daoudagh-7b41bb84/	data protection, data protection & Compliance, GDPR, access control systems, software testing	Sec T	501
Academic experts enlisted from an event: 'Frontiers of Technology: Cybersecurity – Achievements and Challenges', at Lero, University of Limerick (3) (Personal details are available online to the public.)				
Tiziana Margaria	tiziana.margaria@ul.ie https://www.linkedin.com/in/tiziana-margaria-9044a12/	service engineering of process-intensive system, model driven generative development, formal methods, compliance and governance	SE M	7273
John Noll	John.Noll@ul.ie jhnoll@gmail.com https://www.linkedin.com/in/john-noll-a3b0a55/	global software engineering, global software development, agile software development, open-source software, software engineering	SE M	2492
Paolo Prinetto	paolo.prinetto@cybersecnatlab.it https://www.linkedin.com/in/paolo-prinetto-154ab29/	Digital Systems Design & Test, System Dependability, Design-for-Testability, Built-in Self Test & Built-in Self Repair methodologies, Reconfigurable System Design	SE, T	5641
Academic experts from New Zealand universities (5) (Personal details are available online to the public.)				
Sanjay Mathrani	s.mathrani@massey.ac.nz https://www.linkedin.com/in/sanjaymathrani/	Enterprise Systems, Product Development, Agile Manufacturing, Environmental Performance	SE M	869
Kelly Blincoe	k.blincoe@auckland.ac.nz https://www.linkedin.com/in/kelly-blincoe-8b08a455/	software dependencies, software ecosystems, collaborative software development, software requirements engineering, and software developer diversity and inclusion	SE M	3181
Mee Loong	bobby.yang@aut.ac.nz	Cybersecurity	Sec	NA

(Bobby) Yang	https://www.linkedin.com/in/bobby-yang-a1196179/		T	
Ian Welch	ian.welch@vuw.ac.nz https://www.linkedin.com/in/drianwelch/	Computer security	Sec T	3303
Daniel Alencar da Costa	danielcalencar@otago.ac.nz https://www.linkedin.com/in/daniel-alencar-da-costa-30852419/	software engineering, release engineering, continuous integration	SE M	1155
Academic experts from colleagues or acquaintances (3) **(Personal details are available online to the public.)**				
David Nandigam	dnandigam@yahoo.com https://www.linkedin.com/in/david-nandigam-4772ba37/	Cybersecurity	Sec T	137
Hasan Yasar	hyasar@cmu.edu hyasar@sei.cmu.edu hyasar@andrew.cmu.edu https://www.linkedin.com/in/hasanyasar/	DevOps, Software Security, DevSecOps, Agile, SRE	Sec M	177
Marios Fokaefs	fokaefs@yorku.ca https://www.linkedin.com/in/marios-fokaefs-6639947b/	software engineering, cloud computing, self-adaptive systems, performance engineering	SE T	1413
Industrial experts identified from included grey literature (18) **(Personal details are available online to the public.)**				
Full Name	**LinkedIn**	**Positions/Roles**	**Types**	**Experience**
Kirstie Magowan	https://www.linkedin.com/in/kirstie-magowan-cbrm-01466611/	Business Relationship Manger for NZ Police, IT Service Management Consultant for Verso Solutions, Author for ITIL4	M, G	20+ years
Sam Bocetta	https://www.linkedin.com/in/sambocetta/	Security Analyst, Researcher and Evangelist, Retired Network Engineer for Navy	T, S	20+ years
Anastasios Arampatzis	https://www.linkedin.com/in/anastasiosarampatzis/	Cybersecurity Content Writer for Bora	M, S	20+ years
James Darwin	https://www.linkedin.com/in/james-darwin/	Identity and Access Management (IAM) & Cloud Security Solution Architect, CISSP, CCSP at Okta	T, S	20+ years
Pete Cheslock	https://www.linkedin.com/in/petecheslock/	Head of Growth and Community for AppMap, Principal Cloud Economist for the Duckbill Group	M, G	20+ years
Gilad David Maayan	https://www.linkedin.com/in/giladdavidmaayan/	Strategic SEO/PPC, Content Writer for Technology Markets	M, G	20+ years
Tj Blogumas	https://www.linkedin.com/in/tj-blogumas-mba/	Cloud/DevOps Architect at Broadcom Inc	T, S	10-20 years
Mike Spisak	https://www.linkedin.com/in/mike-spisak/	Distinguished Engineer, Master Inventor, Architect, CTO of the IBM	T, S	10-20 years

		Security Garage		
Bojana Dobran	https://www.linkedin.com/in/bojanadobran/	Enterprise Product Marketing Manager at Global Message Services (GMS)	M, S	10-20 years
Isaac Eldridge	https://www.linkedin.com/in/isaac-eldridge-989a117a/	Technical Writer at Outshift by Cisco	M, G	10-20 years
Kev Zettler	https://www.linkedin.com/in/kev-zettler/	Full Stack Engineer at King	T, G	10-20 years
Ethan Miller	https://www.linkedin.com/in/ethan-miller-649206188/	Developer, IT at American Airlines	T, G	10-20 years
Lucian Constantin	https://www.linkedin.com/in/lconstantin/	Cybersecurity Journalist for CSOOnline.com	M, S	10-20 years
Mark Preston	https://www.linkedin.com/in/mgpreston/	Software Engineer at self-employed	T, G	10-20 years
Christy Maerz	https://www.linkedin.com/in/christy-maerz/	Content Marketing Manager at AppDynamics	M, S	5-10 years
Ilai Bavati	https://www.linkedin.com/in/ilai-bavati-0b1a1418a/	Content Editor at Agile SEO Israel	M, G	5-10 years
Marius Rimkus	https://www.linkedin.com/in/mariusrimkus/	Sales Manager at Cherry Servers	M, S	5-10 years
Ayush Singh	https://www.linkedin.com/in/ayushsingh08/	Technical Program Manager at Amazon	M, S	5-10 years
Industrial experts identified from the editors of SWEBOK4 (3) (Personal details are available online to the public.)				
Steve Tockey	https://www.linkedin.com/in/steve-tockey-2a595/	Principal Consultant at Construx Software	T, G	20+ years
Rich Hilliard	https://www.linkedin.com/in/rhilliard/	Software Systems Architect	T, S	20+ years
Steve Schwarm	https://www.linkedin.com/in/steveschwarm/	Software Architecture Consultant	T, S	20+ years
Peter Leather	https://www.linkedin.com/in/peterleather/	Collaborative leadership & stewardship at SFIA Foundation	M, G	20+ years
Bob Aiello	https://www.linkedin.com/in/bobaiello/	Founder, CTO & Principal Consultant at CM Best Practices	M, G	20+ years
Industrial experts identified from GitHub Universe (30) (Personal details are available online to the public.)				
Mike Hanley	https://www.linkedin.com/in/michael-hanley-b6508913/	Chief Security Officer and SVP of Engineering at GitHub	M, S	20+ years
Robert Daugherty	https://www.linkedin.com/in/robertddaugherty/	Chief Security Officer for Sierra Nevada Corporation (SNC)	M, S	20+ years
Niroshan Rajadurai	https://www.linkedin.com/in/niroshanr/	Senior Director, GTM Strategy for GitHub (AI, DevSecOps & Application Security)	T, G	20+ years
René van	https://www.linkedin.com/in/renev	CTO at Xpirit, Xebia	T, G	20+ years

Osnabrugge	anosnabrugge/			
Rob Bos	https://www.linkedin.com/in/bosrob/	DevOps Consultant and Trainer at Xpirit, Xebia	M, S	20+ years
Caleb Queern	https://www.linkedin.com/in/cqueern/	Managing Director at KPMG Cybersecurity Services	M, S	20+ years
James Williams	https://www.linkedin.com/in/james-williams-0352b5/	IT Advisory Director at KPMG	M, G	20+ years
Andrew Hoog	https://www.linkedin.com/in/andrewhoog/	Founder & Board Member of NowSecure	M, G	20+ years
Charlie Rice	https://www.linkedin.com/in/charles-m-rice/	Principal Field Security Specialist at GitHub	T, S	20+ years
Jérôme Djebari	https://www.linkedin.com/in/jedjebari/	Tech Leader, Senior on Cloud Technologies	T, S	20+ years
Asha Chakrabarty	https://www.linkedin.com/in/ashachakrabarty/	Vice President of Product Management for Security & Compute products at GitHub	M, S	10-20 years
Chad Bentz	https://www.linkedin.com/in/chadbentz/	Principal Field Security Specialist at GitHub	T, S	10-20 years
Dan Shanahan	https://www.linkedin.com/in/danshanahan/	Principal Field Security Specialist at GitHub	T, S	10-20 years
Abhishek Dutta	https://www.linkedin.com/in/abhishek-dutta07/	Senior Solution Architect at GitHub	T, S	10-20 years
Mohammad Ismail	https://www.linkedin.com/in/mouismail/	Senior Solution Architect at GitHub	T, S	10-20 years
Igwe Kalu	https://www.linkedin.com/in/igwejk/	DevSecOps Architect at GitHub	T, S	10-20 years
Jeanyhwh Desulme	https://www.linkedin.com/in/jeanyhwh-desulme-18972a13/	Solutions Engineer at Liberty Mutual Insurance	T, G	10-20 years
Nick Liffen	https://www.linkedin.com/in/nickliffen/	Director, Advanced Security at GitHub	T, S	10-20 years
Javier Cardoso	https://www.linkedin.com/in/javiercardoso/	Tech Senior Manager - Cloud & Platform at Mercadolibre	M, S	10-20 years
Erik Arcos	https://www.linkedin.com/in/erik-arcos-b6a0a717/	Tech Project Leader at Mercadolibre	M, G	10-20 years
Justin Hutchings	https://www.linkedin.com/in/hutchingsjustin/	Senior Director of Product Management at GitHub	M, G	10-20 years
Zain Malik	https://www.linkedin.com/in/zainmalik/	Senior Product Manager at GitHub	M, S	10-20 years
Courtney Claessens	https://www.linkedin.com/in/courtneyclaessens/	Senior Product Manager at GitHub	M, S	10-20 years
Erin Havens	https://www.linkedin.com/in/havenserin/	Senior Product Manager at GitHub	M, S	10-20 years
Mathew Payne	https://www.linkedin.com/in/mathewbpayne/	Principal Field Security Specialist at GitHub	T, S	5-10 years

Shadi Samadi	https://www.linkedin.com/in/shadi-s-8261711b3/	Staff Delivery Engineer - Advanced Security at GitHub	T, S	5-10 years
Greg Mohler	https://www.linkedin.com/in/gmohler/	Security Solution Architect at GitHub	T, S	5-10 years
Matthew Chieco	https://www.linkedin.com/in/matthew-chieco/	Principal Software Engineer at Liberty Mutual Insurance	T, G	5-10 years
Charlton Trezevant	https://www.linkedin.com/in/charlton-trezevant/	Application Security Consultant at Xpirit, Xebia	T, S	5-10 years
Joseph Katsioloudes	https://www.linkedin.com/in/jkcso/	Security Specialist at GitHub	T, S	5-10 years
Industrial experts from NZ based business (5) (Personal details are available online to the public.)				
Baren Nel	barend.nel@datacom.com https://www.linkedin.com/in/barendnel/	General Manager at datacom	M, G	10-20 years
Cheryll Singh	https://www.linkedin.com/in/cheryllsingh/	Delivery Lead at Westpac bank	M, S	10-20 years
Kinza Sarwar	https://www.linkedin.com/in/kinzasarwar/	Senior Security Consultant at Axenic Ltd	M, S	5-10 years
Philip Coster	https://www.linkedin.com/in/philcoster/	Independent Adviser at Grey Matter Advisory	M, G	20+ years
Olivia Tang	https://www.linkedin.com/in/olivia-tang-b41b5b218/	Senior Agile Coach at Talkdesk	M, S	10-20 years
Industrial experts from colleagues or acquaintances (4) (Personal details are available online to the public.)				
V. S. Mani	vs.mani@hotmail.com https://www.linkedin.com/in/mani-v-s-9982956/	Head Marketing and Communications at Siemens Healthineers Development Center	M, G	20+ years
Tim Tegeler	tim.tegeler@tu-dortmund.de https://www.linkedin.com/in/tim-tegeler/	Principal Software Engineer at adesso SE	T, S	10-20 years
Song Sun	me@sunsong.org https://www.linkedin.com/in/song-sun-30835a71/	Senior DevOps Engineer at IBM	T, S	10-20 years
Aziel Shaw	nzazielio@gmail.com https://www.linkedin.com/in/aziel-shaw-408713113/	DevOps Engineer at Foundry	T, S	5-10 years

Appendix B.7. Delphi Plan

Phases	Goals	Steps	Estimated time
Preparation **(12 weeks)**	➢ Define research goals ➢ Define Delphi format ➢ Define Delphi statement ➢ Define question format	<u>Initial conceptualisation</u> • Overall goal and research setting	2 weeks
		<u>Creative workshop</u> • Scope of the Delphi • Theory and framework – CPTM Model V1.0 • Multi-rounds/real time – multi-rounds	2 weeks
		<u>Desk research</u> • Topic, experts, and public opinions	2 weeks
		<u>Initial expert assessment</u> • Experts from closer network • Experts' reviews and feedback • Discussion and amendment	2 weeks
		<u>Formulation sessions</u> • Statement formulation and iterative refinement • Question format: close-ended questions (AHP – pairwise comparisons) + open-ended questions (additional comments) • Related information (brief introduction, clarification of intention, survey process, research ethics, explanation of terms, etc.)	4 weeks
Conduct **(50 weeks)**	➢ Select software or tools ➢ Set up software or tools ➢ Identify expert panel ➢ Collect experts' opinions and data	<u>Software selection</u> • Technical Delphi requirement • Online survey tool – Qualtrics • AHP tool – SuperDecisions	2 weeks
		<u>Survey programming</u> • Survey introduction • Dimensions of each statement • Page branching • Survey pre-test and pilot	4 weeks
		<u>Expert selection and grouping</u> • Demographic characteristics • Area of expertise (selection criteria) • Identification strategy (five steps) • Anticipated response rate	8 weeks
		<u>Expert recruitment and invitation</u> • Associated documents, i.e., introductory letter, information sheet and explanation of terms • Emails for the academics and LinkedIn for the industrials • Frist contact with associated documents • Reminder and post-recruitment	12 weeks

		Survey conduct • Research ethical application approval • Termination criteria (time-related, participant-related, and consensus-related). • Dissent is a valid outcome • Number of rounds of survey (3 rounds)	22 weeks
		Expert follow-up • Appreciation for participation • Invitation for follow-up discussion • Reminder messages	2 weeks
Analysis **(15 weeks)**	➢ Analyse AHP results ➢ Analyse answers to open-ended questions ➢ Analyse dissent	AHP results analysis • Use of surveying tool Qualtrics for initial frequency analysis • Use of AHP tool SuperDecisions for AHP pairwise comparison analysis • Consistency analysis	8 weeks
		Answers to open-ended questions (TA) • Additional comments from participants • Differences of DevSecOps between local and global settings	2 weeks
		Dissent analysis • Dissent analysis on AHP pairwise comparison results, based on participants' individuals, groups, and roles. • Dissent analysis on global DevSecOps opinions	5 weeks

Appendix B.8. Sample of Delphi Survey – Round One

Instruction. **Thanks for participating in the Delphi survey (Round One), in terms of the challenges when adopting DevSecOps.**

1. Please do the pairwise comparison of all DevSecOps challenges and their categories.

2. AHP Scale: 1 - Equally important, 3 - Moderately more important, 5 - Strongly more important, 7 - Very strongly more important, 9 - Extremely more important.

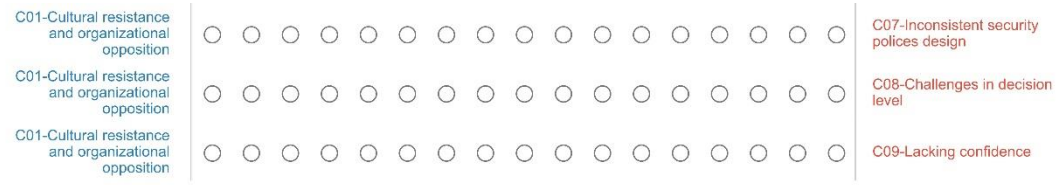
3. At the end of the questionnaire, there are a few choice and open questions relating to DevSecOps in GSE (Global Software Engineering).

Q1. With respect to the DevSecOps challenges, for each pairwise comparison, which category is more important? And how much more on a scale 1 to 9?

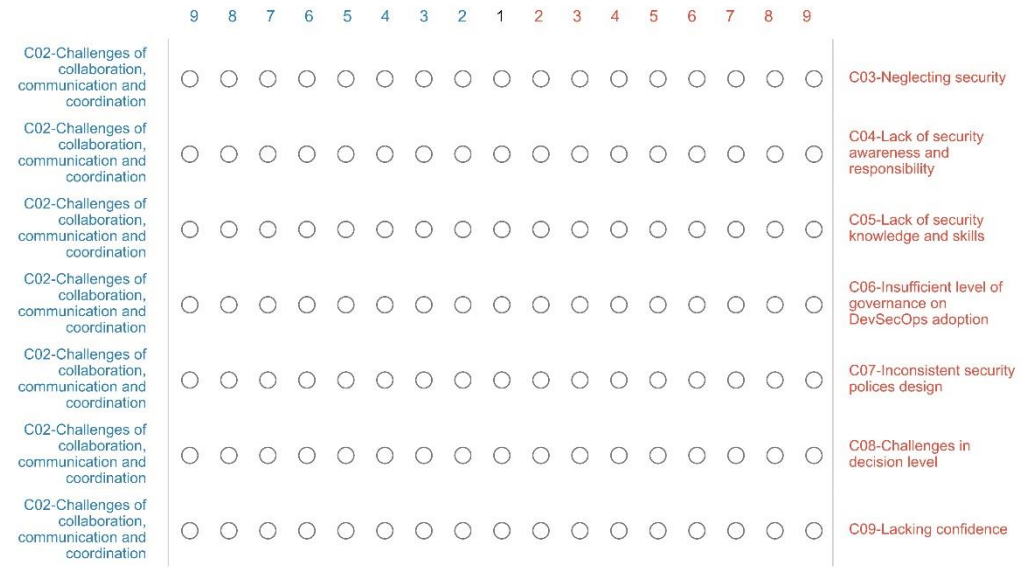
	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Organization, People, and Culture
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Process Capabilities
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology
Organization, People, and Culture	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Process Capabilities
Organization, People, and Culture	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology
Process Capabilities	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology

Q2. With respect to Organization, People, and Culture, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?

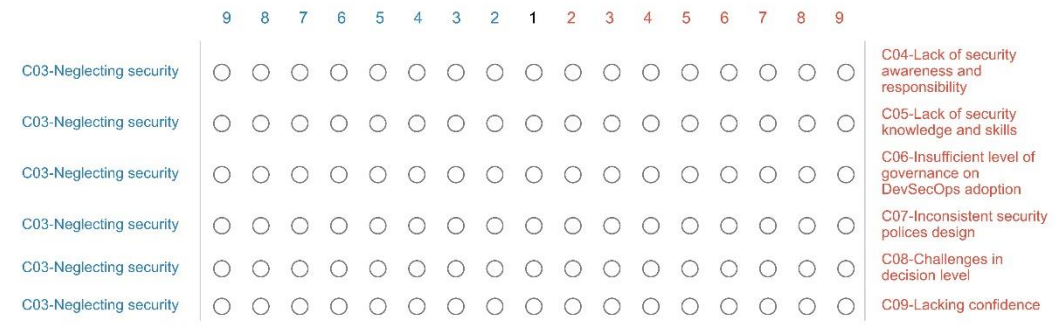
	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
C01-Cultural resistance and organizational opposition	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C02-Challenges of collaboration, communication and coordination
C01-Cultural resistance and organizational opposition	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C03-Neglecting security
C01-Cultural resistance and organizational opposition	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C04-Lack of security awareness and responsibility
C01-Cultural resistance and organizational opposition	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C05-Lack of security knowledge and skills
C01-Cultural resistance and organizational opposition	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C06-Insufficient level of governance on DevSecOps adoption



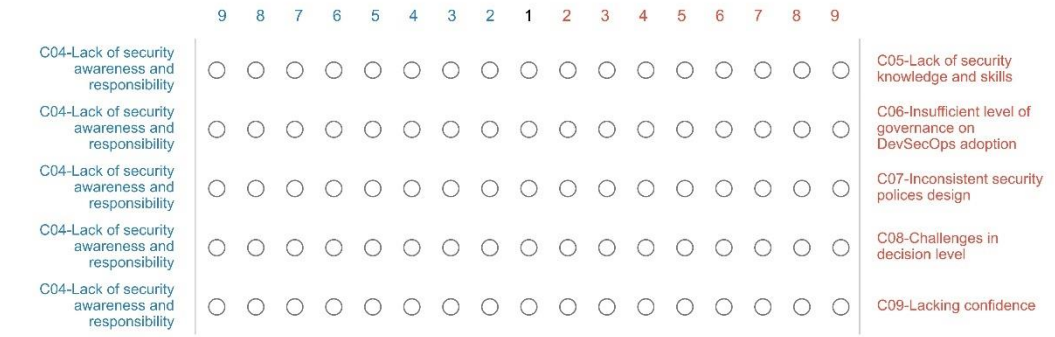
Q3. With respect to Organization, People, and Culture, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



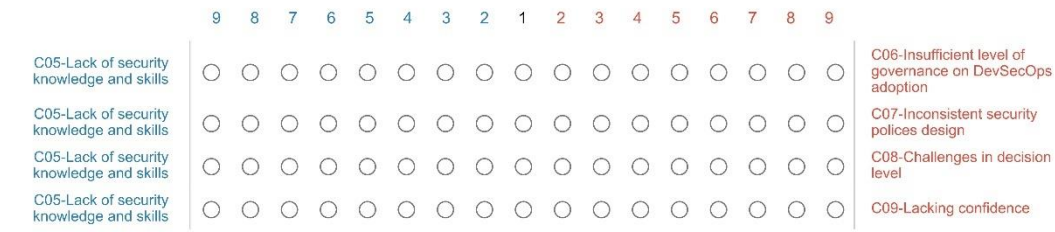
Q4. With respect to Organization, People, and Culture, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



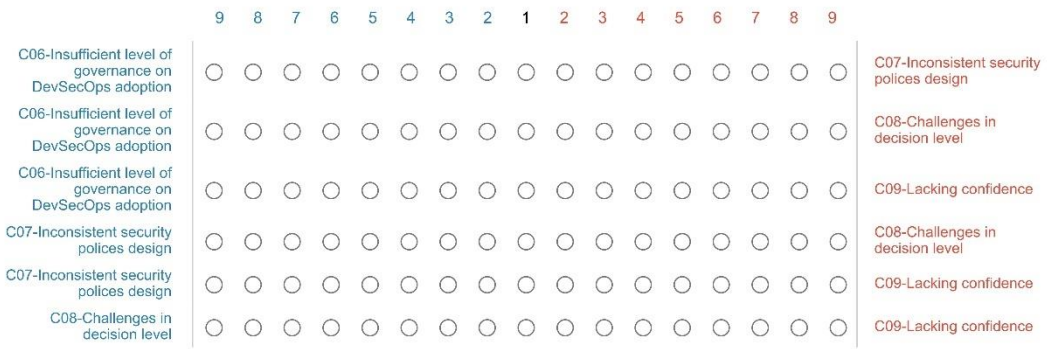
Q5. With respect to Organization, People, and Culture, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



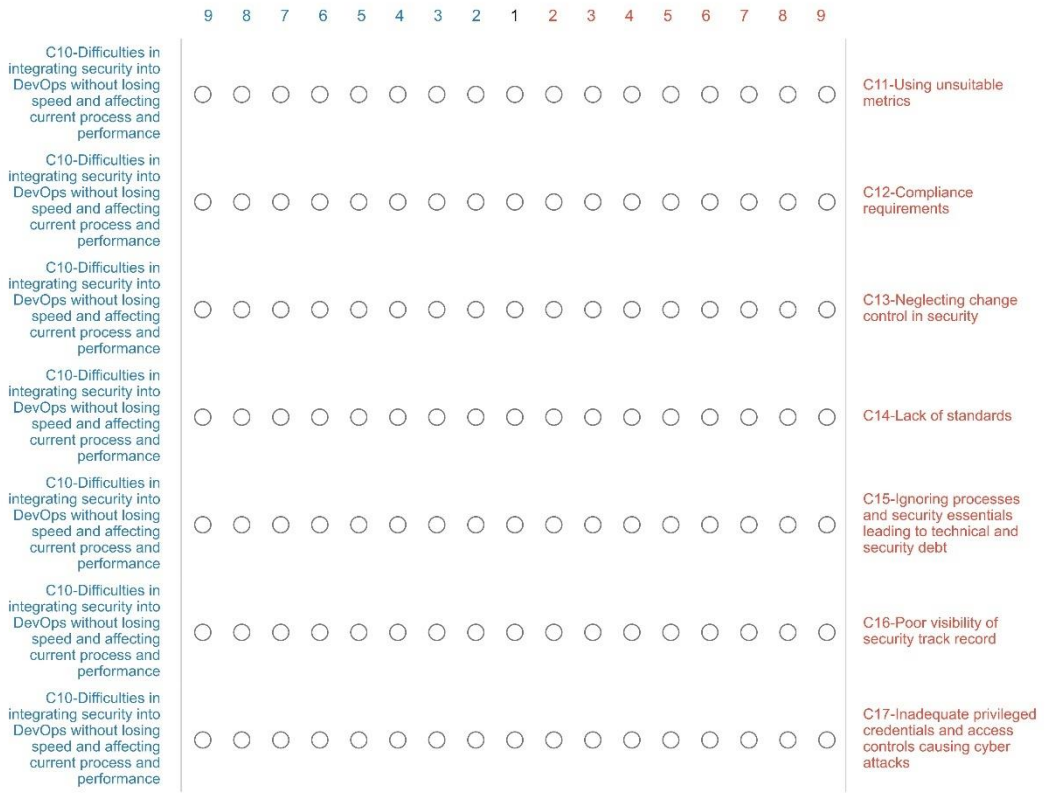
Q6. With respect to Organization, People, and Culture, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



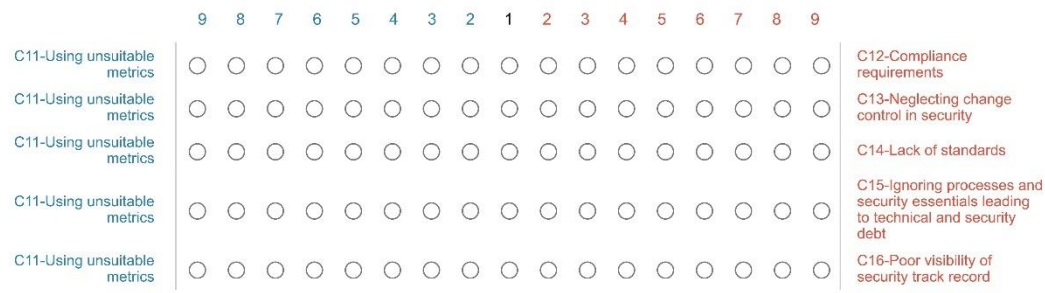
Q7. With respect to Organization, People, and Culture, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



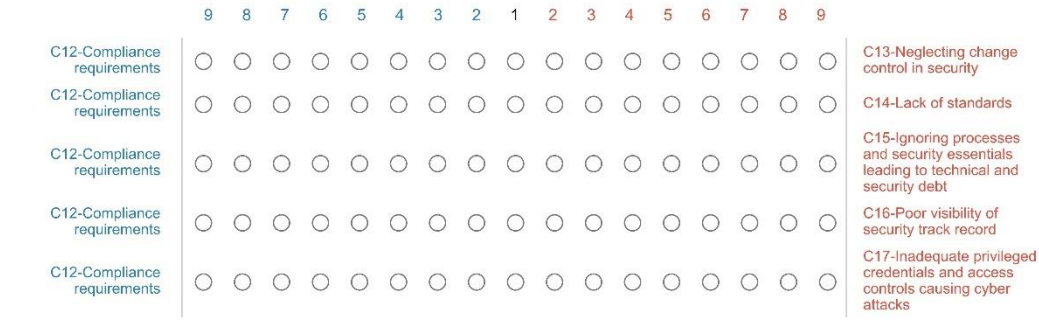
Q8. With respect to Process Capabilities, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



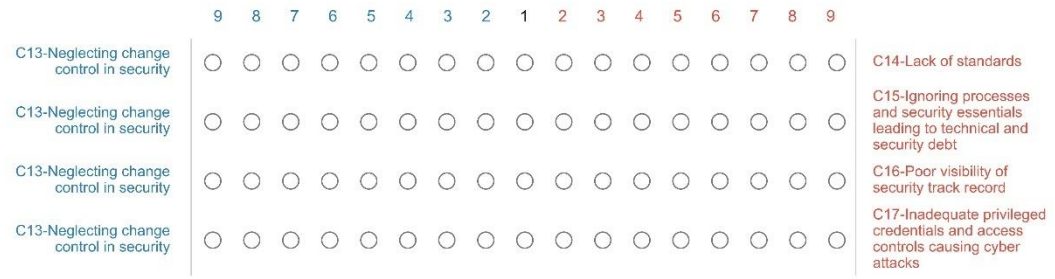
Q9. With respect to Process Capabilities, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



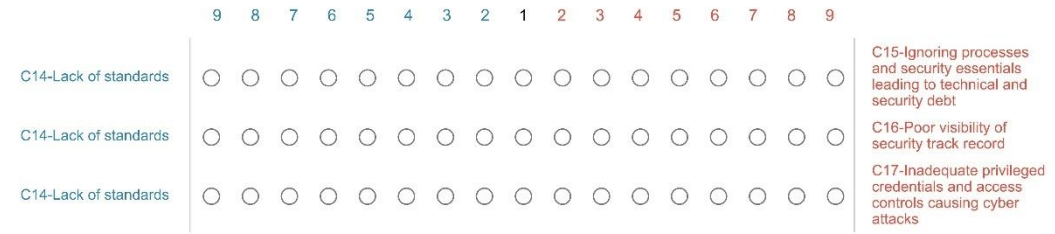
Q10. With respect to Process Capabilities, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?

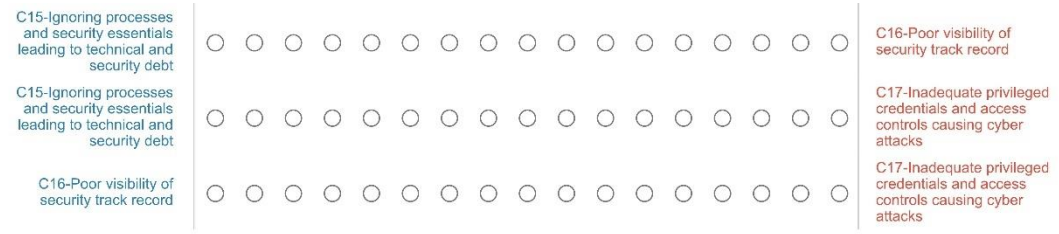


Q11. With respect to Process Capabilities, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?

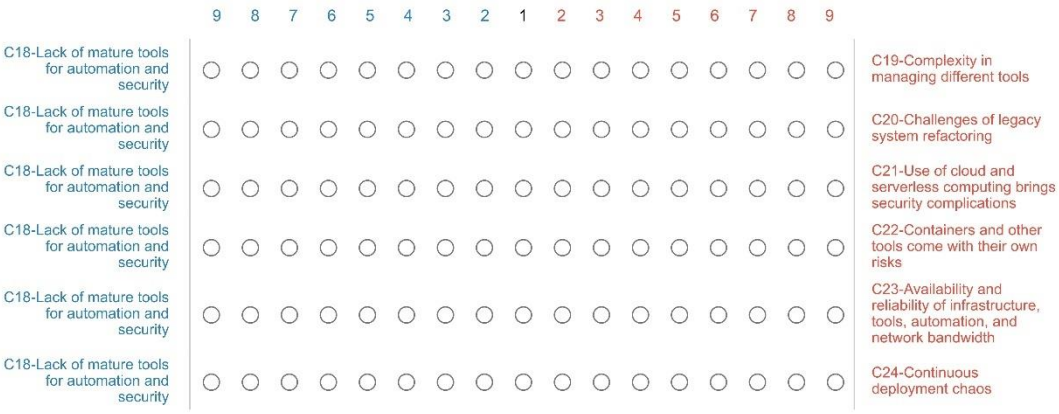


Q12. With respect to Process Capabilities, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?

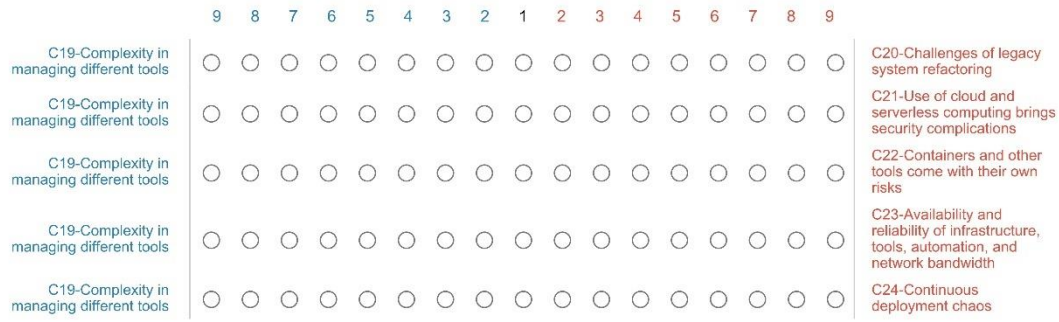




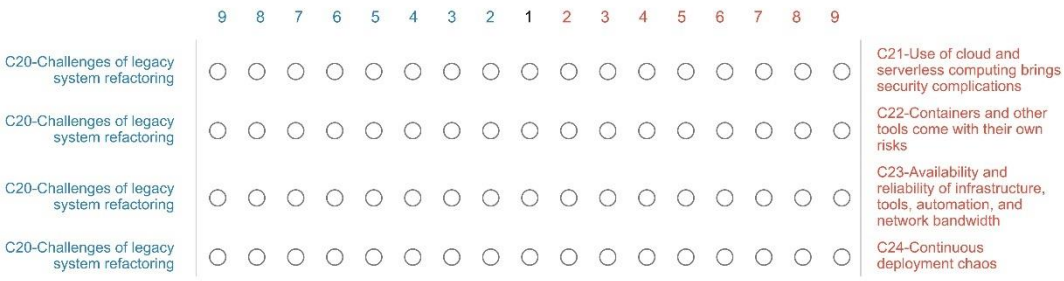
Q13. With respect to Technology, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



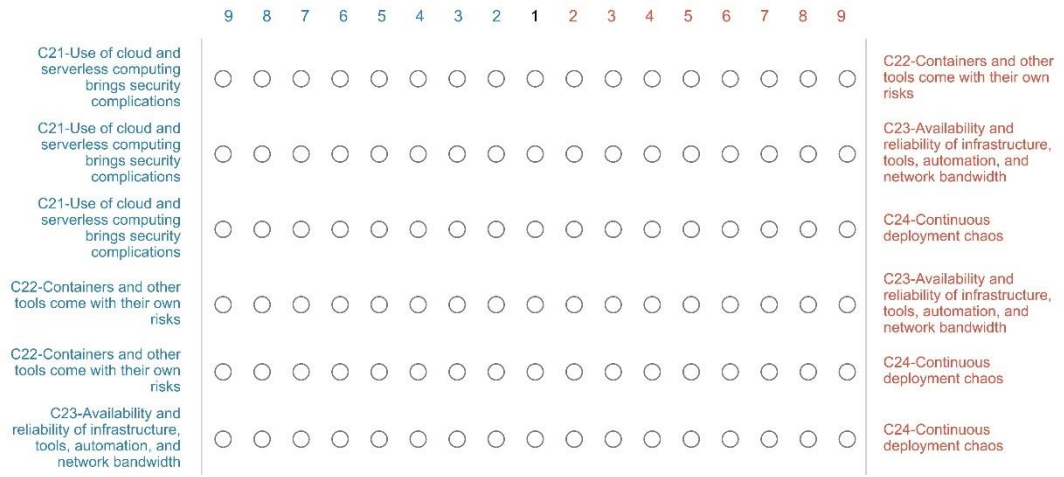
Q14. With respect to Technology, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



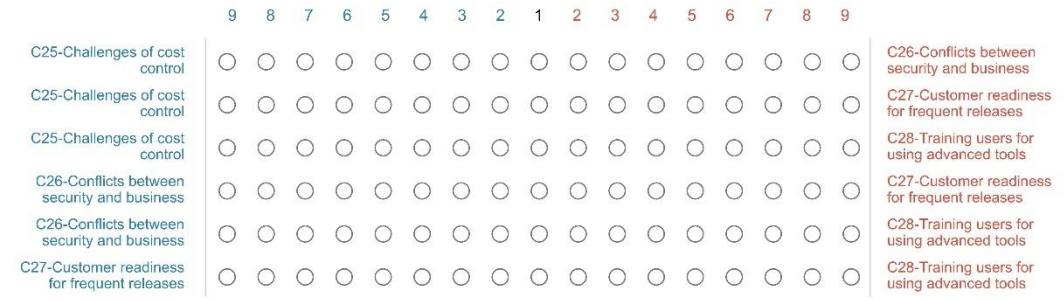
Q15. With respect to Technology, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



Q16. With respect to Technology, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



Q17. With respect to Business, for each pairwise comparison, which challenge is more important? And how much more on a scale 1 to 9?



Q18. What additional DevSecOps challenges do you want to add to this list?

Q19. Regarding to DevSecOps in GES (Global Software Engineering) context, how do you think DevSecOps differs in local and global settings.

- ○ Extremely different
- ○ Slightly different
- ○ Not different
- ○ Uncertain

Q20. If answered "Extremely/Slightly different" in previous question, could you please list some additional challenges when adopting DevSecOps in a global setting.

Location Data

Location: (-36.8504, 174.7675)

Source: GeoIP Estimation

Appendix B.9. Sample of Delphi Survey – Round Two

Instruction. **Thanks for participating in the Delphi survey (Round Two), in terms of the practices relating to DevSecOps.**

1. Please firstly make comparisons between revised DevSecOps Challenges. Based on Round 1 Results, a few new challenges are added and revised.

2. Please make comparisons between 60 identified DevSecOps Practices and their categories.

3. Due to repeating Round 1 and the large volume of identified practices, this round minimizes the number of questions (comparisons), rather than full pairwise comparisons. The minimum number of questions is comprised of only the comparison judgments in the diagonal above the main diagonal (e.g., P1/P2, P2/P3, P3/P4, P4/P5, etc.) of the comparison matrix. Once these comparisons have been done, the rest of comparison judgments in the upper and lower parts of matrix can be calculated.

4. AHP Scale: 1 - Equally important; 3 - Moderately more important; 5 - Strongly more important; 7 - Very strongly more importan; 9 - Extremely more important; 2, 4, 6, 8 - intermediate values.

Q1. With respect to the DevSecOps challenges, which category is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Organization, People, and Culture
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Process Capabilities
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology
Organization, People, and Culture	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Process Capabilities
Organization, People, and Culture	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology
Process Capabilities	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology

Q2. With respect to Organization, People, and Culture, which challenge is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
C01-Cultural resistance and organizational opposition	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C02-Challenges of collaboration, communication and coordination
C02-Challenges of collaboration, communication and coordination	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C03-Neglecting security

C03-Neglecting security	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C04-Lack of security awareness and responsibility
C04-Lack of security awareness and responsibility	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C05-Lack of security knowledge and skills
C05-Lack of security knowledge and skills	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C06-Recruiting challenges
C06-Recruiting challenges	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C07-Inconsistent security polices design
C07-Inconsistent security polices design	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C08-Challenges of governance and leadership
C08-Challenges of governance and leadership	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C09-Lacking confidence
C09-Lacking confidence	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	NC01-Product teams do not follow DevSecOps practices

Q3. With respect to Process Capabilities, which challenge is more important? And how much more on a scale 1 to 9?

	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9	
C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C11-Using unsuitable metrics
C11-Using unsuitable metrics	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C12-Compliance requirements
C12-Compliance requirements	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C13-Neglecting change control in security
C13-Neglecting change control in security	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C14-Lack of standards
C14-Lack of standards	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C15-Ignoring processes and security essentials leading to technical and security debt
C15-Ignoring processes and security essentials leading to technical and security debt	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C16-Poor visibility of security track record
C16-Poor visibility of security track record	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C17-Inadequate privileged credentials and access controls causing cyber attacks
C17-Inadequate privileged credentials and access controls causing cyber attacks	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	NC02-Improper or inadequate risk assessment and management
NC02-Improper or inadequate risk assessment and management	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	NC03-Lack of reference model for DevSecOps process

Q4. With respect to Technology, which challenge is more important? And how much more on a scale 1 to 9?

	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9	
C18-Lack of mature tools for automation and security	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C19-Complexity in managing different tools
C19-Complexity in managing different tools	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C20-Challenges of legacy system refactoring

C20-Challenges of legacy system refactoring	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C21-Use of cloud and serverless computing brings security complications
C21-Use of cloud and serverless computing brings security complications	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C22-Containers and other tools come with their own risks
C22-Containers and other tools come with their own risks	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth
C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C24-Continuous deployment chaos
C24-Continuous deployment chaos	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	NC04-Domain of application does matter like cyber physical systems
NC04-Domain of application does matter like cyber physical systems	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	NC05-Code security / code developers working on their dependences / business logic, excluding external tools

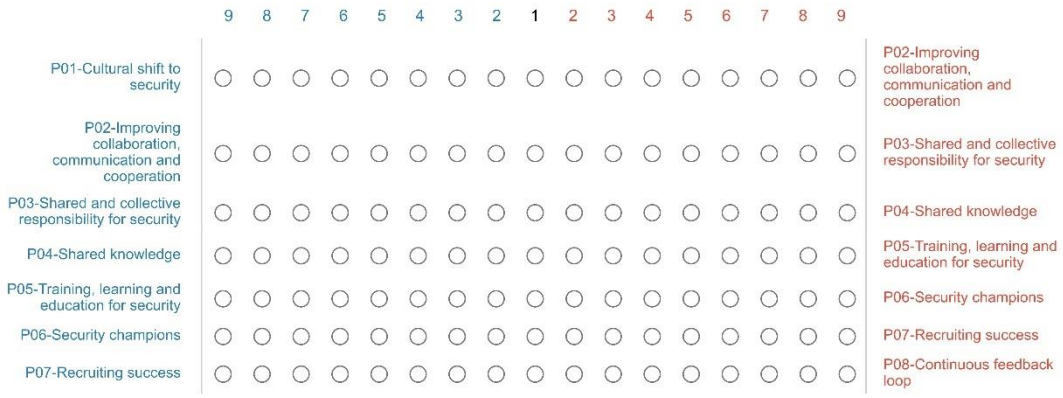
Q5. With respect to Business, which challenge is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
C25-Challenges of cost control	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C26-Conflicts between security and business
C26-Conflicts between security and business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C27-Customer readiness for frequent releases
C27-Customer readiness for frequent releases	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	C28-Training users for using advanced tools
C28-Training users for using advanced tools	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	NC06-The ever-stricter security requirements for businesses that are increasingly challenging are underrepresented
NC06-The ever-stricter security requirements for businesses that are increasingly challenging are underrepresented	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	NC07-Ballance between risk, cost and impact to business

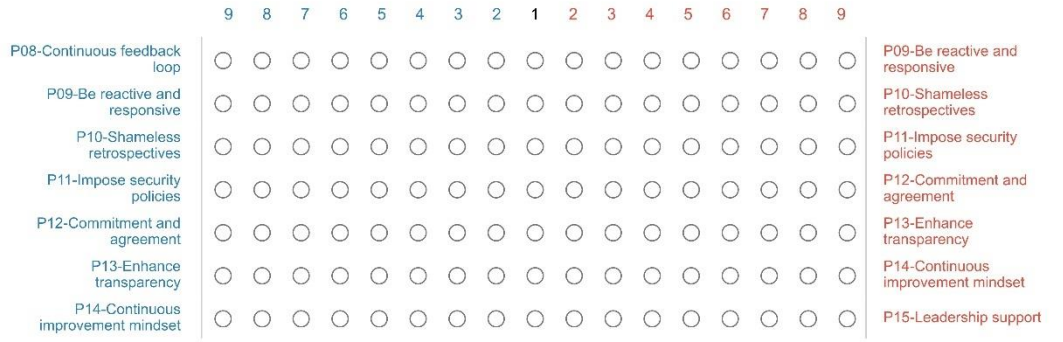
Q6. With respect to the DevSecOps practices, which category is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Organization, People, and Culture
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Process Capabilities
Business	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology
Organization, People, and Culture	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Process Capabilities
Organization, People, and Culture	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology
Process Capabilities	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Technology

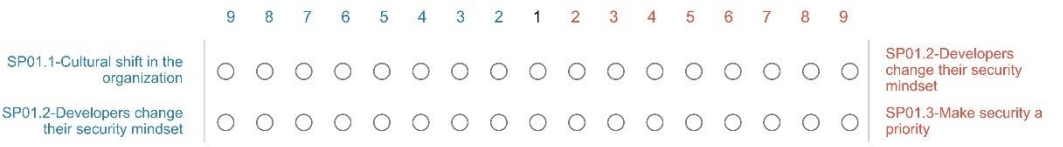
Q7. With respect to Organization, People, and Culture, which practice is more important? And how much more on a scale 1 to 9?



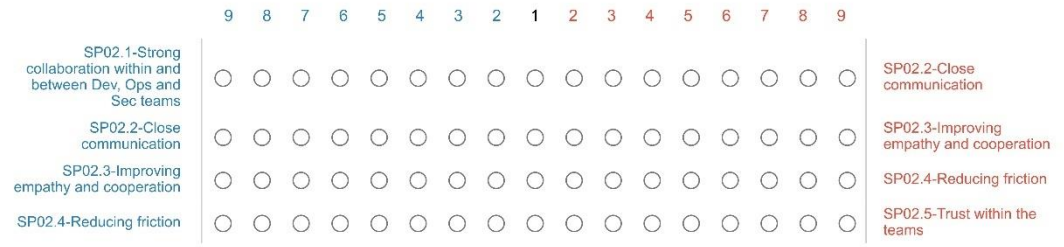
Q8. With respect to Organization, People, and Culture, which practice is more important? And how much more on a scale 1 to 9?



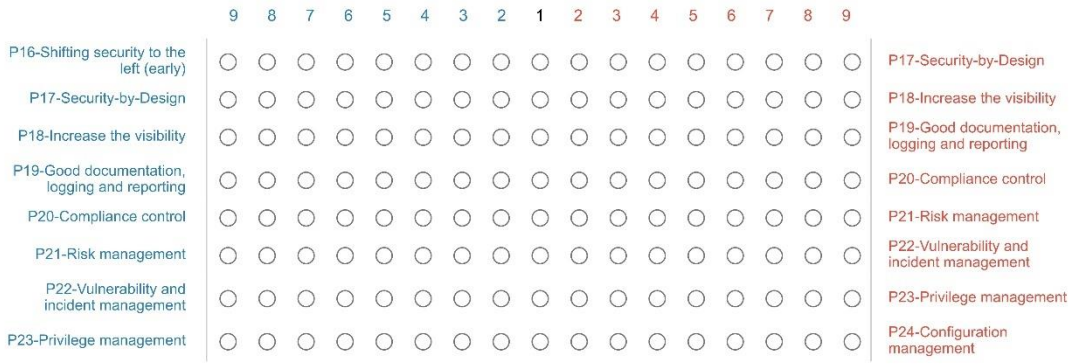
Q9. With respect to P01-Cultural shift to security, which sub-practice is more important? And how much more on a scale 1 to 9?



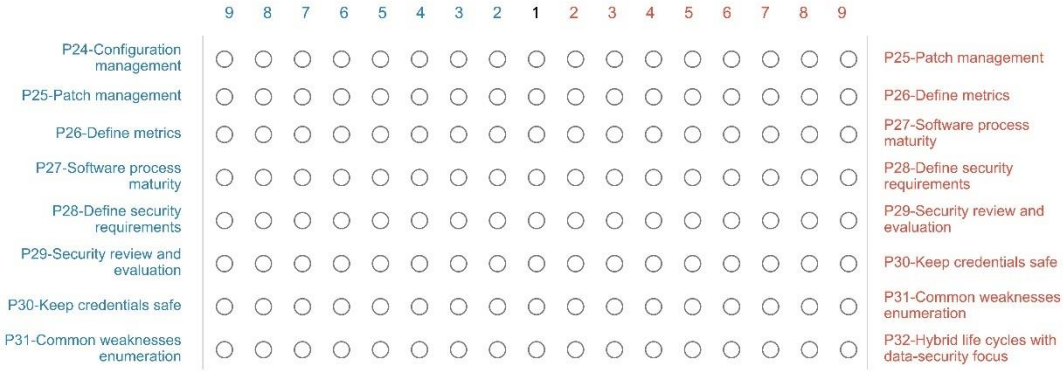
Q10. With respect to P02-Improving collaboration, communication and cooperation, which sub-practice is more important? And how much more on a scale 1 to 9?



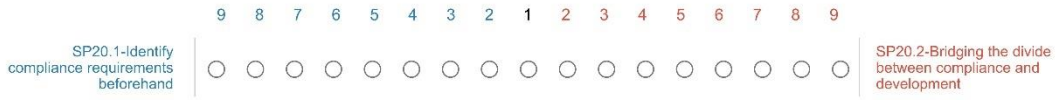
Q11. With respect to Process Capabilities, which practice is more important? And how much more on a scale 1 to 9?



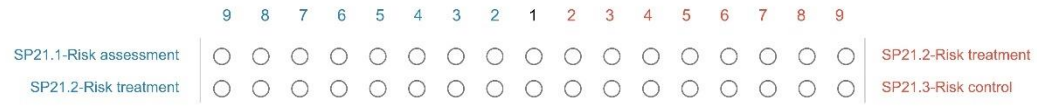
Q12. With respect to Process Capabilities, which practice is more important? And how much more on a scale 1 to 9?



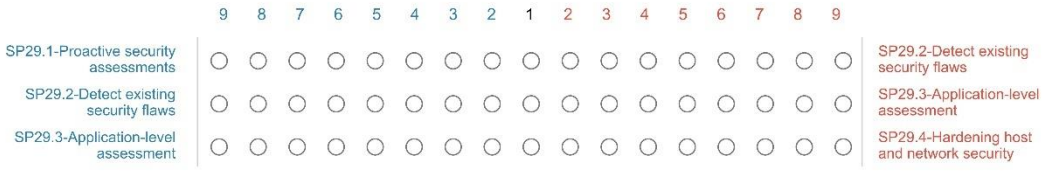
Q13. With respect to P20-Compliance control, which sub-practice is more important? And how much more on a scale 1 to 9?



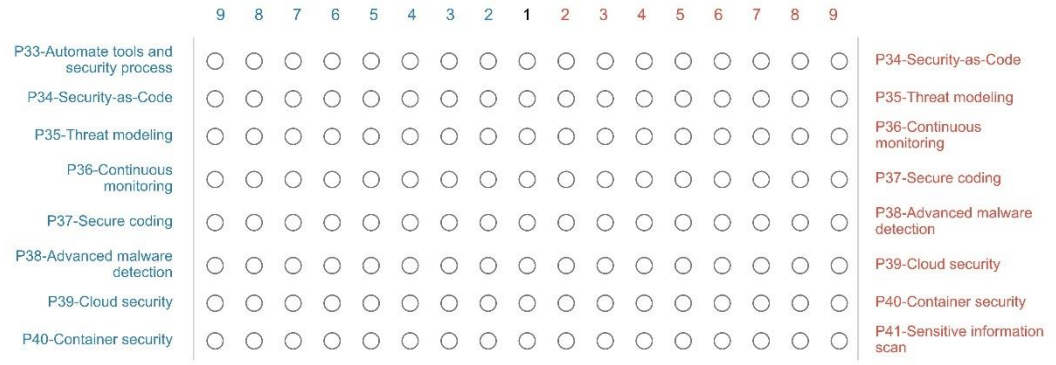
Q14. With respect to P21-Risk management, which sub-practice is more important? And how much more on a scale 1 to 9?



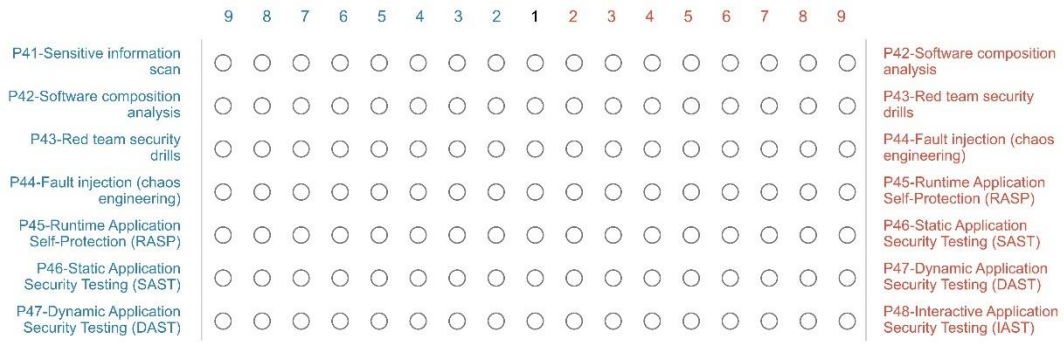
Q15. With respect to P29-Security review and evaluation, which sub-practice is more important? And how much more on a scale 1 to 9?



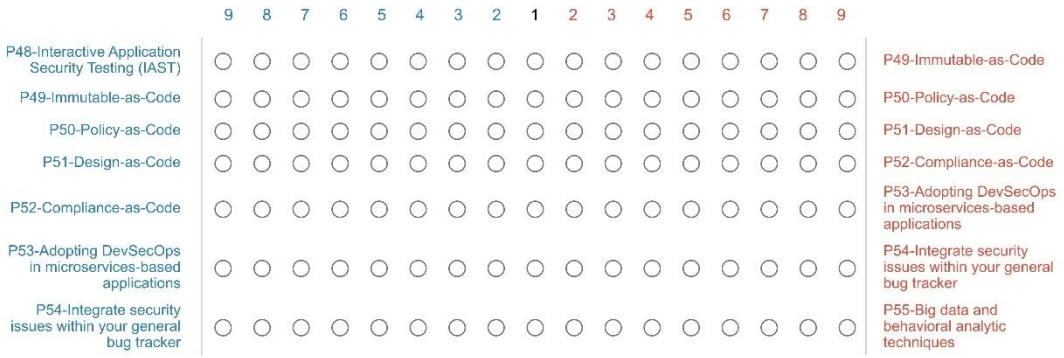
Q16. With respect to Technology, which practice is more important? And how much more on a scale 1 to 9?



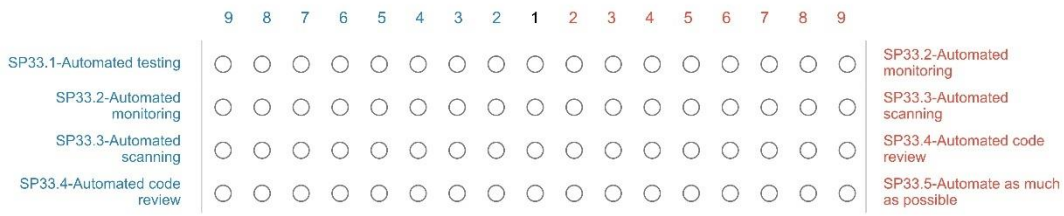
Q17. With respect to Technology, which practice is more important? And how much more on a scale 1 to 9?



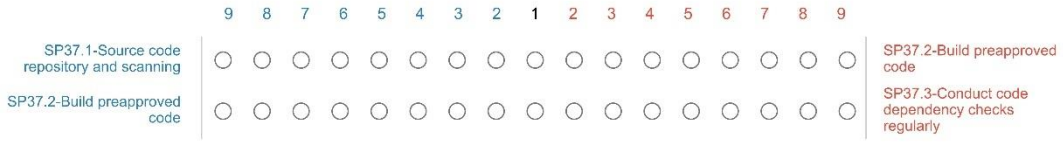
Q18. With respect to Technology, which practice is more important? And how much more on a scale 1 to 9?



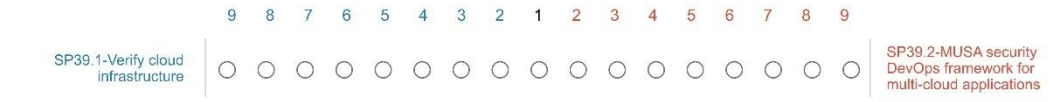
Q19. With respect to P33-Automate tools and security process, which sub-practice is more important? And how much more on a scale 1 to 9?



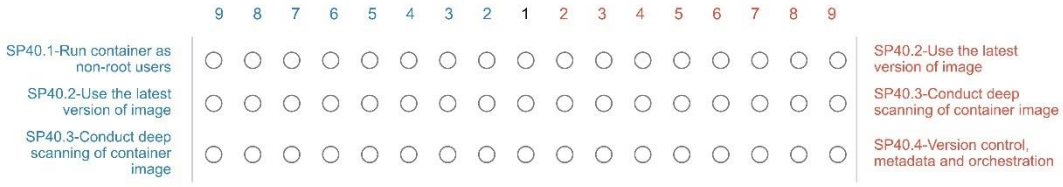
Q20. With respect to P37-Secure coding, which sub-practice is more important? And how much more on a scale 1 to 9?



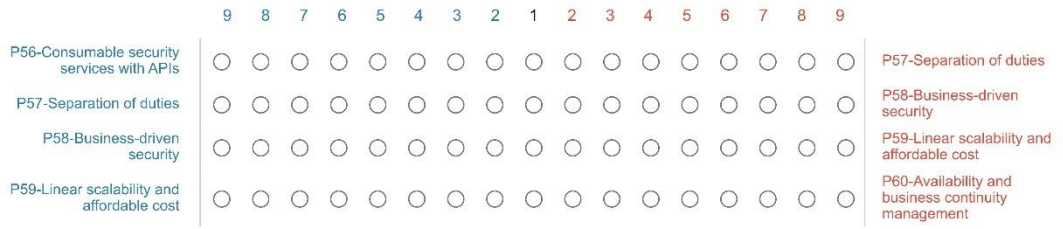
Q21. With respect to P39-Cloud security, which sub-practice is more important? And how much more on a scale 1 to 9?



Q22. With respect to P40-Container security, which sub-practice is more important? And how much more on a scale 1 to 9?



Q23. With respect to Business, which practice is more important? And how much more on a scale 1 to 9?



Q24. What additional DevSecOps practices do you want to add to this list?

Q25. Regarding to DevSecOps practices in GES (Global Software Engineering) context, how do you think DevSecOps differs in local and global settings.

- ◯ Extremely different
- ◯ Slightly different
- ◯ Not different
- ◯ Uncertain

Q26. What additional Global DevSecOps practices do you want to add to this list?

Location Data

Location: (-36.8504, 174.7675)

Source: GeoIP Estimation

Appendix B.10. Sample of Delphi Survey – Round Three

Instruction. **Thanks for participating in the Delphi survey (Round Three), in terms of the metrics and revised practices relating to DevSecOps.**

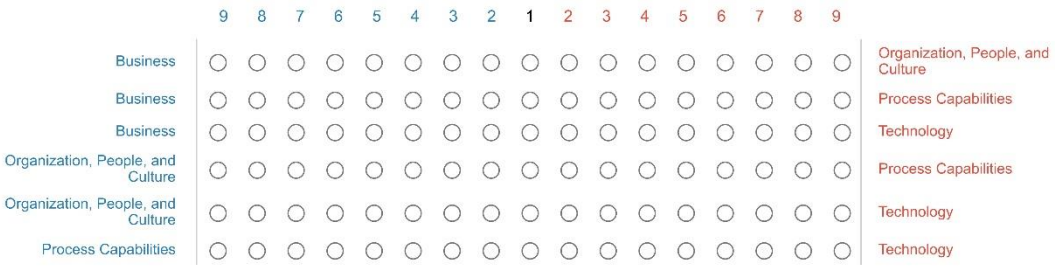
1. Please firstly make pairwise comparisons between the revised DevSecOps Practices. To address the 'recency bias' in Round 2, a few sub-categories are added.

2. Please make pairwise comparisons between the identified DevSecOps Metrics.

3. Same as Round 2, Round 3 minimizes the number of questions (comparisons), instead of the full pairwise comparisons.

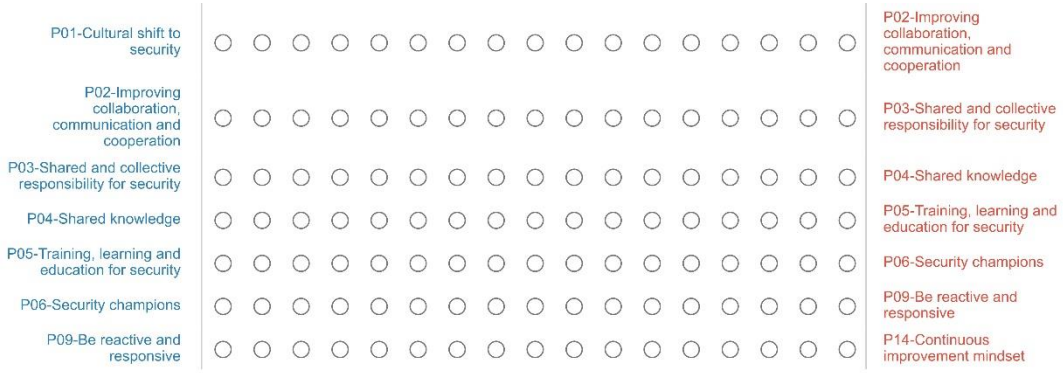
4. AHP Scale: 1 - Equally important; 3 - Moderately more important; 5 - Strongly more important; 7 - Very strongly more importan; 9 - Extremely more important; 2, 4, 6, 8 - intermediate values.

Q1. With respect to the DevSecOps Practices, which category is more important? And how much more on a scale 1 to 9?

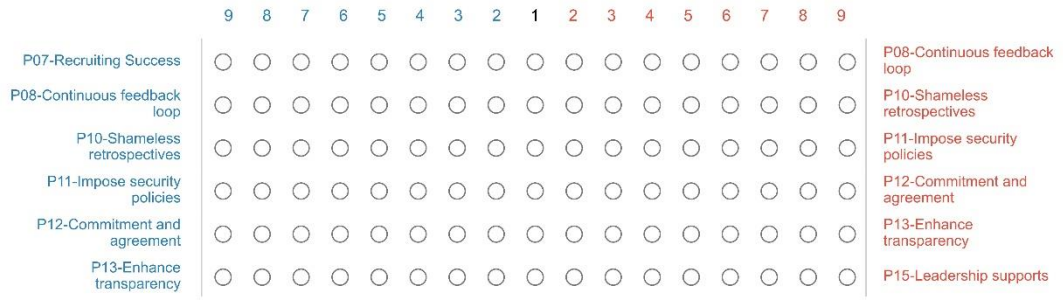


Q2. With respect to Organization, People, and Culture, which sub-category is more important? And how much more on a scale 1 to 9?

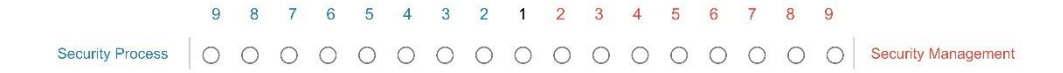
Q3. With respect to People Training, which practice is more important? And how much more on a scale 1 to 9?



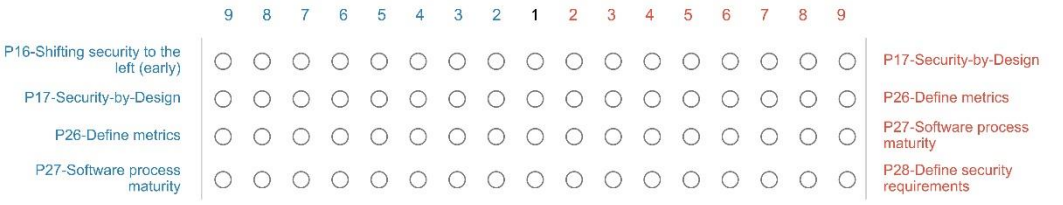
Q4. With respect to Organizational Governance, which practice is more important? And how much more on a scale 1 to 9?

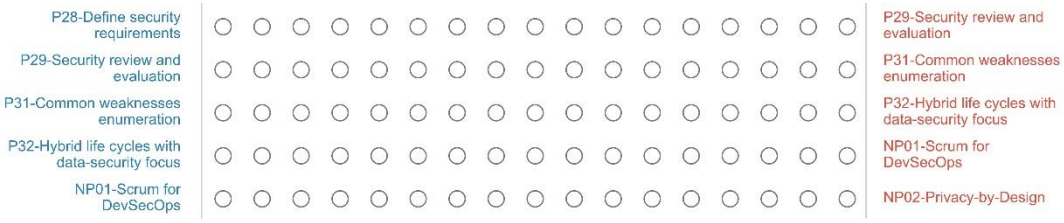


Q5. With respect to Process Capabilities, which sub-category is more important? And how much more on a scale 1 to 9?

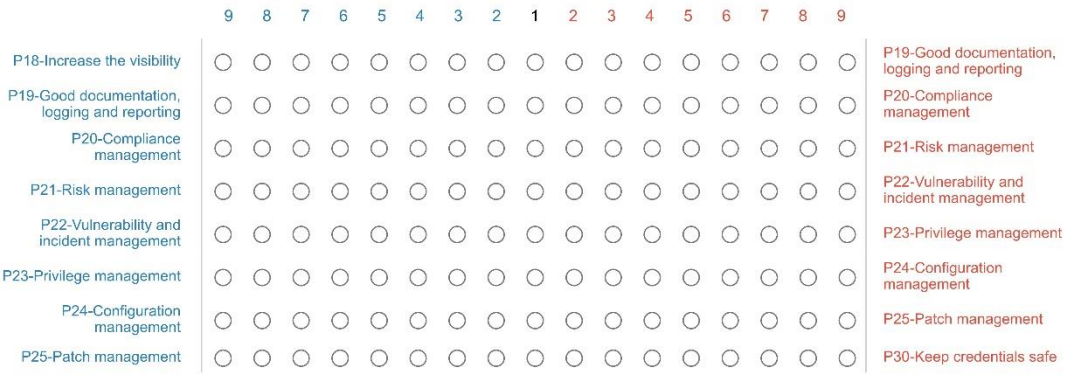


Q6. With respect to Security Process, which practice is more important? And how much more on a scale 1 to 9?

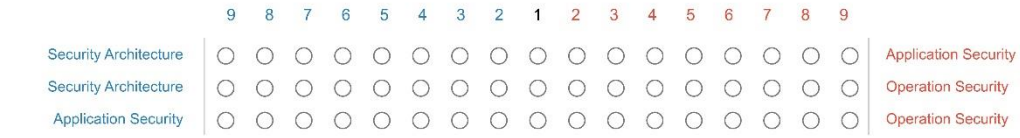




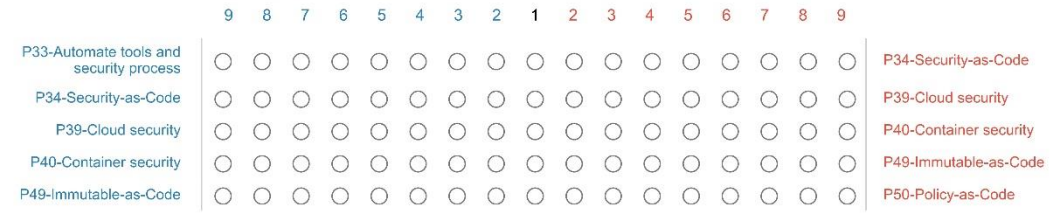
Q7. With respect to Security Management, which practice is more important? And how much more on a scale 1 to 9?



Q8. With respect to Technology, which sub-category is more important? And how much more on a scale 1 to 9?



Q9. With respect to Security Architecture, which practice is more important? And how much more on a scale 1 to 9?



P50-Policy-as-Code	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	P51-Design-as-Code
P51-Design-as-Code	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	P52-Compliance-as-Code
P52-Compliance-as-Code	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	P53-Microservices security
P53-Microservices security	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	P54-Integrate security issues within your general bug tracker

Q10. With respect to Application Secuirty, which practice is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
P37-Secure coding (pre-approved code and source code repository scanning)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P42-Software Composition Analysis (SCA)
P42-Software Composition Analysis (SCA)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P44-Fault injection (chaos engineering)
P44-Fault injection (chaos engineering)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P45-Runtime Application Self-Protection (RASP)
P45-Runtime Application Self-Protection (RASP)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P46-Static Application Security Testing (SAST)
P46-Static Application Security Testing (SAST)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P47-Dynamic Application Security Testing (DAST)
P47-Dynamic Application Security Testing (DAST)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P48-Interactive Application Security Testing (IAST)
P48-Interactive Application Security Testing (IAST)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	NP03-Software Bill of Materials (SBOM)

Q11. With respect to Operation Security, which practice is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
P35-Threat modeling	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P36-Continuous monitoring
P36-Continuous monitoring	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P38-Advanced malware detection
P38-Advanced malware detection	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P41-Sensitive information scan
P41-Sensitive information scan	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P43-Red team security drills
P43-Red team security drills	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P55-Big data and behavioral analytic techniques

Q12. With respect to Business, which practice is more important? And how much more on a scale 1 to 9?

	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	
P56-Consumable security services with APIs	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P57-Separation of duties
P57-Separation of duties	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P58-Business-driven security
P58-Business-driven security	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	P59-Linear scalability and affordable cost

P59-Linear scalability and affordable cost	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	P60-Availability and business continuity management

Q13. With respect to the DevSecOps Metrics, which category is more important? And how much more on a scale 1 to 9?

	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9	
Business	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Organization, People, and Culture
Business	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Process Capabilities
Business	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Technology
Organization, People, and Culture	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Process Capabilities
Organization, People, and Culture	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Technology
Process Capabilities	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Technology

Q14. With respect to Process Capabilities, which sub-category is more important? And how much more on a scale 1 to 9?

	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9	
Security Process	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	Security Management

Q15. With respect to Security Process, which metric is more important? And how much more on a scale 1 to 9?

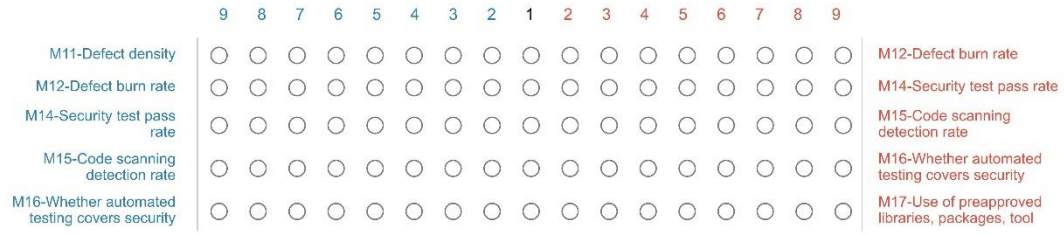
	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9	
M03-Time spent correcting mistakes in each category	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	M04-Security review performance
M04-Security review performance	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	M08-Number of continuous delivery cycles per month

Q16. With respect to Security Management, which metric is more important? And how much more on a scale 1 to 9?

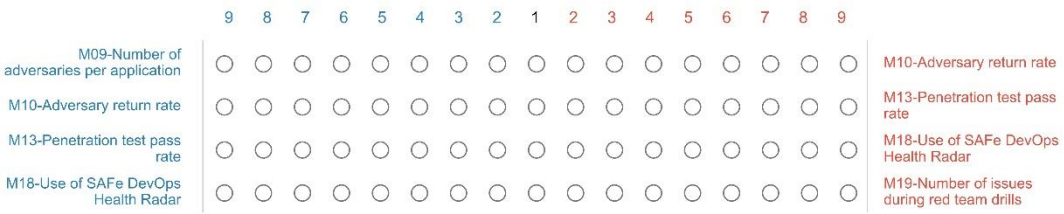
	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9	
M02-Top vulnerability	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	M05-SLA performance
M05-SLA performance	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	M06-Critical risk profiling
M06-Critical risk profiling	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	M07-Point of risk per device

Q17. With respect to Technology, which sub-category is more important? And how much more on a scale 1 to 9?

Q18. With respect to Application Security, which metric is more important? And how much more on a scale 1 to 9?



Q19. With respect to Operation Security, which metric is more important? And how much more on a scale 1 to 9?



Q20. What additional metrics that can be used for assessing the performance of DevSecOps do you want to add to this list?

Q21. Regarding to DevSecOps Metrics in GES (Global Software Engineering) context, how do you think they differ in local and global settings.

- ◯ Extremely different
- ◯ Slightly different
- ◯ Not different

○ Uncertain

Q22. What additional Global DevSecOps Metrics do you want to add to this list?

Location Data

Location: (-36.8504, 174.7675)

Source: GeoIP Estimation

Appendix C. Sample of Thematic Analysis in MLR Study

Thematic synthesis of DevSecOps definitions

21 Themes (Freq)	74 Codes [Papers contributed to the code]
Expansion to DevOps (4)	expansion to DevOps [S1-IEEE-08, S1-SC-21] (2)
	extension to DevOps [S1-SC-01] (1)
	extension of the DevOps [S1-GL_33] (1)
Dev, Sec & Ops (10)	development, operations and security teams [S1-IEEE-05, 08, 12, S1-SC-10, 21, S1-ACM-68, S1-GL-15, 19, 27] (9)
	dev/sec/ops [S1-IEEE_26] (1)
Culture (8)	culture [S1-ACM-45, S1-GL-10, 13, 26] (4)
	cultural approach [S1-IEEE-26] (1)
	cultural shift [S1-ACM-50, S1-GL-11] (2)
	shift the mindset [S1-IEEE-10] (1)
Collaboration (9)	collaboration/collaborate [S1-IEEE-08, 12, 26, S1-SC-10, 21, S1-ACM-45, 68, S1-GL-26] (8)
	team work [S1-GL-02] (1)
Breaking silos of security (4)	breaking silos of security [S1-IEEE-08, 24, 26] (3)
	break down the barrier [S1-IEEE-22] (1)
Sharing knowledge (3)	sharing that knowledge [S1-IEEE-08] (1)
	giving that knowledge to the different teams [S1-IEEE-24, 26] (2)
Shared responsibility (6)	shared responsibility [S1-GL-10, 33] (2)
	everyone's responsibility [S1-GL-10] (1)
	security is a part of everyone's job [S1-GL-12] (1)
	make everyone accountable for security [S1-GL-27] (1)
	at the top of every developer's mind [S1-GL-12] (1)
Philosophy (3)	philosophy [S1-GL-02, 19, 26] (3)
Communication (1)	communication [S1-GL-19] (1)
Combination of DevOps and SecOps (1)	combination of DevOps and SecOps [S1-GL-13]
Integration of security into DevOps (21)	incorporating security practices in the DevOps processes [S1-IEEE-08, S1-SC-21] (2)
	incorporation of security practices in a DevOps environment [S1-SC-10, 11] (2)
	IT processes with security approach [S1-ACM-04, S1-IEEE-21] (2)
	integration of security with development and operation [S1-SC-09] (1)
	integrating security principles [S1-IEEE-12] (1)
	integration of security processes and practices [S1-IEEE-10, S1-GL-10] (2)
	introduction of more security-oriented processes [S1-SC-22] (1)
	integrates continuous security into the original DevOps process [S1-IEEE-03] (1)
	injection of security principles and controls into the DevOps [S1-ACM-50] (1)
	integrating secure development best practices and methodologies into development and deployment processes [S1-IEEE-44] (1)
	integrating the software development and operation processes considering security and compliance requirements [S1-SC-11] (1)
	integrating security methods into a DevOps process [S1-GL-02] (1)
	integrating security practices within the DevOps process [S1-GL-26] (1)
	adding security components to each step of the DevOps [S1-GL-23] (1)

	bake security into the rapid-release cycles [S1-GL-11] (1)
	integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline[S1-GL-16] (1)
	built-in security [S1-GL-04] (1)
Agile (4)	Agile [S1-ACM-45, S1-IEEE-03, S1-GL-05] (3)
	smart and lightweight approach [S1-SC-31] (1)
Security is the main concern throughout the SDLC (7)	security is the main emphasis [S1-SC-14] (1)
	security is given high priority throughout the SDLC [S1-ACM-07] (1)
	a key concern throughout all phases of the development lifecycle and even post deployment [S1-SC-31] (1)
	security practices are implemented at each stage of the cycle [S1-ACM-07] (1)
	security is implemented at the right level and at right time [S1-IEEE-24] (1)
	emphasises the importance of sound information security practices [S1-GL-01] (1)
	adoption of security through the entire SDLC [S1-GL-19] (1)
Shifting security to the start (8)	puts security at the forefront of requirements [S1-IEEE-24] (1)
	shifting security to the early stages [S1-IEEE-06] (1)
	security from the start/beginning [S1-GL-04, 15, 33] (3)
	integrate security objectives as early as possible [S1-GL-10] (1)
	placing security practices early during the SDLC [S1-GL-05] (1)
	avoids any risk of security being an afterthought [S1-GL-01] (1)
Time reduction & Efficiency improvement (4)	time reduction [S1-ACM-04, S1-IEEE-21] (2)
	increase deployment rates [S1-IEEE-22] (1)
	shorten the SDLC [S1-GL-23] (1)
Security assurance (3)	maintaining a secure operational atmosphere [S1-IEEE-22] (1)
	identifying security vulnerabilities [S1-SC-31] (1)
	responsible for application security [S1-IEEE-05] (1)
Tooling (2)	reliance on operational tools [S1-ACM-45] (1)
	tooling [S1-GL-10] (1)
Automation (2)	automation/automating [S1-GL-04, 19] (2)
Security as code (1)	security as code [S1-GL-26] (1)
Quality (4)	without lost quality [S1-ACM-04, S1-IEEE-21] (2)
	quality affirmation [S1-SC-14] (1)
	high software quality [S1-GL-23] (1)
Authors of common definitions (19)	Mohan and Othmane [S1-IEEE_ 08, 26, S1-SC-09, 10, 11, 21, 22, S1-ACM-45, 68] (9)
	Rahman and Williams [S1-IEEE-08, 12, 44, S1-SC-22] (4)
	Carter [S1-IEEE-24, 26] (2)
	Carturan and Goya [S1-IEEE-21, S1-ACM-04] (2)
	Myrbakken and Colomo-Palacios [S1-IEEE-10] (1)
	Mohan, Othmane, and Kres [S1-SC-11] (1)

Thematic synthesis of DevSecOps challenges

23 Themes + 5 = 28 challenges	85 Codes (not including the ones from review papers)
C01-Cultural resistance and organisational opposition (7) *	developer resistance to integrate security protocol [S1-IEEE-08, S1-ACM-05]
	developers lose autonomy [S1-IEEE-06]
	resistance to change [S1-GL-15]
	challenge of the shifting role of security [S1-GL-37]
	organisational opposition [S1-GL-24]
	cultural resistance [S1-GL-20]
C02-Challenges of collaboration, communication and coordination (20) *	teams working towards conflicting objectives [S1-SC-08]
	insufficient monitoring of collaboration [S1-ACM-01]
	challenge of unrestricted collaboration [S1-IEEE-08, S1-ACM-05]
	coordination of security team and DevOps team [S1-IEEE-08, S1-ACM-05]
	un-trusted inputs causing isolation [S1-IEEE-08, S1-ACM-05]
	conflict between security and development [S1-IEEE-06]
	lack of common process and platform for communication, collaboration, and sharing information and feedback [S1-SC-08]
	collaboration challenges [S1-GL-28, 29]
	conflicting aims [S1-GL-38, 40]
	failing to collaborate with the InfoSec team [S1-GL-18]
	lack of coordination between InfoSec team and developers [S1-GL-19]
	gaps between DevOps and Security teams [S1-GL-20]
	disconnect between security and development [S1-GL-39]
	friction between development and security [S1-GL-13]
	communication requirements [S1-GL-15]
C03-Neglecting security (3) *	not prioritise security [S1-IEEE-06]
	focused on velocity, not security [S1-GL-17]
	neglect security [S1-GL-30]
C04-Lack of security awareness and responsibility (3) *	improving security awareness [S1-IEEE-06]
	nobody is responsible for security [S1-IEEE-06]
	security push-pull [S1-IEEE-06]
C05-Lack of security knowledge and skills, need for training (9) *	lacking security education [S1-IEEE-06]
	lacking knowledge and training [S1-IEEE-06]
	lack of security knowledge [S1-IEEE-08, S1-GL-38, 40]
	developers are not security specialists [S1-GL-15]
	being unfamiliar with common security risks [S1-GL-18]
	the skills gap [S1-GL-37]
	not enough company stakeholders are security savvy [S1-GL-39]
C06-Recruiting challenges (3)	the boundary between a specialist and generalist [S1-IEEE-06]
	recruiting challenges [S1-GL-24]
	understaffing InfoSec teams and engaging too late with the InfoSec team [S1-GL-18]
C07-Inconsistent security polices design (2) *	inconsistent security polices design [S1-ACM-05, S1-IEEE-08]
C08-Challenges in decision level (1) *	insufficient level of governance on DevSecOps adoption [S1-SC-08]
	lack of commitment of leadership and senior management for DevSecOps adoption [Myrbakken and Colomo-Palacios' MLR]

C09-Lacking confidence*	low or no confidence in DevSecOps [Myrbakken and Colomo-Palacios' MLR]
	lack of trust and skepticism [Myrbakken and Colomo-Palacios' MLR]
C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance (11) *	difficulties in integrating security practices into a fast moving DevOps pipeline without slowing down the process [S1-SC-08]
	running current product and services in parallel to its transformation to DevSecOps [S1-SC-08]
	tradeoff between security measures and CI system performance [S1-ACM-95]
	implementing security in CI/CD [S1-GL-28]
	rapid pace of change [S1-GL-29]
	faster development process [S1-GL-28]
	keep up with the pace of DevOps [S1-GL-30]
	DevOps velocity [S1-GL-37]
	Slow security testing [S1-GL-38, 40]
	interconnectedness of the DevOps process [S1-GL-28]
C11-Using unsuitable metrics (3) *	using unsuitable metrics [S1-ACM-01, 05, S1-IEEE-08]
C12-Compliance requirements (5) *	compliance requirements [S1-IEEE-07, 08, 11, S1-ACM-05, S1-GL-39]
C13-Neglecting change control in security (1) *	neglecting change control in security [S1-IEEE-08]
C14-Lack of standards (2) *	lack of security standards [S1-IEEE-08]
	lack of tool standards [S1-IEEE-06]
C15-Ignoring processes and security essentials leading to technical and security debt (1)	ignoring processes and security essentials leading to technical debt and security debt [S1-SC-08]
C16-Poor visibility of security track record (1)	poor visibility of security track record [S1-GL-19]
C17-Inadequate privileged credentials and access controls causing cyber attacks (2)	inadequate controls provide an opening for attack [S1-GL-30]
	privileged credentials used in DevOps are targeted by cyber attackers [S1-GL-17]
C18-Lack of mature tools for automation and security (19) *	lack of automated testing tools [S1-IEEE-06, 08, S1-ACM-05]
	lack of integrated testing tools [S1-IEEE-08, S1-ACM-05]
	wrong automated deployment tools [S1-IEEE-12, S1-ACM-01]
	use of immature automated deployment tools [S1-IEEE-08, 12, S1-ACM-01, 05]
	Remaining manual security testing and need for automated testing performance measures [S1-IEEE-08, S1-ACM-05]
	threat modeling scalability issue [S1-IEEE-08, S1-ACM-05]
	inefficient Static AST tools [S1-GL-19]
	manual pen-testing becomes a bottleneck [S1-GL-19]
	mismatched tools [S1-GL-15]
	tool-centric approaches to secrets management create security gaps [S1-GL-17]
C19-Complexity in managing different tools*	complexity in managing different tools [Myrbakken and Colomo-Palacios' MLR]
C20-Challenges of legacy system refactoring (4) *	challenging to automate legacy system [S1-IEEE-06]
	lack of cloud support [S1-GL-19]
	systems are not scalable [S1-GL-19]
	legacy infrastructure [S1-GL-24]
C21-Use of cloud and serverless computing brings security complications (21) *	cloud security complications [S1-SC-25, 44, S1-IEEE-06, 16, 25, 39, S1-ACM- 19, 52, 59, 66, S1-GL-24, 29, 38, 40]

	attacks due to miss-configured cloud environments [S1-IEEE-33, 42]
	security smells in Infrastructure as Code [S1-ACM-06, S1-IEEE-28, S1-SC-26]
	serverless computing [S1-GL-28]
	cloud and open source environments lead to compromise of critical information, configuration errors, compliance issues and security breaches [S1-GL-20]
C22-Containers and other tools come with their own risks (3) *	container and other tools can often be the reason for security concerns [S1-GL-20]
	workload containerisation [S1-GL-29]
	tools come with their own risks [S1-GL-30]
C23-Availability and reliability of infrastructure resources, tools, automation, and network bandwidth for fast deployment cycle*	availability and reliability of infrastructure resources, tools, automation, and network bandwidth for shorter and frequent deployment cycle [Myrbakken and Colomo-Palacios' MLR]
C24-Continuous deployment chaos (1)	continuous deployment chaos [S1-GL-19]
C25-Challenges of cost control (2)	restructuring organisation and implementing DevSecOps practices can lead to high cost such as salaries for security experts, costs on new tools [S1-IEEE-04]
	risk and cost battle [S1-IEEE-06]
C26-Conflicts between security and business (2)	security and business objectives are implemented using conflicting approaches [S1-ACM-64]
	dilemma in selection of business processes in product and service delivery for transformation to DevSecOps [S1-SC-08]
C27-Customer readiness for frequent releases*	customer readiness for applying frequent releases to production setup [Myrbakken and Colomo-Palacios' MLR]
C28-Difficulty in training users for using advanced tools*	users need to be properly trained when using advanced tools [Myrbakken and Colomo-Palacios' MLR]

Thematic synthesis of DevSecOps practices

56 Themes + 4 = 60 Practices	142 Codes (not including the ones from review papers)
P01-Cultural shift to security (3) *	cultural shift [S1-GL-41]
	change the security mindset [S1-GL-32]
	make security a priority [S1-GL-32]
P02-Improving collaboration, communication and cooperation (34) *	work collaboratively [S1-ACM-02]
	enhanced collaboration [S1-ACM-02]
	cross-departmental collaboration [S1-IEEE-04]
	collaborating development, operation and security [S1-IEEE-04, 12]
	close collaboration [S1-IEEE-12]
	collaboration within and between different teams [S1-IEEE-12]
	collaboration amongst different departments [S1-IEEE-12]
	collaboration between Dev and Ops [S1-IEEE-12]
	collaboration between Dev and Sec [S1-IEEE-12]
	collaboration between Sec and Ops [S1-IEEE-12]
	team collaboration [S1-IEEE-15]
	strong collaboration [S1-IEEE-15]
	strong communication [S1-IEEE-12]
	close communication [S1-ACM-02, S1-IEEE-09]
	communication of security requirements [S1-ACM-02]
	virtual communication [S1-ACM-02]
	face-to-face communication [S1-ACM-02]
	physical communication [S1-ACM-02]
	trust [S1-ACM-02, S1-IEEE-29]
	trustworthy [S1-ACM-02]
	trusted relationships [S1-ACM-02]
	mutual trust [S1-ACM-02]
	implicit trust [S1-ACM-02]
	trust within the teams [S1-IEEE-29]
	cross-functional collaboration [S1-GL-30]
	foster collaboration [S1-GL-25]
	open contribution and collaboration [S1-GL-24]
	collaboration and integration [S1-GL-02]
	communicate and collaborate [S1-GL-32]
	improving empathy and cooperation [S1-GL-10]
	reducing friction [S1-GL-10]
P03-Shared and collective responsibility for security (3) *	shared responsibility for security [S1-ACM-02]
	collective responsibility [S1-GL-02]
	assign security responsibility to one person from your DevOps team [S1-GL-28]
P04-Shared knowledge (3) *	knowledge sharing [S1-ACM-02]
	learn from each other [S1-GL-32]
	shared threat intelligence [S1-GL-24]
P05-Training, learning and education for security (6) *	training [S1-GL-06, 10, 32]
	cross-training [S1-GL-35]

	educate developers [S1-GL-25]
	security learning [S1-GL-14]
P06-Security champions (2) *	security champions [S1-ACM-02, S1-GL-10]
P07-Recruiting success (1) *	recruiting success [S1-GL-10]
P08- Continuous feedback loop (6) *	feedback loop [S1-ACM-15]
	continuous feedback loops [S1-GL-09, 13, 15, 22, 35]
P09-Be reactive and responsive (1)	be reactive and responsive [S1-GL-32]
P10-Shameless retrospectives (1) *	shameless retrospectives [S1-IEEE-09]
P11-Security policies (2) *	impose policy and governance [S1-GL-41]
	implement security policies [S1-GL-30]
P12-Commitment and agreement (1) *	commitment and agreement [S1-IEEE-29]
P13-Enhance transparency (2)*	transparency [S1-IEEE-29, S1-SC-09]
P14-Continuous improvement mindset *	continuous improvement mindset [Sánchez-Gordón and Colomo-Palacios' SLR]
P15-Leadership support *	Leadership support [Sánchez-Gordón and Colomo-Palacios' SLR]
P16-Shifting security to the left (early) (18) *	shifting security to the left [S1-IEEE-04, 24, 26, S1-SC-08, 11, S1-ACM-50, 81]
	moving security to the left [S1-GL-08, 09, 13, 15, 18, 31, 35, 36]
	integrate security during the planning phase [S1-GL-35]
	take a proactive approach to security [S1-GL-17]
	include security early [S1-GL-28]
P17-Security-by-Design (12) *	security by design [S1-SC-07, 08, 18, 20, 22, S1-IEEE-16, 29, 30, 36, S1-ACM-45, 69, S1-GL-31]
P18-Increase the visibility (2)	increase the visibility [S1-SC-09]
	enhance visibility [S1-GL-41]
P19-Good documentation, logging and reporting (3) *	good documentation and logging [S1-IEEE-15]
	better reporting [S1-GL-02, 19]
P20-Compliance control (6)	compliance control [S1-IEEE-11, S1-SC-27, S1-GL-10, 24]
	identify compliance requirements beforehand [S1-GL-28]
	bridging the divide between compliance and development [S1-GL-02]
P21-Risk management (9) *	risk management (including risk assessment, risk treatment and risk control) [S1-SC-11, 18, 20, 22, 26, 40, 41, S1-ACM-03, S1-IEEE-34]
P22-Vulnerability and incident management (5) *	vulnerability and incident management [S1-GL-14]
	Incident management [S1-GL-08, 10]
	vulnerability management [S1-GL-23, 30]
P23-Privilege management (3) *	least privilege controls [S1-IEEE-33]
	privileged access management [S1-GL-30]
	secure access via secrets management [S1-GL-41]
P24-Configuration management (1)	configuration management [S1-GL-10]
P25-CI/CD for patch (1)	CI/CD for patching [S1-GL-10]
P26- Define metrics (3) *	define metrics [S1-GL-06, 19]
	measurement [S1-GL-02]
P27-Software process maturity (2) *	software process maturity [S1-SC-32]
	Building Security In Maturity Model (BSIMM) model [S1-ACM-01]
P28-Define security requirements (2) *	define security requirements [S1-GL-06]
	security requirements and design [S1-GL-14]

P29-Security review and evaluation (8) *	security reviews [S1-GL-18]
	security evaluation [S1-GL-14]
	proactive security assessments [S1-GL-10]
	detect existing security flaws [S1-SC-09]
	make sure the basics of host and network security are in place [S1-SC-09]
	operational controls validation and improvement [S1-GL-14]
	host hardening [S1-GL-10]
	application-level assessment [S1-GL-10]
P30-Keep credentials safe (1)	keep credentials safe [S1-GL-06]
P31-Common weaknesses enumeration (1)	common weaknesses enumeration [S1-GL-08]
P32-Hybrid life cycles with data-security focus*	combining data security and software development life cycles [Rajapakse's SLR]
P33-Automation (93) *	Automation [S1-ACM-01, 09, 49, 71, 72, 81, 95, S1-IEEE-06, 07, 09, 10, 12, 13, 15, 20, 21, 26, 38, 41, 54, 57, S1-SC-08, 09, 11, 17, 18, 20, 22, 26, 27, 32, 40, S1-GL-02, 04, 06]
	Automated/automating test/testing [S1-ACM-01, 09, 49, 81, 95, S1-IEEE- 06, 07, 09, 10, 12, 15, 21, 26, 38, 41, 54, 57, S1-SC-08, 09, 11, 17, 18, 22, 26, 27, S1-GL-08,11,13,15, 35]
	Automated monitoring [S1-ACM-01, 71, 72, 81, S1-IEEE-07, 12, 13, 15, 21, 26, 38, S1-SC- 08, 09, 18, 20, 26, 40]
	automated/automating scans [S1-IEEE- 07, S1-SC-32]
	automated/automating code review [S1-IEEE-07, 12, S1-GL-23]
	automate as much as possible [S1-GL-25, 28]
	automate protection of business logic flaws [S1-GL-09]
	automate tools and security processes [S1-GL-17, 30]
	use automated security tools [S1-GL-41]
P34-Security-as-Code (5) *	Security-as-Code [S1-SC-08, 09, 18, S1-IEEE-06, S1-GL-32]
P35-Threat modeling (15) *	threat modeling/analysis [S1-IEEE-02, 04, 07, 11, 30, 36, 39, 61, 71, S1-SC-26, S1-GL-06, 10, 14, 25, 28]
P36-Continuous monitoring (22) *	Continuous monitoring [S1-IEEE-07, 12, 13, 15, 21, 26, 38, S1-SC-08, 09, 18, 20, 26, 40, S1-ACM-01, 15, 71, 72, 81, S1-GL-02, 06, 25, 31]
	24×7 proactive monitoring [S1-GL-24]
P37-Secure coding (6)	source code repository and scanning [S1-GL-10]
	secure coding [S1-GL-10, 14, 28]
	build preapproved code [S1-GL-18]
	conduct code dependency checks regularly [S1-GL-25]
P38-Advanced malware detection (1)	advanced malware detection employs machine learning and deep learning [S1-SC-32]
P39-Cloud security (4)	verify cloud infrastructure [S1-GL-28]
	MUSA Security DevOps framework [S1-ACM-52]
	MUSA DevOps framework for security in multi-cloud applications [S1-IEEE-16, 40]
P40-Container security (14) *	Container/Containerisation security [S1-ACM-52, S1-IEEE-55, S1-GL-28, 41]
	run container as non-root users [S1-IEEE-55, S1-SC-09, 34]
	use the latest version of image [S1-SC-42]
	conduct deep scanning of container image [S1-IEEE-04]
	enhance security of Docker [S1-IEEE-31, S1-GL-10]
	security practices in Kubernetes [S1-IEEE-18, S1-GL-10]
	version control, metadata and orchestration [S1-GL-10]

P41-Sensitive information scan (1)	sensitive information scan [S1-GL-23]
P57-Software Composition Analysis (2)	software composition analysis [S1-GL-06, 23]
P43-Red team security drills (2) *	red team security drills [S1-IEEE-04]
	red and blue team exploit testing [S1-GL-24]
P44-Fault injection (chaos engineering) (1)	Fault injection (chaos engineering) [S1-IEEE-13]
P45-RASP (4)	Runtime Application Self-Protection (RASP) [S1-SC-32, S1-GL-02, 08, 25]
P46-SAST (4) *	Static Application Security Testing (SAST) [S1-GL-02, 08, 23, 25]
P47-DAST (5)	Dynamic Application Security Testing (DAST) techniques integrated into a CI/CD pipeline [S1-IEEE-10]
	Dynamic Application Security Testing (DAST) [S1-GL-02, 08, 23, 25]
P48-IAST (5) *	Interactive Application Security Testing (IAST) [S1-IEEE-15, S1-GL-02, 08, 19, 25]
P49-Immutable-as-Code (1)	Immutable-as-code ensures the immutability of infrastructure and avoid accidental configuration drifts [S1-IEEE-33]
P50-Policy-as-Code (2)	Policy-as-Code is an attempt to code the policy itself [S1-IEEE-33, S1-GL-17]
P51-Design-as-Code (1)	Design-as-code: CAIRIS (Computer Aided Integration of Requirements and Information Security) model [S1-IEEE-36]
P52-Compliance-as-Code (1)	Compliance-as-Code [S1-GL-23]
P53-Adopting DevSecOps in microservices-based applications (8)	adopting DevSecOps in microservices-based applications [S1-IEEE-17, 43, 52, 57, 84, 86, S1-SC-15, 36]
P54-Integrate security issues within your general bug tracker (1)	Integrate security issues within your general bug tracker [S1-GL-19]
P54-Micro-segmentation (1)	micro-segmentation [S1-SC-38]
P55-Big data and behavioral analytic techniques*	obtain fast feedback from end users and predictive analytic for trends in user behaviors [Rajapakse's SLR]
P56-Consumable security services with APIs (1)	Consumable security services with APIs [S1-GL-24]
P57-Separation of duties (2)	separation of duties [S1-GL-14, 17]
P58-Business-driven security (1)	business-driven security [S1-GL-24]
P59-Linear scalability and affordable cost (1)	linear scalability and affordable cost [S1-GL-19]
P60-Availability and business continuity management (1)	availability and business continuity management [S1-GL-14]

Thematic synthesis of DevSecOps metrics

16 Themes + 4 = 20 Metrics	20 Codes (not including the ones from review papers)
M01-Security-trained rate (1)	number of developers that have gone through security-training [S1-IEEE-06]
M02-Top vulnerability (3)*	number of mistakes in different security categories [S1-IEEE-06]
	OWASP top 10 [S1-IEEE-06]
	top vulnerability types and recurring bugs [S1-GL-43]
M03-Time spent correcting mistakes in each category (1)	time spent correcting mistakes in each category [S1-IEEE-06]
M04-Security review performance (3)	whether features undergo a security review [S1-GL-18]
	whether security review slows down the development cycle [S1-GL-18]
	how well security is integrated into the delivery lifecycle [S1-GL-18]
M05-SLA performance (1)	SLA performance [S1-GL-43]
M06-Critical risk profiling (1)*	critical risk profiling [S1-GL-43]
M07-Point of risk per device*	point of risk per device [Prates' MLR]
M08-Number of continuous delivery cycles per month*	number of continuous delivery cycles per month [Prates' MLR]
M09-Number of adversaries per application (1)*	number of adversaries per application [S1-GL-43]
M10-Adversary return rate (1)*	adversary return rate [S1-GL-43]
M11-Defect density (1)*	defect density [S1-GL-43]
M12-Defect burn rate (1)*	defect burn rate [S1-GL-43]
M13-Penetration test pass rate (1)	systems that are affected by internal and external penetration testing [S1-IEEE-06]
M14-Security test pass rate (1)	security test pass rate [S1-IEEE-57]
M15-Code scanning detection rate (1)	code scanning detection rate [S1-IEEE-57]
M16-Whether automated testing covers security requirements (1)	whether automated testing covers security requirements [S1-GL-18]
M17-Use of preapproved libraries, packages, tool chains, and processes (1)	use of preapproved libraries, packages, tool chains, and processes [S1-GL-18]
M18-Use of SAFe DevOps Health Radar (1)	SAFe DevOps Health Radar measures DevOps performance, by assessing the maturity of four aspects and 16 activities of the CI/CD pipeline [S1-GL-01]
M19-Number of issues during red teaming drills*	number of issues during red teaming drills [Prates' MLR]
M20-Business metrics*	business metrics [Myrbakken and Colomo-Palacios' MLR]
	revenue [Myrbakken and Colomo-Palacios' MLR]
	key performance indicators [Myrbakken and Colomo-Palacios' MLR]

Thematic synthesis of DevSecOps tools

16 Themes + 2 = 18 Tool Groups	56 Codes/Tools
T01- Automation tools (11)	Chef [S1-IEEE-07, S1-SC-12, 20, 26], Jenkins [S1-SC-12], Ansible [S1-SC-20, S1-GL-04], Puppet [S1-SC-20], Gauntlt [S1-IEEE-06]*, SaltStack [S1-SC-01, 20]
T02-Automated code review tools (4)	Veracode Greenlight [S1-SC-01], PMD [S1-GL-23], DevSkim [S1-GL-23], FindSecBugs [S1-GL-23]
T03-Threat modeling tools (2)	IriusRisk [S1-SC-01], Microsoft threat modeling tool [S1-IEEE-39]
T04-Containerisation tools (22)	Docker [S1-SC-09, 18, 20, 29, 34, 42, 45, 48, S1-ACM-95, 99, S1-IEEE-31, 55, S1-GL-03, 10], Kubernetes [S1-ACM-52, 76, 89, S1-SC-20, 29, S1-IEEE-18, S1-GL-03, 10]
T05-Container security tools (3)	Twistlock [S1-GL-42], Notary [S1-GL-42], Aqua Security [S1-GL-42]
T06-Cloud security tools (7)	Terraform [S1-SC-12, 20, S1-IEEE-33], AppScan on Cloud [S1-GL-42], AWS Security service [S1-GL-42], ThreatModeler Cloud Edition [S1-GL-42], Trend Micro Cloud One [S1-GL-42]
T07-Sensitive information scanning tools (3)	TruffleHog [S1-GL-23], GitSecrets [S1-GL-23], Talisman [S1-GL-23]
T08-SAST tools (7)	Kiuwan [S1-SC-01], Flawfinder [S1-GL-23], Graudit [S1-GL-23], Bandit [S1-GL-23], Spotbugs [S1-GL-23], SonarQube [S1-GL-23, 42]
T09-DAST tools (7)	OWASP ZAP [S1-GL-23]*, BDD Security [S1-GL-23], Arachini [S1-GL-23]*, Nikto [S1-GL-23], Radamsa [S1-GL-23], FuzzDB [S1-GL-23], Fortify Webinspect [S1-GL-42]
T10-RAST tool (1)	Fortify Application Defender [S1-GL-42]
T11-Advanced malware detection tool (1)	CodeAI [S1-SC-01]

T12-Software composition analysis tools (3)	Retire.js [S1-GL-23], OSSAudit [S1-GL-23], OWASP Dependency-Check [S1-GL-23]
T13-Compliance-as-code tools (3)	nspec [S1-GL-23], ServerSpec [S1-GL-23], OpenSCAP [S1-GL-23]
T14-Vulnerability management tools (8)	Defect Dojo [S1-GL-23], ArcherySec [S1-GL-23], Snyk [S1-GL-10, 21], HackerOne [S1-GL-21], Claire [S1-GL-21], Stethoscope [S1-GL-21], Rapid7 Nexpose [S1-GL-21]
T15-DevOps performance measuring tool (1)	SAFe DevOps Health Radar [S1-GL-01]
T16-Monitoring and alerting tools (2)*	Suricata [S1-GL-21]
	NewRelic [S1-GL-42]*
	Nagios Icinga, Graphite, Ganglia, Cacti, Pager Duty, Sensu, Boundry, Pingdom [Mohan and Othmane's mapping study]
T17-Cyber security tools*	Tripwire [Mohan and Othmane's mapping study]
	Snort [Mohan and Othmane's mapping study]
T18-Logging tools*	PaperTrail, Logstash, Loggly, Splunk, SumoLogic [Mohan and Othmane's mapping study]

Appendix D. Datasets in Delphi-AHP Study

The datasets of responses to three survey rounds of the Delphi-AHP study are available in an open repository at zenodo.org: https://doi.org/10.5281/zenodo.16932278, including:

- Round One – The evaluation of challenges (including Dissent Analysis)
- Round Two – The evaluation of revised challenges and practices (including Dissent Analysis)
- Round Three – The evaluation of revised practices and metrics (including Dissent Analysis)

Appendix E. Research Outputs from MLR Study

Appendix E.1. MLR Included Papers and Quality Assessment Scores

White Literature Papers:

S1-ACM-01: M.G. Jaatun, Software security activities that support incident management in secure DevOps, Proceedings of the 13th International Conference on Availability, Reliability and Security. (2018). doi:10.1145/3230833.3233275.

S1-ACM-02: D. Ashenden, G. Ollis, Putting the SEC in DevSecOps: Using social practice theory to improve secure software development, New Security Paradigms Workshop 2020. (2020). doi:10.1145/3442167.3442178.

S1-ACM-03: M.G. Jaatun, D.S. Cruzes, J. Luna, DevOps for better software security in the cloud invited paper, Proceedings of the 12th International Conference on Availability, Reliability and Security. (2017). doi:10.1145/3098954.3103172.

S1-ACM-04: S.B. Carturan, D.H. Goya, A systems-of-systems security framework for requirements definition in cloud environment, Proceedings of the 13th European Conference on Software Architecture. (2019). doi:10.1145/3344948.3344977.

S1-ACM-05: S. Rafi, W. Yu, M.A. Akbar, Towards a hypothetical framework to secure devops adoption, Proceedings of the Evaluation and Assessment in Software Engineering. (2020). doi:10.1145/3383219.3383285.

S1-ACM-06: A. Rahman, M.R. Rahman, C. Parnin, L. Williams, Security smells in Ansible and Chef Scripts, ACM Transactions on Software Engineering and Methodology. 30 (2021). doi:10.1145/3408897.

S1-ACM-07: J.A. Morales, H. Yasar, A. Volkman, Implementing devops practices in highly regulated environments, Proceedings of the 19th International Conference on Agile Software Development: Companion. (2018). doi:10.1145/3234152.3234188.

S1-ACM-08: M. Anisetti, C.A. Ardagna, F. Gaudenzi, E. Damiani, A continuous certification methodology for DevOps, Proceedings of the 11th International Conference on Management of Digital EcoSystems. (2019). doi:10.1145/3297662.3365827.

S1-ACM-09: J. A. Morales, T. P. Scanlon, A. Volkmann, J. Yankel, H. Yasar, Security impacts of sub-optimal devsecops implementations in a highly regulated environment, Proceedings of the 15th International Conference on Availability, Reliability and Security. (2020). doi:10.1145/3407023.3409186

S1-ACM-15: E. Di Nitto, P. Jamshidi, M. Guerriero, I. Spais, D.A. Tamburri, A software architecture framework for quality-aware DevOps, Proceedings of the 2nd International Workshop on Quality-Aware DevOps. (2016). doi:10.1145/2945408.2945411.

S1-ACM-45: T. Lopez, H. Sharp, T. Tun, A. Bandara, M. Levine, B. Nuseibeh, ‘‘hopefully we are mostly secure’’: Views on Secure Code in professional practice, 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering. (2019). doi:10.1109/chase.2019.00023.

S1-ACM-49: S. Vost, S. Wagner, Keeping continuous deliveries safe, 2017 IEEE/ACM 39th International

Conference on Software Engineering Companion. (2017). doi:10.1109/icse-c.2017.135.

S1-ACM-50: J. Nguyen, M. Dupuis, Closing the feedback loop between UX design, software development, security engineering, and Operations, Proceedings of the 20th Annual SIG Conference on Information Technology Education. (2019).doi:10.1145/3349266.3351420.

S1-ACM-52: G.P. Fernandez, A. Brito, Secure container orchestration in the cloud, Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. (2019). doi:10.1145/3297280.3297296.

S1-ACM-59: E. Rios, E. Iturbe, M.C. Palacios, Self-healing multi-cloud application modelling, Proceedings of the 12th International Conference on Availability, Reliability and Security. (2017). doi:10.1145/3098954.3104059.

S1-ACM-64: K. Rindell, S. Hyrynsalmi, V. Leppanen, Aligning security objectives with Agile Software Development, Proceedings of the 19th International Conference on Agile Software Development: Companion. (2018). doi:10.1145/3234152.3234187.

S1-ACM-66: K.A. Torkura, M.I.H. Sukmana, C. Meinel, Integrating Continuous Security Assessments in microservices and cloud native applications, Proceedings of the 10th International Conference on Utility and Cloud Computing.(2017).doi:10.1145/3147213.3147229.

S1-ACM-68: Y. Rouf, J. Mukherjee, M. Fokaefs, M. Shtren, J. Le, M. Litoiu. Rule-based security management system for data-intensive applications, Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, IBM Corp, USA, 254–263. (2019).

S1-ACM-69: K. Tuma, D. Hosseini, K. Malamas, R. Scandariato, Inspection guidelines to identify security design flaws, Proceedings of the 13th European Conference on Software Architecture. (2019). doi:10.1145/3344948.3344995.

S1-ACM-71: M. Miglierina, D.A. Tamburri, Towards Omnia, Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion. (2017). doi:10.1145/3053600.3053629.

S1-ACM-72: J. Winter, M. Aniche, J. Cito, A.van Deursen, Monitoring-aware IDES, Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. (2019). doi:10.1145/3338906.3338926.

S1-ACM-76: L. F. Rivera, N. M. Villegas, G. Tamura, M. Jiménez, H. A. Müller. UML-Driven Automated Software Deployment, Proceedings of 28th Annual International Conference on Computer Science and Software Engineering, (2018). doi: 10.475/123-4.

S1-ACM-81: A. Wiedemann, N. Forsgren, M. Wiesche, H. Gewald, H. Krcmar, Research for practice, Communications of the ACM. 62 (2019). doi:10.1145/3331138.

S1-ACM-89: E. Yuan, Architecture interoperability and repeatability with microservices: An industry perspective, 2019 IEEE/ACM 2nd International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering. (2019). doi:10.1109/ecase.2019.00013.

S1-ACM-95: M. Hilton, N. Nelson, T. Tunnell, D. Marinov, D. Dig, Trade-offs in Continuous Integration: Assurance, security, and flexibility, Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. (2017). doi:10.1145/3106237.3106270.

S1-ACM-99: Z. Sampedro, A. Holt, T. Hauser, Continuous integration and delivery for HPC, Proceedings

of the Practice and Experience on Advanced Research Computing. (2018). doi:10.1145/3219104.3219147.

S1-IEEE-02: A. Valani, Rethinking secure devops threat modeling: The need for a dual velocity approach, 2018 IEEE Cybersecurity Development (SecDev). (2018). doi:10.1109/secdev.2018.00032.

S1-IEEE-03: K. Zunnurhain, S.R. Duclervil, A new project management tool based on devsecops, 2019 International Conference on Computational Science and Computational Intelligence. (2019). doi:10.1109/csci49370.2019.00049.

S1-IEEE-04: C. Fayollas, H. Bonnin and O. Flebus, SafeOps: A Concept of Continuous Safety, 2020 16th European Dependable Computing Conference. (2020). doi: 10.1109/EDCC51268.2020.00020. S1-IEEE-05: Z. Ahmed, S. C. Francis, Integrating security with devsecops: Techniques and challenges, Proceedings of the 2019 International Conference on Digitization. (2019). doi:10.1109/icd47981.2019.9105789.

S1-IEEE-06: N. Tomas, J. Li, H. Huang, An empirical study on culture, automation, measurement, and sharing of devsecops, Proceedings of 2019 International Conference on Cyber Security and Protection of Digital Services. (2019). doi:10.1109/cybersecpods.2019.8884935.

S1-IEEE-07: M. Z. Abrahams, J. J. Langerman, Compliance at Velocity within a DevOps Environment, 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany. (2018) doi:10.1109/ICDIM.2018.8847007.

S1-IEEE-08: S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, A. Gumaei, Prioritization based taxonomy of devops security challenges using promethee, IEEE Access 8 (2020). doi:10.1109/ACCESS.2020.2998819.

S1-IEEE-09: L. Williams, Continuously integrating security, Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment. (2018). doi:10.1145/3194707.3194717.

S1-IEEE-10: T. Rangnau, R.v. Buijtenen, F. Fransen, F. Turkmen, Continuous Security Testing: A case study on Integrating Dynamic Security Testing Tools in CI/CD pipelines, 2020 IEEE 24th International Enterprise Distributed Object Computing Conference. (2020). doi:10.1109/edoc49727.2020.00026.

S1-IEEE-11: J.R. Michener, A.T. Clager, Mitigating an oxymoron: Compliance in a DevOps Environments, 2016 IEEE 40th Annual Computer Software and Applications Conference. (2016). doi:10.1109/compsac.2016.155.

S1-IEEE-12: A. A. U. Rahman, L. Williams, Software security in devops: Synthesizing practitioners' perceptions and practices, Proceedings of the International Workshop on Continuous Software Evolution and Delivery, ACM, New York, NY, USA, 2016, pp. 70–76. doi:10.1145/2896941.2896946.

S1-IEEE-13: T.F. Düllmann, C. Paule, A. van Hoorn, Exploiting devops practices for dependable and secure continuous delivery pipelines, Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering.(2018).doi:10.1145/3194760.3194763.

S1-IEEE-15: V. Mohan, L. ben Othmane, A. Kres, BP: Security Concerns and best practices for automation of software deployment processes: An industrial case study, 2018 IEEE Cybersecurity Development (SecDev). (2018). doi:10.1109/secdev.2018.00011.

S1-IEEE-16: E. Rios, E. Iturbe, W. Mallouli, M. Rak, Dynamic Security Assurance in multi-cloud DevOps, 2017 IEEE Conference on Communications and Network Security (CNS). (2017). doi:10.1109/cns.2017.8228701.

S1-IEEE-17: A. Avritzer, Challenges and approaches for the assessment of Micro-Service Architecture Deployment Alternatives in devops: A tutorial presented at ICSA 2020, 2020 IEEE International Conference on Software Architecture Companion. (2020). doi:10.1109/icsac50368.2020.00007.

S1-IEEE-18: M.S. Islam Shamim, F. Ahamed Bhuiyan, A. Rahman, XI commandments of Kubernetes Security: A systematization of knowledge related to Kubernetes Security Practices, 2020 IEEE Secure Development (2020). doi:10.1109/secdev45635.2020.00025.

S1-IEEE-20: A. Rahman, Characteristics of defective infrastructure as code scripts in DevOps, Proceedings of the 40th International Conference on Software Engineering. (2018). doi:10.1145/3183440.3183452.

S1-IEEE-21: S. Carturan, D. Goya, Major challenges of systems-of-systems with cloud and devops – a financial experience report, 2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES). (2019). doi:10.1109/sesos/wdes.2019.00010.

S1-IEEE-22: E.C. Burkard, Usability testing within a Devsecops environment, 2020 Integrated Communications Navigation and Surveillance Conference. (2020). doi:10.1109/icns50378.2020.9222919.

S1-IEEE-24: Francois raynaud on devsecops, IEEE Software 34 (5) (2017) 93–96. doi:10.1109/ms.2017.3571578.

S1-IEEE-25: M.H. Syed, E.B. Fernandez, Cloud ecosystems support for internet of things and devops using patterns, 2016 IEEE 1st International Conference on Internet-of-Things Design and Implementation. (2016). doi:10.1109/iotdi.2015.31.

S1-IEEE-26: J. Diaz, J.E. Perez, M.A. Lopez-Pena, G.A. Mena, A. Yague, Self-service cybersecurity monitoring as enabler for devsecops, IEEE Access. 7 (2019) 100283–100295. doi:10.1109/access.2019.2930000.

S1-IEEE-28: A. Rahman, C. Parnin, L. Williams, The Seven sins: Security smells in infrastructure as code scripts, 2019 IEEE/ACM 41st International Conference on Software Engineering. (2019). doi:10.1109/icse.2019.00033.

S1-IEEE-29: P. Frijns, R. Bierwolf, T. Zijderhand, Reframing security in Contemporary Software Development Life cycle, 2018 IEEE International Conference on Technology Management, Operations and Decisions. (2018). doi:10.1109/itmc.2018.8691277.

S1-IEEE-30: L. Sion, K. Tuma, R. Scandariato, K. Yskout, W. Joosen, Towards Automated Security Design Flaw Detection, 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop. (2019). doi:10.1109/asew.2019.00028.

S1-IEEE-31: Amith Raj MP, A. Kumar, S.J. Pai, A. Gopal, Enhancing security of Docker using linux hardening techniques, 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology(2016).doi:10.1109/icatcct.2016.7911971.

S1-IEEE-33: R. Rompicharla, B.R. P. V, Continuous compliance model for hybrid multi-cloud through self-service orchestrator, 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics. (2020).doi:10.1109/icstcee49637.2020.9276897.

S1-IEEE-34: N. Ferry, P.H. Nguyen, Towards model-based continuous deployment of secure IOT Systems, 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems

Companion. (2019). doi:10.1109/models-c.2019.00093.

S1-IEEE-36: S. Faily, C. Iacob, Design as code: Facilitating collaboration between usability and Security Engineers using Cairis, 2017 IEEE 25th International Requirements Engineering Conference Workshops. (2017). doi:10.1109/rew.2017.23.

S1-IEEE-38: B.S. Farroha, D.L. Farroha, A framework for managing mission needs, compliance, and trust in the devops environment, 2014 IEEE Military Communications Conference. (2014). doi:10.1109/milcom.2014.54.

S1-IEEE-39: M.G. Jaatun, Architectural risk analysis in agile development of cloud software, 2019 IEEE International Conference on Cloud Computing Technology and Science. (2019). doi:10.1109/cloudcom.2019.00050.

S1-IEEE-40: V. Casola, A. De Benedictis, M. Rak, U. Villano, E. Rios, A. Rego, et al. Musa deployer: Deployment of Multi-cloud applications, 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises. (2017). doi:10.1109/wetice.2017.46.

S1-IEEE-41: Y. Wang, M. Pyhajarvi, M.V. Mantyla, Test Automation Process Improvement in a DevOps Team: Experience Report, 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops. (2020). doi:10.1109/icstw50294.2020.00057.

S1-IEEE-42: Tran Quang Thanh, S. Covaci, T. Magedanz, P. Gouvas, A. Zafeiropoulos, Embedding security and privacy into the development and operation of cloud applications and services, 2016 17th International Telecommunications Network Strategy and Planning Symposium. (2016). doi:10.1109/netwks.2016.7751149.

S1-IEEE-43: J. McZara, S. Kafle, D. Shin, Modeling and analysis of dependencies between microservices in devsecops, 2020 IEEE International Conference on Smart Cloud. (2020). doi:10.1109/smartcloud49737.2020.00034.

S1-IEEE-44: C. Izurieta, M. Prouty, Leveraging secdevops to tackle the technical debt associated with cybersecurity attack tactics, 2019 IEEE/ACM International Conference on Technical Debt. (2019). doi:10.1109/techdebt.2019.00012.

S1-IEEE-52: T. Soenen, S. Van Rossem, W. Tavernier, F. Vicens, D. Valocchi, P. Trakadas, et al. Insights from Sonata: Implementing and integrating a microservice-based NFV service platform with a DevOps methodology, 2018 IEEE/IFIP Network Operations and Management Symposium. (2018). doi:10.1109/noms.2018.8406139.

S1-IEEE-54: M. Johnson, D. Cummings, B. Leinwand, C. Elsberry, Continuous testing and deployment for Urban Air Mobility, 2020 AIAA/IEEE 39th Digital Avionics Systems Conference. (2020). doi:10.1109/dasc50938.2020.9256435.

S1-IEEE-55: A.J. Younge, K. Pedretti, R.E. Grant, R. Brightwell, A tale of two systems: Using containers to deploy HPC applications on supercomputers and clouds, 2017 IEEE International Conference on Cloud Computing Technology and Science. (2017). doi:10.1109/cloudcom.2017.40.

S1-IEEE-57: T.J. Wagner, T.C. Ford, Metrics to meet Security and Privacy Requirements with Agile Software Development Methods in a regulated environment, 2020 International Conference on Computing, Networking and Communications. (2020). doi:10.1109/icnc47757.2020.9049681.

S1-IEEE-61: L. Sion, D.V. Landuyt, W. Joosen, The never-ending story: On the need for Continuous Privacy Impact Assessment, 2020 IEEE European Symposium on Security and Privacy Workshops. (2020). doi:10.1109/eurospw51379.2020.00049.

S1-IEEE-67: D. Preuveneers, W. Joosen, Towards multi-party policy-based access control in federations of cloud and edge microservices, 2019 IEEE European Symposium on Security and Privacy Workshops. (2019). doi:10.1109/eurospw.2019.00010.

S1-IEEE-71: C. Paule, T.F. Dullmann, A. Van Hoorn, Vulnerabilities in continuous delivery pipelines? A case study, 2019 IEEE International Conference on Software Architecture Companion. (2019). doi:10.1109/icsa-c.2019.00026.

S1-IEEE-84: J. Bogner, J. Fritzsch, S. Wagner, A. Zimmermann, Microservices in industry: Insights Into Technologies, characteristics, and software quality, 2019 IEEE International Conference on Software Architecture Companion. (2019). doi:10.1109/icsa-c.2019.00041.

S1-IEEE-86: A. Luntovskyy, B. Shubyn, Highly-distributed systems based on micro-services and their construction paradigms, 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering. (2020). doi:10.1109/tcset49122.2020.235378.

S1-SC-01: A. Sen, Devops, devsecops, aiops- paradigms to it operations, Lecture Notes in Electrical Engineering. (2021). doi:10.1007/978-981-15-7804-5-16.

S1-SC-06: G. Siewruk, W. Mazurczyk, A. Karpiński, Security assurance in DevOps methodologies and related environments, INTL Journal of Electronics and Telecommunications, 65 (2019) 211-216. doi: 10.24425/ijet.2019.126303.

S1-SC-07: V. Casola, A. De Benedictis, M. Rak, G. Salzillo, A cloud secdevops methodology: From design to testing, Communications in Computer and Information Science. (2020) 317–331. doi:10.1007/978-3-030-58793-2-26.

S1-SC-08: R. Kumar, R. Goyal, Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over Cloud (ADOC), Computers and Security. 97 (2020) 101967. doi:10.1016/j.cose.2020.101967.

S1-SC-09: K.V.D.Kiran, P.J.R.Shalem Raju, Performance Analysis of Automation Monitoring System shifting from devops to devsecops, International Journal of Emerging Trends in Engineering Research. 8 (2020) 5128–5134. doi:10.30534/ijeter/2020/40892020.

S1-SC-10: F. Moyón, R. Soares, M. Pinto-Albuquerque, D. Mendez, K. Beckers, Integration of security standards in DevOps Pipelines: An industry case study, Product-Focused Software Process Improvement. (2020) 434–452. doi:10.1007/978-3-030-64148-1-27.

S1-SC-11: X. Larrucea, A. Berreteaga, I. Santamaria, Dealing with security in a real devops environment, Communications in Computer and Information Science. (2019) 453–464. doi:10.1007/978-3-030-28005-5-35.

S1-SC-12: S. Vignesh, B.R. Kanna, AWS Infrastructure Automation and Security Prevention using DevOps, Advances in Intelligent Systems and Computing. (2020) 537–549. doi:10.1007/978-981-15-0199-9-46.

S1-SC-14: R. Ravinder, V. Sucharita, A Secure Cloud Service Deployment Framework for DevOps, Indonesian Journal of Electrical Engineering

and Computer Science. 21 (2021) 874. doi:10.11591/ijeecs.v21.i2.pp874-885.

S1-SC-15: U. Zdun, E. Wittern, P. Leitner, Emerging trends, challenges, and experiences in DevOps and microservice apis, IEEE Software. 37 (2020) 87–91. doi:10.1109/ms.2019.2947982.

S1-SC-17: L. Bass, The software architect and DevOps, IEEE Software. 35 (2018) 8–10. doi:10.1109/ms.2017.4541051.

S1-SC-18: E. Zheng, P. Gates-Idem, M. Lavin, Building a virtually air-gapped secure environment in AWS, Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security. (2018). doi:10.1145/3190619.3190642.

S1-SC-19: S. Schork, F. Zahid, D. Pradhan, S. Kicin, A. Schwichtenberg, Building an open-source Cross-Cloud devops stack for a CRM enterprise application: A case study, IFIP Advances in Information and Communication Technology. (2019) 3–11. doi:10.1007/978-3-030-20883-7-1.

S1-SC-20: S.D. Duque Anton, D. Fraunholz, D. Krohmer, D. Reti, H.D. Schotten, F. Selgert, et al. Creating it from scratch: A practical approach for enhancing the security of IOT-Systems in a devopsenabled software development environment, Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops. (2020) 266–281.doi:10.1007/978-3-030-55583-2-20.

S1-SC-21: M.A. Akbar, S. Mahmood, M. Shafiq, A. Alsanad, A. Gumaei, Identification and prioritization of devops success factors using Fuzzy-AHP approach, Soft Computing. (2020).doi:10.1007/s00500-020-05150-w.

S1-SC-22: V. Casola, A. De Benedictis, M. Rak, U. Villano, A novel security-by-design methodology: Modeling and assessing security by SLAS with a quantitative approach, Journal of Systems and Software. 163 (2020) 110537. doi:10.1016/j.jss.2020.110537.

S1-SC-25: Y. Verginadis, I. Patiniotakis, M. Prusinski, M. Rozanska, S. Schork, G. Mentzas, A security and privacy-preserving path for enhancing information systems that manage Cross-Cloud Applications, Advances in Intelligent Systems and Computing. (2020) 1119–1132. doi:10.1007/978-3-030-44038-1-103.

S1-SC-26: S. Almuairfi, M. Alenezi, Security controls in infrastructure as code, Computer Fraud and Security. (2020) 13–19. doi:10.1016/s1361-3723(20)30109-3.

S1-SC-27: C. Dyess, Maintaining a balance between agility and security in the cloud, Network Security. (2020) 14–17. doi:10.1016/s1353-4858(20)30031-3.

S1-SC-29: N.C. Mendonca, P. Jamshidi, D. Garlan, C. Pahl, Developing self-adaptive microservice systems: Challenges and directions, IEEE Software. 38 (2021) 70–79. doi:10.1109/ms.2019.2955937.

S1-SC-31: B. Fitzgerald, K.-J. Stol, Continuous Software Engineering and beyond: Trends and challenges, Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014. (2014). doi:10.1145/2593812.2593813.

S1-SC-32: E. Amoroso, Recent progress in software security, IEEE Software. 35 (2018) 11–13. doi:10.1109/ms.2018.1661316.

S1-SC-34: A. Martin, S. Raponi, T. Combe, R. Di Pietro, Docker ecosystem – vulnerability analysis, Computer Communications. 122 (2018) 30–43. doi:10.1016/j.comcom.2018.03.011.

S1-SC-36: F. Boyer, X. Etchevers, N. de Palma, X. Tao, Architecture-based automated updates of

distributed microservices, Service-Oriented Computing. (2018) 21–36. doi:10.1007/978-3-030-03596-9-2.

S1-SC-38: D. Klein, Micro-segmentation: Securing Complex Cloud Environments, Network Security. (2019) 6–10. doi:10.1016/s1353-4858(19)30034-0.

S1-SC-40: N. Ferry, J. Dominiak, A. Gallon, E. Gonzalez, E.iderIturbe, S. Lavirotte, S. Martinez, A. Metzger, V. Muntes-Mulero, P. H. Nguyen, A. Palm, A. Rego, E. Rios, D. Riviera, A. Solberg, H. Song, J. Tigli, T. Winter, Development and operation of trustworthy smart IoT systems: the ENACT framework, DEVOPS 2019, (2020) 121–138, doi: 10.1007/978-3-030-39306-9-9

S1-SC-41: T. Pawlik, P.H. Meland, T. Stalhane, G.K. Hanssen, The agile RAMSS lifecycle for the future, Proceedings of the 29th European Safety and Reliability Conference. (2019). doi:10.3850/978-981-11-2724-3-0170-cd.

S1-SC-42: S. Kitajima, A. Sekiguchi, Latest image recommendation method for automatic base image update in dockerfile, Service-Oriented Computing. (2020) 547–562. doi:10.1007/978-3-030-65310-1-40.

S1-SC-44: J. Sandobalin, E. Insfran, S. Abrahao, Towards model-driven infrastructure provisioning for multiple clouds, Lecture Notes in Information Systems and Organisation. (2019) 207–225. doi:10.1007/978-3-030-22993-1-12.

S1-SC-45: S. Sugandi, I. Riadi, A. Sugandi, Forensic analysis of docker swarm cluster using GRR Rapid Response Framework, International Journal of Advanced Computer Science and Applications. 10 (2019). doi:10.14569/ijacsa.2019.0100260.

S1-SC-48: S. Abraham, A.K. Paul, R.I. Khan, A.R. Butt, On the use of containers in high performance computing environments, 2020 IEEE 13th International Conference on Cloud Computing. (2020). doi:10.1109/cloud49709.2020.00048.

S2-ACM-04: R. K. Gupta, M. Venkatachalapathy, F. K. Jeberla, Challenges in adopting continuous delivery and devops in a globally distributed product team: A case study of a healthcare organization, Proceedings of 2019 ACM/IEEE 14th International Conference on Global Software Engineering. (2019). doi:10.1109/ICGSE.2019.00020.

S2-ACM-05: M. Viggiato, J. Oliveira, E. Figueiredo, P. Jamshidi, C. Kastner, Understanding similarities and differences in software development practices across domains, Proceedings of 2019 ACM/IEEE 14th International Conference on Global Software Engineering. (2019). doi:10.1109/icgse.2019.00013.

Grey Literature Articles (accessed June 30, 2021)

S1-GL-01: DevOps, Scaled Agile Framework. (2021). https://www.scaledagileframework.com/devops/.

S1-GL-02: What is the difference between DevOps and DevSecOps?, PVS. (2020). https://pvs-studio.com/en/blog/posts/0710/.

S1-GL-03: M. Foster, DevOps vs. devsecops - here's how they fit together, Red Hat OpenShift Makes Container Orchestration Easier. (2021). https://www.openshift.com/blog/devops-vs.-devsecops-hereshow-they-fit-togethe.

S1-GL-04: What is DevSecOps?, Red Hat - We Make Open Source Technologies for the Enterprise. (2018). https://www.redhat.com/en/topics/devops/what-is-devsecops.

S1-GL-05: A. Singh, DevOps vs devsecops – what is the difference? Security Boulevard. (2020). https://securityboulevard.com/2020/08/devops-vs-devsecops-what-is-the-difference/.

S1-GL-06: Microsoft Security devops, Microsoft Security DevOps. (n.d.). https://www.microsoft.com/en-us/securityengineering/devsecops.

S1-GL-07: Security – Disciplined Agile (DA) - PMI, (n.d.). https://www.pmi.org/disciplined-agile/process/security.

DevSecOps? AppDynamics. (2021). https://www.appdynamics.com/blog/product/devops-vs-devsecops/.

S1-GL-09: E. Miller, Difference between DevOps and devsecops. Invensis Learning Blog. (2019). https://www.pmi.org/disciplined-agile/process/security.

S1-GL-10: What is DevSecOps?: Devsecops model, Snyk. (2021). https://snyk.io/devsecops/.

S1-GL-11: L. Constantin, What is devsecops? Why it's hard to do well? CSO Online. (2020). https://www.csoonline.com/article/3245748/what-is-devsecops-developing-more-secure-applications.html.

S1-GL-12: K. Magowan, What is devsecops? Combining development, Security and Operations, BMC Blogs. (2020). https://www.bmc.com/blogs/devops-devsecops/.

S1-GL-13: M, Preston, DevOps VS DevSecOps: The differences. (2020). https://www.clouddefense.ai/blog/devops-vs-devsecops-the-differences.

S1-GL-14: M. Spisak and J. Darwin, Secure DevOps architecture. IBM. (n.d.). https://www.ibm.com/cloud/architecture/architectures/secure-devops-arch/.

S1-GL-15: What is DevSecOps: Devops security tools: Imperva, Learning Center. (2021). https://www.imperva.com/learn/application-security/devsecops-devops-security/.

S1-GL-16: K. Zettler, DevSecOps Tools, Atlassian. (2021). https://www.atlassian.com/devops/devops-tools/devsecops-tools.

S1-GL-17: DevOps security, CyberArk. (2021). https://www.cyberark.com/what-is/devops-security/.

S1-GL-18: DevOps Tech: Shifting left on security, Google. (2021). https://cloud.google.com/architecture/devops/devops-tech-shifting-left-on-security.

S1-GL-19: R. Velasco, DevSecOps: The 7 key factors to secure your DevOps practice, Hdiv Security. (2020). https://hdivsecurity.com/bornsecure/devsecops-the-7-key-factorsto-secure-your-devops-practice/.

S1-GL-20: VeritisAdmin, DevOps security: An overview of Challenges and Best Practices, Go to Veritis Group Inc. (n.d.). https://www.veritis.com/blog/devops-security-an-overview-of-challenges-and-bestpractices/.

S1-GL-21: I. Eldridge, SecDevOps: Injecting Security into DevOps Processes, New Relic. (2018). https://newrelic.com/blog/best-practices/what-is-secdevops.

S1-GL-22: S. Bocetta, How to seamlessly evolve DevOps into devsecops, InfoQ. (2019). https://www.infoq.com/articles/evolve-devops-devsecops/.

S1-GL-23: P. Academy, DevSecOps: Integrating security with DevOps, Medium. (2021). https://blog.pentesteracademy.com/devsecops-learning-path-integrating-security-with-devops-

1cc03670552f.

S1-GL-24: B. Dobran, Why you should be using devops security to deliver secure software, PhoenixNAP Blog. (2019). https://phoenixnap.com/blog/devops-security-best-practice.

S1-GL-25: Top 10 devsecops best practices for building secure software: Synopsys, Application Security Blog. (n.d.). https://codedx.com/blog/how-to-join-devops-and-security-best-practices-in-devsecops/.

S1-GL-26: What is DevSecOps? Sumo Logic. (2019). https://www.sumologic.com/insight/devsecops-rugged-devops/.

S1-GL-27: What is DevSecOps?, Forcepoint. (2021). https://www.forcepoint.com/cyber-edu/devsecops.

S1-GL-28: G. Maayan, DevOps security challenges and how to overcome them, CCSI. (2019). https://www.ccsinet.com/blog/devops-security-challenges/.

S1-GL-29: A. Uss, DevOps security challenges and best practices, Snyk. (2021). https://snyk.io/learn/devops-security.

S1-GL-30: DevOps security challenges and how to deal with them: Scalyr, SentinelOne. (2019). https://www.sentinelone.com/blog/devopssec-challenges/.

S1-GL-31: Why security testing should be a part of the DevOps process, 6point6. (2021). https://6point6.co.uk/insights/why-securitytesting-should-be-a-part-of-the-devops-process/.

S1-GL-32: P. Cheslock, How to integrate security into a DevOps World, Threat Stack. (2021). https://www.threatstack.com/blog/how-to-integrate-security-intoa-devops-world.

S1-GL-33: L. Terquem, How to apply devops principles to increase security? (2020). https://www.padok.fr/en/blog/devsecops-security.

S1-GL-34: C. Brimhall, Closer than you think: Bridging the devops security gap, Anitian. (2019). https://www.anitian.com/closer-thanyou-think-bridging-the-devops-security-gap/.

S1-GL-35: M. Rimkus, From DevOps to devsecops: Securing the CI/CD pipeline, Cherry Servers. (2020). https://blog.cherryservers.com/from-devops-to-devsecops-securing-the-cicd-pipeline.

S1-GL-36: F. Reimer, Cybersecurity for Business Leaders, Security-RoundTable.org. (n.d.). https://www.securityroundtable.org/.

S1-GL-37: A. Arampatzis, Why is it such a challenge to integrate security into devops?, DATAVERSITY. (2021). https://www.dataversity.net/why-is-it-such-a-challenge-to-integrate-security-into-devops/.

S1-GL-38: H. Bavati, From DevOps to DevSecOps: The Security Challenges of DevOps. Datafloq. (2019). https://datafloq.com/read/from-devops-devsecops-security-challenges.

S1-GL-39: R. Annadi, Overcoming the Top 3 DevOps Security Challenges. Devopsdigest. (2020). https://www.devopsdigest.com/overcoming-the-top-3-devops-security-challenges.

S1-GL-40: S. Ben-Hador, From devops to devsecops: The security challenges of devops, Exabeam. (2019). https://www.exabeam.com/information-security/devsecops-and-the-security-challenges-of-devops/.

S1-GL-41: M. Vernon, Devsecops: The intersection of devops and security, Victorops Blog. (2019). https://victorops.com/blog/devsecops-the-intersection-of-devops-and-security.

S1-GL-42: T. Blogumas, Top 15 devsecops tools for an enterprise CI/CD pipeline, Medium. (2020). https://levelup.gitconnected.com/top-15-devsecops-tools-for-an-enterprise-ci-cd-pipeline-bd865b47ed5f.

S1-GL-43: E. Chickowski, Seven winning DevSecOps metrics security should track. (2018). https://businessinsights.bitdefender.com/seven-winning-devsecops-metrics-security-should-track.

New Literature from Confirmatory Search (2021-2022)

CS-ACM-01: R.N. Rajapakse, M. Zahedi, M.A. Babar, An empirical analysis of practitioners' Perspectives on Security Tool Integration into DevOps, Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. (2021). doi:10.1145/3475716.3475776.

CS-ACM-02: D. Gonzalez, P.P. Perez, M. Mirakhorli, Barriers to shift-left security, Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. (2021).doi:10.1145/3475716.3475786.

CS-ACM-03: R. Brasoveanu, Y. Karabulut, I. Pashchenko, Security maturity self-assessment framework for software development lifecycle, Proceedings of the 17th International Conference on Availability, Reliability and Security.(2022).doi:10.1145/3538969.3543806.

CS-ACM-04: L. Liu, D. Xie, Y.C. Cheng, G. Li, Architecture scheme of devops for Cross Network and multiple environment collaboration, The 5th International Conference on Computer Science and Application Engineering. (2021). doi:10.1145/3487075.3487116.

CS-IEEE-01: S. Throner, H. Hutter, N. Sanger, M. Schneider, S. Hanselmann, P. Petrovic, et al. An advanced devops environment for Microservice-based applications, 2021 IEEE International Conference on Service-Oriented System Engineering. (2021). doi:10.1109/so se52839.2021.00020.

CS-IEEE-02: S.F. Ahamed, M. Dhar M S, S.K. Kishore, M.P. Borawake, T.D. R, M. Thenmozhi, DevOps security and privacy in the development of Multi-cloud Applications, 2022 International Conference on Electronics and Renewable Systems. (2022). doi:10.1109/icears53 579.2022.9752387.

CS-IEEE-03: A. Sojan, R. Rajan, P. Kuvaja, Monitoring solution for cloud-native devsecops, 2021 IEEE 6th International Conference on Smart Cloud. (2021). doi:10.1109/smartcloud52277.2021.00029.

CS-IEEE-04: F. Angermeir, M. Voggenreiter, F. Moyon, D. Mendez, Enterprise-driven open source software: A case study on security automation, 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice. (2021). doi:10.1109/icse-seip52600.2021.00037.

CS-IEEE-05: A. Ibrahim, A.H. Yousef, W. Medhat, DevSecOps: A security model for infrastructure as code over the cloud, 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference. (2022). doi:10.1109/miucc55081.2022.9781709.

CS-IEEE-06: Y. Yang, W. Shen, B. Ruan, W. Liu, K. Ren, Security challenges in the container cloud, 2021 3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications. (2021). doi:10.1109/tpsisa52974.2021.00016.

CS-SC-01: M.A. Akbar, K. Smolander, S. Mahmood, A. Alsanad, Toward successful DevSecOps in software development organizations: A decision-making framework, Information and Software Technology. 147 (2022) 106894. doi:10.1016/j.infsof.2022.106894.

CS-SC-02: Nisha T. N., A. Khandebharad, Migration from devops to devsecops, International Journal of

Cloud Applications and Computing. 12 (2022) 1–15. doi:10.4018/ijcac.2022010102.

CS-SC-03: R.N. Rajapakse, M. Zahedi, M.A. Babar, H. Shen, Challenges and solutions when adopting DevSecOps: A systematic review, Information and Software Technology. 141 (2022) 106700. doi:10.1016/j.infsof.2021.106700.

CS-GL-01: S. Ingalls, Best DevSecOps Tools for 2022: eSecurity Planet, ESecurityPlanet. (2022). https://www.esecurityplanet.com/products/devsecops-tools/.

CS-GL-02: What is DevSecOps? JFrog. (2022). https://jfrog.com/devops-tools/what-is-devsecops/.

CS-GL-03: A. Neto, What is devsecops: Top 5 automation tools for CI pipelines, RSS. (n.d.). https://bluelight.co/blog/what-is-devsecops.

CS-GL-04: DevSecOps Best practices, Tigera. (2022). https://www.tigera.io/learn/guides/devsecops/devsecops-best-practices/.

CS-GL-05: S. Manjaly, The top 10 best devsecops tools for 2022, IT Management Software. (2022). https://blog.invgate.com/devsecops-tools.

CS-GL-06: J. Hirschauer, Top 10 best practices for devsecops, Harness.io. (2022). https://harness.io/blog/best-practices-devsecops.

CS-GL-07: M. Hales, Devsecops challenges, DevSecOps Challenges. (2021). https://www.adaptavist.com/blog/8-common-devsecops-challenges-and-how-to-overcome-them

Quality Assessment Scores

White Literature								
Paper ID	Criteria							
	Overall (18)	Authority (3)	Methodology (4)	Objectivity (3)	Novelty (2)	Impact (1)	Publication Date (1)	Literature Type (4)
S1-ACM-01	15	2	3	3	2	0	1	4
S1-ACM-02	14	1	3	3	2	0	1	4
S1-ACM-03	15	3	2	3	2	1	1	4
S1-ACM-04	12	1	2	3	2	0	1	4
S1-ACM-05	11	1	2	2	1	0	1	4
S1-ACM-06	16	3	3	3	1	1	1	4
S1-ACM-07	15	3	3	2	1	1	1	4
S1-ACM-08	12	1	2	2	1	1	1	4
S1-ACM-09	16	3	3	3	1	1	1	4
S1-ACM-15	13	1	3	2	1	1	1	4
S1-ACM-45	13	1	3	2	1	1	1	4
S1-ACM-49	12	1	2	2	1	1	1	4
S1-ACM-50	11	1	2	1	1	1	1	4
S1-ACM-52	11	1	2	2	1	0	1	4
S1-ACM-59	11	1	2	2	1	0	1	4
S1-ACM-64	12	1	2	2	1	1	1	4
S1-ACM-66	14	1	2	3	2	1	1	4
S1-ACM-68	12	1	2	2	2	0	1	4

S1-ACM-69	11	1	2	2	1	0	1	4
S1-ACM-71	11	1	2	2	1	0	1	4
S1-ACM-72	11	1	2	2	1	0	1	4
S1-ACM-76	11	1	2	2	1	0	1	4
S1-ACM-81	12	1	2	2	1	1	1	4
S1-ACM-89	11	1	2	2	1	0	1	4
S1-ACM-95	14	1	3	2	2	1	1	4
S1-ACM-99	11	1	2	2	1	0	1	4
S1-IEEE-02	11	1	2	2	1	0	1	4
S1-IEEE-03	11	1	2	2	1	0	1	4
S1-IEEE-04	11	1	2	2	1	0	1	4
S1-IEEE-05	11	1	2	2	1	0	1	4
S1-IEEE-06	13	1	3	2	1	1	1	4
S1-IEEE-07	12	1	3	2	1	0	1	4
S1-IEEE-08	13	1	3	2	1	1	1	4
S1-IEEE-09	15	3	3	2	1	1	1	4
S1-IEEE-10	13	1	2	2	2	1	1	4
S1-IEEE-11	12	1	3	2	1	0	1	4
S1-IEEE-12	16	3	3	3	1	1	1	4
S1-IEEE-13	15	3	3	2	1	1	1	4
S1-IEEE-15	16	3	3	3	1	1	1	4
S1-IEEE-16	13	2	2	2	1	1	1	4
S1-IEEE-17	12	1	2	2	2	0	1	4
S1-IEEE-18	14	2	2	2	2	1	1	4
S1-IEEE-20	15	2	3	2	2	1	1	4
S1-IEEE-21	11	1	2	2	1	0	1	4
S1-IEEE-22	11	1	2	2	1	0	1	4
S1-IEEE-24	12	1	2	2	1	1	1	4
S1-IEEE-25	13	2	2	2	1	1	1	4
S1-IEEE-26	14	2	3	2	1	1	1	4
S1-IEEE-28	17	3	3	3	2	1	1	4
S1-IEEE-29	15	2	3	2	2	1	1	4
S1-IEEE-30	16	3	3	2	2	1	1	4
S1-IEEE-31	14	2	2	2	2	1	1	4
S1-IEEE-33	12	1	2	2	2	0	1	4
S1-IEEE-34	14	2	2	2	2	1	1	4
S1-IEEE-36	13	2	2	2	2	0	1	4
S1-IEEE-38	14	2	2	2	2	1	1	4
S1-IEEE-39	13	2	2	2	2	0	1	4
S1-IEEE-40	16	3	3	2	2	1	1	4
S1-IEEE-41	14	2	2	2	2	1	1	4
S1-IEEE-42	14	2	2	2	2	1	1	4
S1-IEEE-43	12	1	2	2	2	0	1	4
S1-IEEE-44	14	2	2	2	2	1	1	4

S1-IEEE-52	15	2	3	2	2	1	1	4
S1-IEEE-54	12	1	2	2	2	0	1	4
S1-IEEE-55	17	3	3	3	2	1	1	4
S1-IEEE-57	12	1	2	2	2	0	1	4
S1-IEEE-61	14	2	3	2	2	0	1	4
S1-IEEE-67	15	2	3	2	2	1	1	4
S1-IEEE-71	15	2	3	2	2	1	1	4
S1-IEEE-84	16	3	3	2	2	1	1	4
S1-IEEE-86	13	2	2	2	2	0	1	4
S1-SC-01	13	2	2	2	1	1	1	4
S1-SC-06	12	2	2	2	1	0	1	4
S1-SC-07	14	3	3	2	1	0	1	4
S1-SC-08	14	2	3	2	1	1	1	4
S1-SC-09	12	2	2	2	1	0	1	4
S1-SC-10	11	1	2	2	1	0	1	4
S1-SC-11	12	2	2	2	1	0	1	4
S1-SC-12	12	1	2	2	2	0	1	4
S1-SC-14	12	1	2	2	2	0	1	4
S1-SC-15	16	3	3	2	2	1	1	4
S1-SC-17	16	3	3	2	2	1	1	4
S1-SC-18	12	1	2	2	2	0	1	4
S1-SC-19	13	2	2	2	2	0	1	4
S1-SC-20	13	3	2	2	1	0	1	4
S1-SC-21	15	3	3	2	1	1	1	4
S1-SC-22	15	3	3	2	1	1	1	4
S1-SC-25	12	2	2	2	1	0	1	4
S1-SC-26	12	2	2	2	1	0	1	4
S1-SC-27	11	1	2	2	1	0	1	4
S1-SC-29	16	3	3	2	2	1	1	4
S1-SC-31	15	3	2	2	2	1	1	4
S1-SC-32	11	1	2	1	1	1	1	4
S1-SC-34	15	3	2	2	2	1	1	4
S1-SC-36	13	2	2	2	2	0	1	4
S1-SC-38	12	1	2	2	2	0	1	4
S1-SC-40	13	2	2	2	2	0	1	4
S1-SC-41	13	2	2	2	2	0	1	4
S1-SC-42	12	1	2	2	2	0	1	4
S1-SC-44	14	2	3	2	2	0	1	4
S1-SC-45	14	2	2	2	2	1	1	4
S1-SC-48	13	1	3	2	1	1	1	4
S2-ACM-04	14	1	3	2	2	1	1	4
S2-ACM-05	13	1	3	2	2	0	1	4
CS-ACM-01	16	3	3	3	1	1	1	4
CS-ACM-02	12	2	2	2	1	0	1	4

CS-ACM-03	12	1	2	2	2	0	1	4
CS-ACM-04	12	1	2	2	2	0	1	4
CS-IEEE-01	13	1	2	2	2	1	1	4
CS-IEEE-02	12	1	2	2	2	0	1	4
CS-IEEE-03	13	1	2	2	2	1	1	4
CS-IEEE-04	14	2	2	2	2	1	1	4
CS-IEEE-05	14	2	2	2	2	1	1	4
CS-IEEE-06	13	1	2	2	2	1	1	4
CS-SC-01	16	3	3	3	1	1	1	4
CS-SC-02	12	1	2	2	2	0	1	4
CS-SC-03	16	3	3	3	1	1	1	4

Grey Literature

Paper ID	**Criteria**						
	Overall (18)	**Authority (3)**	**Relevance (7)**	**Novelty (2)**	**Impact (1)**	**Posted Date (1)**	**Literature Type (4)**
S1-GL-01	12	2	5	1	1	1	2
S1-GL-02	11	1	6	1	1	1	1
S1-GL-03	11	1	6	1	1	1	1
S1-GL-04	14	3	6	1	1	1	2
S1-GL-05	12	1	6	1	1	1	2
S1-GL-06	15	3	7	2	1	0	2
S1-GL-07	12	2	5	2	1	0	2
S1-GL-08	11	1	6	1	1	1	1
S1-GL-09	12	2	5	2	1	1	1
S1-GL-10	15	2	7	2	1	1	2
S1-GL-11	15	2	7	2	1	1	2
S1-GL-12	13	2	7	1	1	1	1
S1-GL-13	11	2	5	1	1	1	1
S1-GL-14	15	3	7	2	1	0	2
S1-GL-15	14	1	7	2	1	1	2
S1-GL-16	15	2	7	2	1	1	2
S1-GL-17	13	1	7	1	1	1	2
S1-GL-18	16	3	7	2	1	1	2
S1-GL-19	13	1	7	1	1	1	2
S1-GL-20	12	1	7	2	1	0	1
S1-GL-21	14	2	7	2	1	1	1
S1-GL-22	14	2	6	2	1	1	2
S1-GL-23	11	1	6	1	1	1	1
S1-GL-24	11	2	5	1	1	1	1
S1-GL-25	12	1	7	2	1	0	1
S1-GL-26	11	1	6	1	1	1	1
S1-GL-27	11	1	6	1	1	1	1
S1-GL-28	14	2	7	2	1	1	1
S1-GL-29	14	1	7	2	1	1	2

S1-GL-30	12	1	7	1	1	1	1
S1-GL-31	11	1	5	2	1	1	1
S1-GL-32	11	1	6	1	1	1	1
S1-GL-33	11	1	6	1	1	1	1
S1-GL-34	12	1	6	1	1	1	2
S1-GL-35	11	1	6	1	1	1	1
S1-GL-36	11	1	7	1	1	0	1
S1-GL-37	13	2	7	2	1	1	2
S1-GL-38	11	1	6	1	1	1	1
S1-GL-39	11	1	6	1	1	1	1
S1-GL-40	11	1	6	1	1	1	1
S1-GL-41	12	1	7	1	1	1	1
S1-GL-42	12	1	6	2	1	1	1
S1-GL-43	13	1	6	2	1	1	2
CS-GL-01	11	1	6	1	1	1	1
CS-GL-02	12	1	7	1	1	1	1
CS-GL-03	12	1	6	2	1	1	1
CS-GL-04	12	1	7	1	1	1	1
CS-GL-05	11	1	6	1	1	1	1
CS-GL-06	12	1	7	1	1	1	1
CS-GL-07	12	1	7	1	1	1	1

Appendix E.2. Glossary of Findings

DevSecOps Challenges			
Category	**Challenge/Theme**	**Explanation**	**Codes [Papers contributed to the code]**
Organisation, People, and Culture	C01-Cultural resistance and organisational opposition	Cultural resistance and organisational opposition to change, e.g., developers' resistance to integrate security due to losing autonomy.	Developer resistance to integrate security protocol [S1-IEEE-08, S1-ACM-05] Developers lose autonomy [S1-IEEE-06] Resistance to change [S1-GL-15] Challenge of the shifting role of security [S1-GL-37] Organisational opposition [S1-GL-24] Cultural resistance [S1-GL-20]
	C02-Challenges of collaboration, communication and coordination	Friction/gap/disconnect/isolation/conflict between security, development and operation teams.	Teams working towards conflicting objectives [S1-SC-08] Insufficient monitoring of collaboration [S1-ACM-01] Challenge of unrestricted collaboration [S1-IEEE-08, S1-ACM-05] Coordination of security team and DevOps team [S1-IEEE-08, S1-ACM-05] Untrusted inputs causing isolation [S1-IEEE-08, S1-ACM-05] Conflict between security and development [S1-IEEE-06] Collaboration challenges [S1-GL-28, 29] Conflicting aims [S1-GL-38, 40] Failing to collaborate with the InfoSec team [S1-GL-18] Lack of coordination between InfoSec team and developers [S1-GL-19] Gaps between DevOps and Security teams [S1-GL-20] Disconnect between security and development [S1-GL-39] Friction between development and security [S1-GL-13] Communication requirements [S1-GL-15] Lack of common process and platform for communication, collaboration, and sharing information and feedback [S1-SC-08]
	C03-Neglecting security	Focused on velocity, not prioritise security.	Not prioritise security [S1-IEEE-06] Focused on velocity, not security [S1-GL-17] Neglect security [S1-GL-30]
	C04-Lack of security awareness and responsibility	Nobody is aware of security and is responsible for security.	Security awareness [S1-IEEE-06] Nobody is responsible for security [S1-IEEE-06] Security push-pull [S1-IEEE-06]
	C05-Lack of security knowledge and skills	Developers are not security specialists; they are lacking security knowledge and training.	Lacking security education [S1-IEEE-06] Lacking knowledge and training [S1-IEEE-06] Lack of security knowledge [S1-IEEE-08, S1-GL-38,

			40] Developers are not security specialists [S1-GL-15] Unfamiliar with common security risks [S1-GL-18] The skills gap [S1-GL-37] Not enough security savvy [S1-GL-39]
	C06-Recruiting challenges	Challenges of recruiting DevOps/DevSecOps talents and engaging with security teams	Recruiting challenges [S1-GL-24] Understaffing InfoSec teams and engaging too late with the InfoSec team [S1-GL-18] Boundary between specialist and generalist [S1-IEEE-06]
	C07-Inconsistent security polices design	Organisations design inconsistent security polices when adopting DevSecOps	Inconsistent security polices design [S1-ACM-05, S1-IEEE-08]
	C08-Challenges of governance and leadership	Insufficient level of governance on DevSecOps adoption and lack of leadership's commitment.	Insufficient level of governance on DevSecOps adoption [S1-SC-08] Lack of clarity and transparency in strategy [Myrbakken and Colomo-Palacios' MLR] Lack of commitment of leadership and senior management [Myrbakken and Colomo-Palacios' MLR]
	C09-Lacking confidence	Low or no confidence in DevSecOps adoption.	Low or no confidence in DevSecOps [Myrbakken and Colomo-Palacios' MLR]
Process Capabilities	C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance	Integrate security practices into a fast-moving DevOps pipeline without slowing down, and without affecting development process and CI/CD system performance.	Integrate security practices into a fast-moving DevOps pipeline without slowing down [S1-SC-08] Implementing security in CI/CD [S1-GL-28] Rapid pace of change [S1-GL-29] Faster development process [S1-GL-28] Keep up with the pace of DevOps [S1-GL-30] DevOps velocity [S1-GL-37] Slow security testing [S1-GL-38, 40] Running current product and services in parallel to its transformation to DevSecOps [S1-SC-08] Tradeoff between security measures and CI system performance [S1-ACM-95] Interconnectedness of the DevOps process [S1-GL-28]
	C11-Using unsuitable metrics	Using unsuitable metrics to assess DevSecOps performance.	Using unsuitable metrics [S1-ACM-01, 05, S1-IEEE-08]
	C12-Compliance challenges	Challenge between compliance control and requirements.	Compliance requirements [S1-IEEE-07, 08, 11, S1-ACM-05, S1-GL-39]
	C13-Neglecting change control in security	Change control is always neglected in DevSecOps.	Neglecting change control in security [S1-IEEE-08]
	C14-Lack of standards	Lack of security standards and tool standards.	Lack of security standards [S1-IEEE-08] Lack of tool standards [S1-IEEE-06]
	C15-Ignoring processes and security essentials leading to	Technical and security debt might be caused by neglecting security essentials	Ignoring processes and security essentials leading to technical debt and security debt [S1-SC-08]

	technical and security debt	in the planning step.	
	C16-Poor visibility of security track record	The security track record is not visible.	Poor visibility of security track record [S1-GL-19]
	C17-Inadequate privileged credentials and access controls causing cyber attacks	Privileged credentials used in DevOps are targeted by cyber attackers, and the inadequate controls provide an opening for attack.	Inadequate controls provide an opening for attack [S1-GL-30] Privileged credentials used in DevOps are targeted by cyber attackers [S1-GL-17]
Technology	C18-Lack of mature tools for automation and security	Need for mature automation tools, e.g., automated testing tools, automated deployment tools, automated scans tools, automated monitoring tools, etc.	Lack of automated testing tools [S1-IEEE-06, 08, S1-ACM-05] Lack of integrated testing tools [S1-IEEE-08, S1-ACM-05] Wrong automated deployment tools [S1-IEEE-12, S1-ACM-01] Immature automated tools [S1-IEEE-08, 12, S1-ACM-01, 05] Need for automated testing [S1-IEEE-08, S1-ACM-05] Mismatched tools [S1-GL-15] Tool-centric approaches to secrets management create security gaps [S1-GL-17] Inefficient SAST tools [S1-GL-19] Manual pen-testing becomes a bottleneck [S1-GL-19] Threat modeling scalability issue [S1-IEEE-08, S1-ACM-05]
	C19-Complexity in managing different tools	Complexity and difficulties in managing different types of tools used in DevSecOps.	Complexity in managing different tools [Myrbakken and Colomo-Palacios' MLR]
	C20-Challenges of legacy system refactoring	Challenging to automate legacy system, if systems are not scalable or lack of cloud support.	Challenging to automate legacy system [S1-IEEE-06] Lack of cloud support [S1-GL-19] Systems are not scalable [S1-GL-19] Legacy infrastructure [S1-GL-24]
	C21-Use of cloud and serverless computing brings security complications	Use of Cloud, serverless computing, and open-source environments lead to compromise of critical information, configuration errors, compliance issues and security breaches.	Cloud security complications [S1-SC-25, 44, S1-IEEE-06, 16, 25, 39, S1-ACM-19, 52, 59, 66, S1-GL-24, 29, 38, 40] Attacks due to miss-configured cloud environments [S1-IEEE-33, 42] Security smells in Infrastructure as Code [S1-ACM-06, S1-IEEE-28, S1-SC-26] Security smells in serverless computing [S1-GL-28] Cloud and open-source environments lead to compromise of critical information, configuration errors, compliance issues and security breaches [S1-GL-20]
	C22-Containers and other tools come with their own risks	Container and other tools can often be the reason for security concerns.	Container and other tools can often be the reason for security concerns [S1-GL-20]

			Workload containerisation [S1-GL-29] Tools come with their own risks [S1-GL-30]
	C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth	Availability and reliability of infrastructure resources, tools, automation, and network bandwidth for shorter and frequent deployment cycle.	Availability and reliability of infrastructure resources, tools, automation, and network bandwidth for shorter and frequent deployment cycle [Myrbakken and Colomo-Palacios' MLR]
	C24-Continuous deployment chaos	Difficulties in responding continuous deployment chaos.	Continuous deployment chaos [S1-GL-19]
Business	C25-Challenges of cost control	High cost such as salaries for security experts, costs on new tools, trade-off between risk and cost.	High cost such as salaries for security experts, costs on new tools [S1-IEEE-04] Risk and cost battle [S1-IEEE-06]
	C26-Conflicts between security and business	Security and business objectives are implemented using conflicting approaches, e.g., velocity and security battle.	Security and business objectives are implemented using conflicting approaches [S1-ACM-64] Dilemma in selection of business processes in product and service delivery for transformation to DevSecOps [S1-SC-08]
	C27-Customer readiness for frequent releases	Customers are not ready for more frequent releases in DevSecOps adoption.	Customer readiness for applying frequent releases [Myrbakken and Colomo-Palacios' MLR]
	C28-Training users for using advanced tools	Users need to be properly trained when using advanced tools in DevSecOps.	Users need to be properly trained when using advanced tools [Myrbakken and Colomo-Palacios' MLR]

DevSecOps Practices			
Category	**Practice/Theme**	**Explanation**	**Codes [Papers contributed to the code]**
Organisation, People, and Culture	P01-Cultural shift to security	Cultural shift, change the security mindset and make security a priority.	Cultural shift [S1-GL-41] Change the security mindset [S1-GL-32] Make security a priority [S1-GL-32]
	P02-Improving collaboration, communication and cooperation	Enhanced collaboration, strong communication, and trusted relationships between development, operation and security.	Work collaboratively [S1-ACM-02] Enhanced collaboration [S1-ACM-02] Cross-departmental collaboration [S1-IEEE-04] Collaborating development, operation and security [S1-IEEE-04, 12] Close collaboration [S1-IEEE-12] Collaboration within and between different teams [S1-IEEE-12] Collaboration amongst different departments [S1-IEEE-12] Collaboration between Dev and Ops [S1-IEEE-12] Collaboration between Dev and Sec [S1-IEEE-12] Collaboration between Sec and Ops [S1-IEEE-12] Team collaboration [S1-IEEE-15] Strong collaboration [S1-IEEE-15] Strong communication [S1-IEEE-12]

			Close communication [S1-ACM-02, S1-IEEE-09] Communication of security requirements [S1-ACM-02] Virtual communication [S1-ACM-02] Face-to-face communication [S1-ACM-02] Physical communication [S1-ACM-02] Trust [S1-ACM-02, S1-IEEE-29] Trustworthy [S1-ACM-02] Trusted relationships [S1-ACM-02] Mutual trust [S1-ACM-02] Implicit trust [S1-ACM-02] Trust within the teams [S1-IEEE-29] Cross-functional collaboration [S1-GL-30] Foster collaboration [S1-GL-25] Open contribution and collaboration [S1-GL-24] Collaboration and integration [S1-GL-02] Communicate and collaborate [S1-GL-32] Improving empathy and cooperation [S1-GL-10] Reducing friction [S1-GL-10]
	P03-Shared and collective responsibility for security	Assign security responsibility to one person from DevOps team.	Shared responsibility for security [S1-ACM-02] Collective responsibility [S1-GL-02] Assign security responsibility to one person from DevOps team [S1-GL-28]
	P04-Shared knowledge	Share security knowledge and learn from each other.	Knowledge sharing [S1-ACM-02] Learn from each other [S1-GL-32] Shared threat intelligence [S1-GL-24]
	P05-Training, learning and education for security	Educate developers, cross-training, and security learning.	Training [S1-GL-06, 10, 32] Cross-training [S1-GL-35] Educate developers [S1-GL-25] Security learning [S1-GL-14]
	P06-Security champions	Organisations build security champions programs to create their positive security culture.	Security champions [S1-ACM-02, S1-GL-10]
	P07-Recruiting success	Recruit DevSecOps talents timely and successfully	Recruiting success [S1-GL-10]
	P08-Continuous feedback loop	Enhance the continuous feedback loops within and across teams.	Feedback loop [S1-ACM-15] Continuous feedback loops [S1-GL-09, 13, 15, 22, 35]
	P09-Be reactive and responsive	Everyone should be reactive and responsive for security.	Be reactive and responsive [S1-GL-32]
	P10-Shameless retrospectives	Conduct shameless retrospectives within and across teams.	Shameless retrospectives [S1-IEEE-09]
	P11-Impose security policies	Organisations impose security policies and governance.	Impose policy and governance [S1-GL-41] Implement security policies [S1-GL-30]
	P12-Commitment and agreement	Get commitment and agreement for DevSecOps adoption.	Commitment and agreement [S1-IEEE-29]

	P13-Enhance transparency	Enhance the transparency in security strategy.	Transparency [S1-IEEE-29, S1-SC-09]
	P14-Continuous improvement mindset	Build and keep continuous improvement mindset in DevSecOps.	Continuous improvement mindset [Sánchez-Gordón and Colomo-Palacios' SLR]
	P15-Leadership support	Supports from leadership and senior management when adopting DevSecOps	Leadership support [Sánchez-Gordón and Colomo-Palacios' SLR]
Process Capabilities	P16-Shifting security to the left (early)	Shifting security to the left (early), i.e., integrate security during the planning step.	Shifting security to the left [S1-IEEE-04, 24, 26, S1-SC-08, 11, S1-ACM-50, 81] Moving security to the left [S1-GL-08, 09, 13, 15, 18, 31, 35, 36] Integrate security during the planning phase [S1-GL-35] Take a proactive approach to security [S1-GL-17] Include security early [S1-GL-28]
	P17-Security-by-Design	Embedding security into a development project when design it.	Security by design [S1-SC-07, 08, 18, 20, 22, S1-IEEE-16, 29, 30, 36, S1-ACM-45, 69, S1-GL-31]
	P18-Increase the visibility	Enhance the visibility for security track records.	Increase the visibility [S1-SC-09] Enhance visibility [S1-GL-41]
	P19-Good documentation, logging and reporting	Ensure good documentation, logging, and reporting.	Good documentation and logging [S1-IEEE-15] Better reporting [S1-GL-02, 19]
	P20-Compliance control	Good compliance control, e.g., identify compliance requirements beforehand and bridge the divide between compliance and development.	Compliance control [S1-IEEE-11, S1-SC-27, S1-GL-10, 24] Identify compliance requirements beforehand [S1-GL-28] Bridging the divide between compliance and development [S1-GL-02]
	P21-Risk management	Risk management, including risk assessment, risk treatment and risk control.	Risk management (including risk assessment, risk treatment and risk control) [S1-SC-11, 18, 20, 22, 26, 40, 41, S1-ACM-03, S1-IEEE-34]
	P22-Vulnerability and incident management	Increase vulnerability management and incident management.	Vulnerability and incident management [S1-GL-14] Incident management [S1-GL-08, 10] Vulnerability management [S1-GL-23, 30]
	P23-Privilege management	Increase privileged access management, such as least privilege controls.	Least privilege controls [S1-IEEE-33] Privileged access management [S1-GL-30] Secure access via secrets management [S1-GL-41]
	P24-Configuration management	A systematic process for maintaining the desired state of a system by controlling and tracking changes to its components and their configurations throughout its lifecycle	Configuration management [S1-GL-10]
	P25-Patch management	A process that involves identifying, acquiring, testing, and deploying software updates/patches, to fix vulnerabilities, improve performance, and enhance security	CI/CD for patching management [S1-GL-10]
	P26-Define metrics	Define and adopt suitable metrics to measure the performance of DevSecOps.	Define metrics [S1-GL-06, 19] Measurement [S1-GL-02]
	P27-Software process	Ensure software process maturity, e.g.,	Software process maturity [S1-SC-32]

	maturity	Building Security In Maturity Model (BSIMM) model.	Building Security In Maturity Model (BSIMM) model [S1-ACM-01]
	P28-Define security requirements	Clearly define security requirements during planning.	Define security requirements [S1-GL-06] Security requirements and design [S1-GL-14]
	P29-Security review and evaluation	Proactive security review and evaluation, to detect existing security flaws.	Security reviews [S1-GL-18] Security evaluation [S1-GL-14] Proactive security assessments [S1-GL-10] Detect existing security flaws [S1-SC-09] Make sure the basics of host and network security are in place [S1-SC-09] Host hardening [S1-GL-10] Application-level assessment [S1-GL-10] Operational controls validation and improvement [S1-GL-14]
	P30-Keep credentials safe	Make sure all the credentials are secured.	Keep credentials safe [S1-GL-06]
	P31-Common weaknesses enumeration	Common weaknesses enumeration (CWE) is a community-developed list of software and hardware weakness and vulnerabilities.	Common weaknesses enumeration [S1-GL-08]
	P32-Hybrid life cycles with data-security focus	Combining data security and software development life cycles.	Combining data security and software development life cycles [Rajapakse's SLR]
Technology	P33-Automate tools and security process	Automate the security process and use automated tools as much as possible, e.g., automated testing, automating code review, automating scans, automated monitoring.	Automation [S1-ACM-01, 09, 49, 71, 72, 81, 95, S1-IEEE-06, 07, 09, 10, 12, 13, 15, 20, 21, 26, 38, 41, 54, 57, S1-SC-08, 09, 11, 17, 18, 20, 22, 26, 27, 32, 40, S1-GL-02, 04, 06] Automated/automating test/testing [S1-ACM-01, 09, 49, 81, 95, S1-IEEE-06, 07, 09, 10, 12, 15, 21, 26, 38, 41, 54, 57, S1-SC-08, 09, 11, 17, 18, 22, 26, 27, S1-GL-08,11,13,15, 35] Automated monitoring [S1-ACM-01, 71, 72, 81, S1-IEEE-07, 12, 13, 15, 21, 26, 38, S1-SC-08, 09, 18, 20, 26, 40] Automated/automating scans [S1-IEEE-07, S1-SC-32] Automated/automating code review [S1-IEEE-07, 12, S1-GL-23] Automate as much as possible [S1-GL-25, 28] Automate protection of business logic flaws [S1-GL-09] Automate tools and security processes [S1-GL-17, 30] Use automated security tools [S1-GL-41]
	P34-Security-as-Code	Integrate security into DevOps tools and implement security practices, tests and policies in SDLC.	Security as code [S1-SC-08, 09, 18, S1-IEEE-06, S1-GL-32]
	35-Threat modelling	Proactively identify and prioritise potential threats to a system.	Threat modeling/analysis [S1-IEEE-02, 04, 07, 11, 30, 36, 39, 61, 71, S1-SC-26, S1-GL-06, 10, 14, 25, 28]

	P36-Continuous monitoring	24 x 7 continuous and proactive monitoring	Continuous monitoring [S1-IEEE-07, 12, 13, 15, 21, 26, 38, S1-SC-08, 09, 18, 20, 26, 40, S1-ACM-01, 15, 71, 72, 81, S1-GL-02, 06, 25, 31] 24 x 7 proactive monitoring [S1-GL-24]
	P37-Secure coding	Source code repository and scanning, build preapproved code, conduct code dependency checks regularly.	Source code repository and scanning [S1-GL-10] Secure coding [S1-GL-10, 14, 28] Build preapproved code [S1-GL-18] Conduct code dependency checks regularly [S1-GL-25]
	P38-Advanced malware detection	Advanced malware detection employs machine learning and deep learning.	Advanced malware detection employs machine learning and deep learning [S1-SC-32]
	P39-Cloud security	Verify cloud infrastructure, such as use MUSA Security DevOps framework.	Verify cloud infrastructure [S1-GL-28] MUSA Security DevOps framework [S1-ACM-52] MUSA DevOps framework for security in multi-cloud applications [S1-IEEE-16, 40]
	P40-Container security	Ensure container/containerisation security, version control, metadata and orchestration, e.g., run container as non-root users, use the latest version of image, conduct deep scanning of container image.	Container/Containerisation security [S1-ACM-52, S1-IEEE-55, S1-GL-28, 41] Run container as non-root users [S1-IEEE-55, S1-SC-09, 34] Use the latest version of image [S1-SC-42] Conduct deep scanning of container image [S1-IEEE-04] Enhance security of Docker [S1-IEEE-31, S1-GL-10] Security practices in Kubernetes [S1-IEEE-18, S1-GL-10] Version control, metadata and orchestration [S1-GL-10]
	P41-Sensitive information scan	Identifies sensitive data within a system, like a codebase, database, or cloud storage, to protect it from unauthorised access and potential breaches.	Sensitive information scan [S1-GL-23]
	P42-Software composition analysis	Software composition analysis (SCA) is an automated process that identifies the open-source software in a codebase.	Software composition analysis [S1-GL-06, 23]
	P43-Red team security drills	Conduct red and blue team exploit testing.	Red team security drills [S1-IEEE-04] Red and blue team exploit testing [S1-GL-24]
	P44-Fault injection (chaos engineering)	Deliberately introduce errors to a system to ensure it can withstand and recover from error conditions.	Fault injection (chaos engineering) [S1-IEEE-13]
	P45-Runtime Application Self-Protection (RASP)	Runtime instrumentation to detect and block computer attack.	Runtime Application Self-Protection (RASP) [S1-SC-32, S1-GL-02, 08, 25]
	P46-Static Application Security Testing (SAST)	Analyse source code to find vulnerabilities. It scans an application before the code is compiled (white box testing).	Static Application Security Testing (SAST) [S1-GL-02, 08, 23, 25]
	P47-Dynamic Application	Analyse an application in runtime, through	Dynamic Application Security Testing (DAST) [S1-

	Security Testing (DAST)	the front-end to find vulnerabilities through simulated attacks. It does not require access to the source code (black box testing).	IEEE-10, S1-GL-02, 08, 23, 25]
	P48-Interactive Application Security Testing (IAST)	Test an application for vulnerabilities in execution, while the app is actually being used.	Interactive Application Security Testing (IAST) [S1-IEEE-15, S1-GL-02, 08, 19, 25]
	P49-Immutable-as-Code	Use of code to increase immutability.	Immutable-as-code ensures the immutability of infrastructure and avoids accidental configuration drifts [S1-IEEE-33]
	P50-Policy-as-Code	Use of code to define and manage rules and conditions.	Policy-as-Code is an attempt to code the policy itself [S1-IEEE-33, S1-GL-17]
	P51-Design-as-Code	Define design elements using code, which is then used to programmatically generate visual assets.	Design-as-code: CAIRIS (Computer Aided Integration of Requirements and Information Security) model [S1-IEEE-36]
	P52-Compliance-as-Code	Define compliance requirements in such a way that you can automate it and write tests for it.	Compliance as code [S1-GL-23]
	P53-Adopting DevSecOps in microservices-based applications	Practices of adopting DevSecOps in microservices-based applications.	Adopting DevSecOps in microservices-based applications [S1-IEEE-17, 43, 52, 57, 84, 86, S1-SC-15, 36]
	P54-Integrate security issues within general bug tracker	Integrate security issues within the general bug tracker when adopting DevSecOps.	Integrate security issues within your general bug tracker [S1-GL-19]
	P55-Big data and behavioral analytic techniques	Obtain fast feedback from end users and predictive analytic for trends in user behaviors.	Obtain fast feedback from end users and predictive analytic for trends in user behaviors [Rajapakse's SLR]
Business	P56-Consumable security services with APIs	Consume relevant security services when consume APIs.	Consumable security services with APIs [S1-GL-24]
	P57-Separation of duties	Divide tasks among individuals or teams to prevent errors, fraud, and other financial misconduct.	Separation of duties [S1-GL-14, 17]
	P58-Business-driven security	Proactively define, establish and manage security posture for business process.	Business-driven security [S1-GL-24]
	P59-Linear scalability and affordable cost	Trade-off between scalability and cost when adopting DevSecOps.	Linear scalability and affordable cost [S1-GL-19]
	P60-Availability and business continuity management	Ensure the availability and business continuity for DevSecOps adoption.	Availability and business continuity management [S1-GL-14]

DevSecOps Tools		
Category	**Tools Function/Theme**	**Tools/Codes[Papers contributed to the code]**
Technology	T01-Automation tools	Chef [S1-IEEE-07, S1-SC-12, 20, 26], Jenkins [S1-SC-12], Ansible [S1-SC-20, S1-GL-04], Puppet [S1-SC-20], Gauntlt [S1-IEEE-06], SaltStack [S1-SC-01, 20]

	T02-Automated code review tools	Veracode Greenlight [S1-SC-01], PMD [S1-GL-23], DevSkim [S1-GL-23], FindSecBugs [S1-GL-23]
	T03-Threat modeling tools	IriusRisk [S1-SC-01], Microsoft threat modeling tool [S1-IEEE-39]
	T04-Containerisation tools	Docker [S1-SC-09, 18, 20, 29, 34, 42, 45, 48, S1-ACM-95, 99, S1-IEEE-31, 55, S1-GL-03, 10], Kubernetes [S1-ACM-52, 76, 89, S1-SC-20, 29, S1-IEEE-18, S1-GL-03, 10]
	T05-Container security tools	Twistlock [S1-GL-42], Notary [S1-GL-42], Aqua Security [S1-GL-42]
	T06-Cloud security tools	Terraform [S1-SC-12, 20, S1-IEEE-33], AppScan on Cloud [S1-GL-42], AWS Security service [S1-GL-42], ThreatModeler Cloud Edition [S1-GL-42], Trend Micro Cloud One [S1-GL-42]
	T07-Sensitive information scanning tools	TruffleHog [S1-GL-23], GitSecrets [S1-GL-23], Talisman [S1-GL-23]
	T08-SAST tools	Kiuwan [S1-SC-01], Flawfinder [S1-GL-23], Graudit [S1-GL-23], Bandit [S1-GL-23], Spotbugs [S1-GL-23], SonarQube [S1-GL-23, 42]
	T09-DAST tools	OWASP ZAP [S1-GL-23], BDD Security [S1-GL-23], Arachini [S1-GL-23], Nikto [S1-GL-23], Radamsa [S1-GL-23], FuzzDB [S1-GL-23], Fortify Webinspect [S1-GL-42]
	T10-RAST tool	Fortify Application Defender [S1-GL-42]
	T11-Advanced malware detection tool	CodeAI [S1-SC-01]
	T12-Software composition analysis tools	Retire.js [S1-GL-23], OSSAudit [S1-GL-23], OWASP Dependency-Check [S1-GL-23]
	T13-Compliance-as-Code tools	nspec [S1-GL-23], ServerSpec [S1-GL-23], OpenSCAP [S1-GL-23]
	T14-Vulnerability management tools	Defect Dojo [S1-GL-23], ArcherySec [S1-GL-23], Snyk [S1-GL-10, 21], HackerOne [S1-GL-21], Claire [S1-GL-21], Stethoscope [S1-GL-21], Rapid7 Nexpose [S1-GL-21]
	T15-DevOps performance measuring tool	SAFe DevOps Health Radar [S1-GL-01]
	T16-Monitoring and alerting tools	Suricata [S1-GL-21], NewRelic [S1-GL-42], Nagios Icinga, Graphite, Ganglia, Cacti, Pager Duty, Sensu, Boundry, Pingdom [Mohan and Othmane's mapping study]
	T17-Cyber security tools	Tripwire, Snort [Mohan and Othmane's mapping study]
	T18-Logging tools	PaperTrail, Logstash, Loggly, Splunk, SumoLogic [Mohan and Othmane's mapping study]

DevSecOps Metrics		
Category	**Metric/Theme**	**Explanation/codes [Papers contributed to the code]**
Organisation, People, and Culture	M01-Security-trained rate	The ratio of developers that have gone through security-training in the team. [S1-IEEE-06] *Measuring: The number of developers that have gone through security-training divided by the total number of developers in the team. Higher rate means better training.* *Goal: Know the number and the level of developers with good security mindset, knowledge and skills.*
Process Capabilities	M02-Top vulnerability	Number of mistakes in different security categories [S1-IEEE-06] OWASP top 10 [S1-IEEE-06]

		Top vulnerability types and recurring bugs [S1-GL-43] *Measuring: Count the number of different types of mistakes and keep track of most recurring vulnerabilities.* *Goal: Help planning training provided to developers accordingly and capacitate them with knowledge to handle and mitigate returning vulnerabilities.*
	M03-Time spent correcting mistakes in each category	Time spent correcting mistakes in each category. [S1-IEEE-06] *Measuring: Count the time spent correcting mistakes different vulnerability types. The shorter, the easier.* *Goal: Assess the difficulties of addressing different vulnerability types.*
	M04-Security review performance	Whether features undergo a security review. [S1-GL-18] *Measuring: The percentage of features that undergo security review early in the design process. This percentage should go up over time.* *Goal: Know the current state and progress of security reviews.* Whether security review slows down the development cycle. *Measuring: How much time the reviews add to the development process. The time that security reviews take should go down until it reaches an agreed-to minimum.* *Goal: Assess the efficiency of security reviews.* How well security is integrated into the delivery lifecycle. *Measuring: Measure the number of security reviews captured at each of the stages of the software development lifecycle. This number should go up until it reaches a value that suggests that InfoSec is fully integrated into the lifecycle.* *Goal: Know the degree of InfoSec team's involvement in each step of the software delivery lifecycle.*
	M05-SLA performance	How well Service Level Agreements (SLAs) perform. [S1-GL-43] *Measuring: Set up service level agreements (SLAs) based on criticality and tracking the SLA performance religiously* *Goal: Assess the SLA performance.*
	M06-Critical risk profiling	Critical risk profiling–the relation between issue criticality and the value of that vulnerability to possible attackers. [S1-GL-43] *Measuring: Vulnerability should be associated with a score for a criticality and another that defines the value of that vulnerability to attackers. Vulnerabilities that have high scores in both criticality and value should be addressed first. The scores are expected being as small as possible.* *Goal: Prioritise the order of addressing issues.*
	M07-Point of risk per device	Risks and vulnerabilities on per device. [Prates' MLR] *Measuring: Identify and keep track of un-patched vulnerabilities per server. The number of vulnerabilities should tend to zero.* *Goal: Prioritise vulnerabilities according to their criticality giving special attention to the ones that are most exposed to attack from the internet.*
	M08-Number of continuous delivery cycles per month	How many successful continuous delivery cycles per month. [Prates' MLR] *Measuring: Count the number of attempts to deploy versus the number of successful attempts. A positive value is to have the highest number of successful attempts.* *Goal: Measure how quickly code changes can be deployed to production.*
Technology	M09-Number of adversaries per application	Number of adversaries per application, this is associated with the practice of Threat Modelling and Risk Analysis. [Prates' MLR] *Measuring: Team exercise where the objective is to think how many adversaries they think an application as and register those findings.* *Goal: Identify the applications inside an organisation that are more exposed to possible attacks and*

		prepare accordingly.
	M10-Adversary return rate	How often an adversary will use the same strategy and procedures. [S1-GL-43] *Measuring: Measure is done by counting the number of times adversaries use the same attacking strategy and compiling into a ranking that visible for every team member. Ideal is to have a plan to handle each attacking strategy.* *Goal: Define appropriate training and preparing to better handle these known attacks.*
	M11-Defect density	Number of confirmed defects detected in software/component during a defined period of development/operation divided by the size of the software/component. [S1-GL-43] *Measuring: Defect density is measured by dividing the total number of confirmed defects by the total line of codes of all the modules in the new release. Ideal is to have the lowest density value possible.* *Goal: Helps Sec team and developers negotiate reasonable goals to reduce defect density over time.*
	M12-Defect burn rate	How quickly the team is addressing defects. [S1-GL-43] *Measuring: Take the total number of defects found in development and divided it by the sum of defects found in development and production and multiplied by 100. The rate is higher, the team is more effective.* *Goal: Measure Dev team productivity solving defects.*
	M13-Penetration test pass rate	Systems that are affected by internal and external penetration testing. [S1-IEEE-06] *Measuring: The degree of system that passed authorised and simulated cyberattacks.* *Goal: evaluate the security of the system in a simulated scenario.*
	M14-Security test pass rate	The rate of failed-versus-passed security test. [S1-IEEE-57] *Measuring: The ratio of failed-versus-passed static security source code scans in a given time period.* *Goal: Identify security vulnerabilities in the build stage.*
	M15-Code scanning detection rate	The rate of successful code scanning detection. [S1-IEEE-57] *Measuring: Count the number of security scans that come back with a problem in a given timeframe or given process phase, as well as the number of problems. This rate should decrease with time or with movement from one stage to the next.* *Goal: Improvements in this metric over time can increase confidence in the safety and security of the product.*
	M16-Whether automated testing covers security requirements	Whether automated testing covers security requirements. [S1-GL-18] *Measuring: As InfoSec gains greater input into the testing process, the number or percentage of security requirements that are included in the automated testing process. This percentage should go up over time.* *Goal: Know the degree of InfoSec team's involvement in writing automated tests.*
	M17-Use of preapproved libraries, packages, tool chains, and processes	Use of preapproved libraries, packages, tool chains, and processes. [S1-GL-18] *Measuring: Initially, measure whether InfoSec is engaged in tools development. As work progresses, the number of InfoSec approved libraries, packages, and tool chains that are available, or the number of these resources that are used by the development and operations teams. Engagement should increase over time until the organisation agrees that InfoSec oversight of tools is at the correct level. Similarly, the percentage or number of preapproved tools in use should increase until the team uses all the tools that InfoSec has created or approved.* *Goal: Know the degree of InfoSec team's engagement in tools development and the usage of preapproved libraries, packages, tool chains.*
	M18-Use of SAFe DevOps Health Radar	Use of DevOps measuring tool SAFe DevOps Health Radar. [S1-GL-01] *Measuring: Use SAFe DevOps Health Radar to measure DevOps performance, by assessing the maturity of four aspects and 16 activities of the CI/CD pipeline.* *Goal: know the maturity of DevOps.*

	M19-Number of issues during red teaming drills	How many issues identified during red teaming drills. [Prates' MLR] *Measuring: Count the number of defects found and fixed by the Red Team.* *Goal: Measure the effectiveness of Red Team.*
Business	M20-Business metrics	Business related metrics, e.g., Revenue, Key performance indicators (KPI). [Myrbakken and Colomo-Palacios' MLR] *Measuring: Define suitable DevOps KPIs for the organisation and assess the revenue accurately.* *Goal: Know the current state in business views, and find out how to improve it.*

Other terminologies relating to MLR findings and CPTM model	
Five Aspects of DevSecOps Research / Four Elements of the Model	
Definitions	The Definitions for the term 'DevSecOps' and equivalent terms. "Meanings", "Perceptions" and "Concepts" were categorised as fitting the ''Definitions'' aspect. "Characteristics" and "Benefits" were often mentioned in definitions, were therefore considered as the codes or themes under the "Definitions" aspect.
Challenges	The obstacles and uphill tasks encountered when adopting DevSecOps require ongoing efforts to overcome. "Problems", ''Issues'' and "Concerns" were categorised as fitting the ''Challenges'' aspect.
Practices	DevOps and security activities suited for DevSecOps. "Activities'', "Approaches'', "Solutions" and ''Strategies'' were categorised as fitting the ''Practices'' aspect.
Tools/Technologies	Specific tools and technical approaches that are used for DevSecOps practices. They could be in a subset of DevSecOps practices, especially the technology-related practices.
Metrics/Measurement	The means to track progress, facilitate decision-making, and improve performance of DevSecOps practices by measuring implementation, effectiveness, efficiency, and impact.
Four Categories of Themes	
Organisation, People and Culture (OPC)	The category that includes themes relating to organisational structure, people management, and cultural strategies, e.g., breaking silos, collaboration, communication, sharing, training, recruiting, etc.
Process Capabilities (PC)	The category that includes themes relating to the capabilities of DevSecOps process, e.g., integration of security, security-left, continuous activities, risk management, faster lifecycle, etc.
Technology	The category that includes themes relating to technological approaches and software and hardware tools, e.g., automation, cloud, containerisation, testing techniques and tools.
Business	The category that includes themes relating to business benefits, customers, quality of product and service, e.g., increasing value, higher quality, fewer impacts to users, etc. The reason for adding this category was that the MLR, especially the GL results, showed a business perspective on DevSecOps.
Ten Phases of the DevSecOps Model by Gartner (MacDonald and Head, 2016).	
Plan	The phase to set project objectives, identify security requirements, plan security measures, define metrics and policies, prepare organisations/teams, select technologies/tools, and develop budgets.
Create	The phase to start executing the plan, prepare security practices, and set up security tools.
Verify	The phase to conduct security practices by using appropriate (automated) tools and technologies, such as security tests (SAST, DAST, IAST) and software composition analysis (SCA).

Preproduction	The phase to include further security tests, such as chaos engineering and red team drilling.
Release	The phase to sign the software and get it ready to be released and build it into the production environment, by reviewing configuration, infrastructure, network bandwidth, compliance, etc.
Prevent	The phase to protect the runtime environment architecture.
Detect	The phase to continuously monitor and scan the runtime environment architecture.
Respond	The phase to address the vulnerabilities detected in the previous phase.
Predict	The phase to analyse the vulnerabilities to identify the causes.
Adapt	The phase to improve security processes and re-plan the DevSecOps lifecycle, based on the lessons learned from the previous phases.

Appendix F. Research Outputs from Delphi-AHP Study

The research outputs of three iterations of the Delphi-AHP study are available in an open repository at zenodo.org: https://doi.org/10.5281/zenodo.16932278, including:

- Round One – Results of the evaluation of challenges (including Dissent Analysis)
- Round Two – Results of the evaluation of revised challenges and practices (including Dissent Analysis)
- Round Three – Results of the evaluation of revised practices and metrics (including Dissent Analysis)
- Final results of the prioritisation of DevSecOps challenges, practices, metrics, and tools
- List of DevSecOps tools
- The DevSecOps CPTM Model (Version 2.0)