The DevSecOps Journey: Unraveling Crucial Skillsets for Software Engineers

Germán Dugarte Peña

Email: gdugarte@inf.uc3m.es

Contents

Abstract

DevSecOps involves an agile operating structure that incorporates aspects of development, operation, and security in a fluid, functional, and efficient manner. However, there is still much work to be done in transitioning from idealized designs to real industry practices, driven by individuals equipped with the necessary knowledge and skills. This report examines the main standards that provide guidance on the competencies required by professionals and organizations in technological environments. Drawing from this knowledge, we propose a set of skills, both personal and organizational, that are essential for the successful establishment and implementation of DevSecOps. To support this objective, we conduct a process of correspondence identification that compares critical sectors of DevSecOps with the skills defined by industry standards. It is crucial for DevSecOps to evolve into complex systems that are sustainable and viable over time, while ensuring proper operation, development, and security, so the Viable System Model is used as the lens to identify the critical skills that may guarantee viability, in cybernetic terms, of DevSecOps in time. Therefore, DevSecOps must be shaped as intelligent, resilient, and adaptable systems capable of navigating change. The report also refines the characteristics of DevSecOps and its practitioners that contribute to defining this DevSecOps model.

Figures

Tables

Acknowledgements

The author would like to thank the advisors for their comments. This work is the result of an international collaboration resulting from a stay funded by the Spanish Margarita Salas scholarship program in which German-Lenin Dugarte-Peña, the first author of this article, visited Østfold University College, in Halden, Norway, during the academic year 2022-2023. Figure 5 shows the logo of the Ministry of universities, in order to identify the origin of public nature of the grant associated with the Margarita Salas scholarship, in accordance with the provisions of Royal Decree 1465/1999, of September 17, establishing the criteria for institutional image and regulating the documentary production and printed material of the General State Administration, as well as the provisions of the Resolution of July 10, 2018, of the Secretary of State for Public Function, updating the Institutional Image Manual.

FIGURE 1. Institutional logo of the Spanish Ministry of Universities

Abbreviations

TABLE 1. Abbreviations

Full form	Abbreviation / Acronym
Acceptance testing	BPTS
Animation development	ADEV
Application support	ASUP
Asset management	ASMG
Audit	AUDT
Availability management	AVMT
Benefits management	BENM
Business administration	ADMN
Business intelligence	BINT
Business modelling	BSMO
Business process improvement	BPRE
Business situation analysis	BUSA
Capacity management	CPMG
Certification scheme operation	CSOP
Change control	CHMG
Competency assessment	LEDA
Configuration management	CFMG
Consultancy	CNSL
Content authoring	INCA
Content publishing	ICPM
Continuity management	COPL
Contract management	ITCM
Customer service support	CSMG
Data engineering	DENG
Data management	DATM
Data modelling and design	DTAN
Data science	DATS
Data visualisation	VISL
Database administration	DBAD
Database design	DBDS
Demand management	DEMM
DevOps	Development-Operations
DevSecOps	Development-Security-Operations
Digital forensics	DGFS
Emerging technology monitoring	EMRG
Employee experience	EEXP
Enterprise and business architecture	STPL
Facilities management	DCMA
Feasibility assessment	FEAS
Financial management	FMIT

Full form	Abbreviation / Acronym
Governance	GOVN
Hardware design	HWDE
High-performance computing	HPCC
Incident management	USUP
Information assurance	INAS
Information management	IRMG
Information security	SCTY
Information systems coordination	ISCO
Innovation	INOV
Investment appraisal	INVA
IT infrastructure	ITOP
Knowledge management	KNOW
Learning and development management	ETMG
Learning delivery	ETDL
Learning design and development	TMCR
Machine learning	MLNG
Marketing	MKTG
Measurement	MEAS
Methods and tools	METL
Network design	NTDS
Network support	NTAS
Numerical analysis	NUAN
Organisation design and implementation	ORDI
Organisational capability development	OCDV
Organisational change management	CIPM
Organisational facilitation	OFCL
Penetration testing	PENT
Performance management	PEMT
Personal data protection	PEDP
Portfolio management	POMG
Portfolio, programme and project support	PROF
Problem management	PBMG
Product management	PROD
Professional development	PDSV
Programme management	PGMG
Programming/software development	PROG
Project management	PRMG
Quality assurance	QUAS
Quality management	QUMG
Radio frequency engineering	RFEN
Real-time/embedded systems development	RESD
Release and deployment	RELM
Requirements definition and management	REQM

Full form	Abbreviation / Acronym
Research	RSCH
Resourcing	RESC
Risk management	BURM
Safety assessment	SFAS
Safety engineering	SFEN
Sales support	SSUP
Scientific modelling	SCMO
Security operations	SCAD
Selling	SALE
Service acceptance	SEAC
Service catalogue management	SCMG
Service level management	SLMO
Software configuration	PORT
Software design	SWDN
Solution architecture	ARCH
Sourcing	SORC
Specialist advice	TECH
Stakeholder relationship management	RLMT
Storage management	STMG
Strategic planning	ITSP
Subject formation	SUBF
Supplier management	SUPP
Sustainability	SUST
System software	SYSP
Systems and software life cycle engineering	SLEN
Systems design	DESN
Systems development management	DLMG
Systems installation and removal	HSIN
Systems integration and build	SINT
Teaching	TEAC
Technology service management	ITMG
Testing	TEST
Threat intelligence	THIN
User experience analysis	UNAN
User experience design	HCEV
User experience evaluation	USEV
User research	URCH
Vulnerability assessment	VUAS
Vulnerability research	VURE
Viable System Model	VSM
Workforce planning	WFPL

1. Introduction

Technological companies in the software industry have for long time been working on implementing practices and organizational modes that allow them to optimize the use of resources while making a greater impact on the achievement of their goals. DevOps, understood as "*a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions while guaranteeing their correctness and reliability*" (Akbar et al., 2022; Leite et al., 2019), became the leading paradigm for implementing agility as an organizational standard while ensuring that operations were done correctly and that developers were able to generate their deliverables in an effective manner. This is something that has been widely accepted both in industry and academia, which is why it is among the most significant change drivers and inputs in the ongoing update of the formal Software Engineering Body of Knowledge (SWEBOK)(IEEE-CS Professional & Educational Activities Board - SWEBOK Evolution Team, 2023). However, "*the time scale of standardisation bodies and academia is far slower than the one of the digital economy sector*" (Capozucca & Guelfi, 2020), bringing up to discussion the importance of counting on an updated framework for understanding the balance between the profession requirements and the real skills of professionals involved in the sector. As DevOps began to gain traction years ago, a compelling need emerged and became increasingly important: the integration of security with development and operations, giving rise to what later came to be known as DevSecOps or SecDevOps (terms understood indistinctively in this report). DevSecOps is understood as "*a model on integrating the software development and operational process considering security activities: requirements, design, coding, testing, delivery, deployment and incident response*" (Yasar, 2020).

1.1 Context

The most recent evolutionary efforts in the software industry in this regard go towards establishing the need for DevSecOps to be properly integrated and for the overall strategy and related actions to aim at maintaining agility while development, operations and security coexist, mutually nurturing each other and sharing their common goal. In this regard, the top 5 challenges for DevSecOps have recently been stated as (Yasar & Yankel, 2023):

- Lack of security assurance at the business and project levels.
- Organizational barriers related to collaboration, tooling, and culture.
- Impact to quality because security is not a priority while systems are getting more complex.
- Lack of security skills for developers, business stakeholders, and auditors.
- Insufficient security guidance due to lack of resources, standards, and data.

This list of the top 5 challenges for DevSecOps reveals the importance of the need to count on the skills to address problems related to aspects such as project management, organizational behavior, collaboration and integration of teams, measurement of quality, complexity, and security skills in team members at all levels. Conversely, what is the skillset that allows a directorate board to successfully implement DevSecOps with the guarantee of accomplishing the referred challenges?

In this report, a set of skills is proposed based on the identification of the skills required in each critical sector of DevSecOps. For this purpose, the SFIA standard (SFIA Foundation, 2021), known as the broadest reference framework of competencies required in the IT sector, is taken as a reference, and the coverage that this standard can have on the DevSecOps areas is assessed. Additionally, a series of desirable characteristics of DevSecOps are introduced based on Stafford Beer's Viable Systems Model, which from an organizational point of view could allow DevSecOps to be viable over time, guaranteeing operability, coordination, attention to changes in the environment, intelligence and strategic vision(Espejo & Reyes, 2011; Johnson & Leydesdorff, 2013).

An interesting nuance to study in this regard is the role that humans play in the implementation of DevSecOps. The human factor is fundamental, and a correct implementation of DevSecOps cannot be conceived without taking into account that those behind its success or failure are professionals, generally from the software industry, each with an important part of interest in the functioning of DevSecOps as a whole.

In this report we conduct an analysis of the skills demanded from professionals in the DevSecOps industry to ensure its long-term success and sustainability. We start from understanding the most well-known standards, such as the SFIA (SFIA Foundation, 2021) as a generic but very complete reference of skills, the P-CMM (Curtis et al., 2001) as a reference of maturity in regard to people skills, the US IT Competence Model (*Information Technology Competency Model*, 2021), and the Software Engineering Competence Model (SWECOM)(Burgess et al., 2014), which also considers other standards such as the CC2020 (Ormond, 2021).

To proceed with the study, first we list the fundamental activities of DevSecOps that must be considered to guarantee their effective deployment and functioning, we introduce the most relevant standards, and then propose a mapping between the skills from the most complete standard and the identified fundamental activities of DevSecOps. This mapping will allow to identify which are the critical capabilities of industry professionals that affect in some way the organizational performance of DevSecOps. Also, we present a perspective on the organizational capabilities for the systemic deployment of DevSecOps.

1.2 Understanding DevSecOps

DevSecOps is a methodology that integrates security practices into the entire software development life cycle (SDLC). It involves a collaborative approach between the development, security, and operations teams to ensure that software is developed securely and delivered quickly.

The areas of DevSecOps can be broadly categorized into three main categories:

Development: This includes the processes involved in designing, coding, and testing software. In DevSecOps, development teams need to prioritize security from the beginning of the SDLC. They should incorporate security practices, such as static code analysis, vulnerability scanning, and penetration testing, to identify and fix security issues early on in the development process.

Security: The security team's role in DevSecOps is to ensure that the software is secure throughout the SDLC. This includes assessing the risk of vulnerabilities, ensuring compliance with security standards and regulations, and implementing security controls to mitigate risks. The security team should also provide guidance to the development and operations teams on secure coding practices, threat modeling, and secure configurations.

Operations: In DevSecOps, the operations team's focus is on ensuring that the software is deployed and operated securely. This includes managing the infrastructure, network, and application security controls, as well as monitoring the system for security incidents. The operations team should also ensure that security controls are integrated into the continuous delivery pipeline, and that security incidents are detected and responded to in a timely manner.

Overall, the three areas of DevSecOps work together to create a secure and efficient software development process. By integrating security into the SDLC, organizations can improve the quality and security of their software, reduce the risk of security incidents, and increase the speed of software delivery.

DevSecOps is often seen as a natural evolution of the DevOps approach, as it addresses a key weakness of traditional DevOps workflows: a lack of focus on security. By incorporating security considerations into the DevOps process, organizations can ensure that their software is secure, reliable, and resilient, and can respond quickly to security threats and incidents.

DevSecOps is a software development and delivery approach that emphasizes collaboration and automation. It involves several processes that work together to help organizations deliver software faster and with higher quality and reliability. Here are the key processes involved in the safe operation of DevOps, or DevSecOps that we will explore in this work:

- Continuous Integration (CI)
- Continuous Deployment (CD)
- Continuous Delivery (CDel)
- Infrastructure as Code (IaC)
- Configuration Management
- Monitoring and Logging
- Collaboration and Communication
- Security testing Load testing
- Performance testing

DevOps is a software development and delivery approach that emphasizes collaboration and automation. It involves several processes that work together to help organizations deliver software faster and with higher quality. Here are some of the key processes involved in DevOps:

Continuous Integration (CI): This process involves automating the process of building, testing, and integrating code changes. This allows developers to detect and resolve conflicts and bugs early in the development process.

Continuous Deployment (CD): This process involves automating the process of deploying code changes to production. CD helps ensure that software changes are tested and deployed quickly and safely.

Continuous Delivery (CDel): This is an extension of CD that involves automating the entire software delivery pipeline, from code check-in to deployment to production. CDel helps organizations achieve faster and more reliable software delivery.

Infrastructure as Code (IaC): This process involves using code to manage and provision infrastructure. IaC helps organizations automate the process of setting up and managing servers, networks, and other infrastructure components.

Configuration Management: This process involves automating the process of configuring and managing software and infrastructure components. Configuration management helps organizations maintain consistency and control over their software and infrastructure.

Monitoring and Logging: This process involves collecting and analyzing data from software and infrastructure components to detect and resolve problems. Monitoring and logging help organizations detect and resolve problems quickly, reducing downtime and improving software reliability.

Collaboration and Communication: DevOps emphasizes collaboration and communication between developers, operations teams, and other stakeholders. This helps ensure that everyone is working together towards a common goal of delivering high-quality software quickly and reliably.

These are just a few of the key processes involved in DevOps. In practice, DevOps may also involve other processes, such as *security testing, load testing,* and *performance testing*. The goal of DevOps is to continuously improve the software delivery process, so the specific processes involved can vary from organization to organization.

2. Background

2.1 State of the Art

The literature on DevSecOps is extensive, and many theoretical articles and case studies address the need to identify possible required skill sets that can be used as a reference when implementing DevSecOps. The following are a few that come close in that direction.

Since the moment that DevSecOps started being conceived, essential aspects to be considered emerged and were identified. As an example, the work of (Lee, 2018) describes the importance of collaboration between different groups within IT, specifically development, security, and operations/implementation, for the successful implementation of DevSecOps. However, it acknowledges that trust issues and conflicts between these groups are common. In response, the research proposes a framework based on Agency Theory to understand the role of goal incongruency and information asymmetry in the context of DevSecOps. The framework aims to shed light on these factors and their impact on effective DevSecOps implementation, but there are no advances towards the definition of a competency framework that explicitly identifies what are the critical characteristics to make DevSecOps succeed.

The work of Sanchez-Gordon and Colomo-Palacios (Sánchez-Gordón & Colomo-Palacios, 2020) characterizes DevSecOps culture based on the findings from the selected studies. Thirteen attributes of DevSecOps culture are identified and classified, including: collaboration, sharing knowledge, feedback, continuous improvement mindset, communication, responsibility, trust, experimentation, leadership, commitment and agreement, blamelessness, among others. The authors note that DevSecOps culture is relatively underexplored in the academic community. However, the identified attributes provide valuable insights for further research on the topic. They emphasize the need for collaboration and knowledge sharing among development, operations, and security teams. Continuous feedback loops, a focus on continuous improvement, effective communication, and building trust are also highlighted as essential elements of DevSecOps culture. The paper emphasizes the role of leadership in driving cultural change and the importance of personal commitment to security practices. Overall, the paper sheds light on the cultural aspects of DevSecOps and highlights the need for organizations to build a security culture to effectively integrate security into the DevOps process. The identified attributes provide a starting point for further research and understanding of DevSecOps culture.

In the same line, (Myrbakken & Colomo-Palacios, 2017) discusses the characteristics of DevSecOps based on a review of the literature. It identifies the principles and practices that characterize successful DevSecOps implementation in software development processes. Among the principles is mentioned Culture, Automation, Measurement, Sharing and Shift security to the left. Among the practices the authors highlight the importance of controlling Threat modeling and risk assessments, Continuous testing, Monitoring and logging, Security as code, and Red-Team and security drills. According to the authors, by adopting these principles and practices, organizations can successfully integrate security into their software development processes, aligning with the DevSecOps approach and enhancing overall security. In this paper, besides inspiring and very illustrative about the state of the practice, there is no a clear framework definition to be used as a guide towards guaranteeing DevSecOps effective deployment.

From an educational perspective, the work of Capozucca and Guelfi (Capozucca & Guelfi, 2020) presents a standard-based process to obtain a DevOps graduate education programme with no explicit inclusion of the security branch and restricted to one standard (SWECOM). Similarly, (Impagliazzo et al., 2020) based on SWECOM and CC2020 (the Global Guidelines for Baccalaureate Degrees in Computing (CC2020 Task Force, 2020)) to present a skillset for the software engineering profession, however, not focusing on DevSecOps albeit mentioning security as an important complement to development and operations in IT industry, besides enabled with the generic purpose of CC2020 of reflecting "*the changing dynamics of computing, computing education research, and the workplace*" (Ormond, 2021). In the same line, (Alarifi et al., 2016) present a SWEBOK based process for critical factors identification in the goal of mitigating the skills shortage in the software market, with no explicit addressing of the DevSeOps or Devops agility or generic success enablers.

Despite the extensive literature on what DevSecOps is and how to implement it, no work has been found that explicitly attempts to formally frame the definition of skills required for the different areas of DevSecOps to function properly. That is why this report addresses these concerns in the following sections.

2.2 Relevant Competency Frameworks

In this section we present the most pioneering existing competency models (Frezza et al., 2018). These models are on the bases of most of the current educational and training programs being used to train software professionals. Comparing DevSecOps skills needed, with the ones owned by current software professionals we will be able to provide a picture of the gaps that already exist in current software professional to put DevSecOps into practice and cover existing gaps for the near future training needs.

2.2.1 Software Engineering Competency Model (SWECOM)

The Software Engineering Competency Model (SWECOM) (Burgess et al., 2014) is a framework designed to assess the competencies of software engineers, and consists of a set of competency areas that are relevant to software engineering, including knowledge, skills and attitudes. The competency areas defined here are further divided into specific competencies that can be assessed and measured, which is of interest when approached from an improvement, effectiveness and efficiency perspective.

The SWECOM evaluation process involves assessing software engineers based on their performance in different competency areas. This can be done through a combination of self-assessment, peer review and expert panel evaluation processes. SWECOM aims to provide a structured and systematic approach to assessing and improving the competence of software engineers. By identifying areas of strength and weakness, software engineers, or those in charge of the strategy that takes care of their functions, can take steps to improve their skills and knowledge, which would result in helping them to be more effective in their roles.

SWECOM has been used by organizations and universities around the world to assess the competence of software engineers and develop training programs that address areas of weakness. It is a valuable tool for improving the quality of software engineering practices and ensuring that software engineers have the skills and knowledge necessary to succeed in their roles.

The principal elements of the SWECOM may be grouped as: Cognitive Skills, Behavioral Attributes and Skills, Requisite Knowledge, Technical Skills, Related Disciplines.

Among the weaknesses of SWECOM, it stands out the fact that it covers technical skills but does not include project management or general management skills other than "*to identify the behavioral attributes and skills of effective software developers and the leadership skills needed for software project technical leaders of various skill areas*"(Burgess et al., 2014).

2.2.2 US IT Competency Model

The US IT Competency Model framework was developed by the US Department of Labor to define and measure the required competencies to succeed in the information technology (IT) industry. Its main goal is to provide a common language and standard for IT jobs, education, training, and career development.

The competency model consists of seven big skill areas that are essential for IT professionals:

Technical Skills: referred to the ability of applying technical knowledge and skills to design, develop, test, and maintain software, hardware, and other technological systems.

- ***Business Skills:*** comprising the ability to understand and analyze business processes, business requirements, and efficiency in project management.
- ***Communication Skills:*** refers to the ability of communicating effectively with colleagues, clients, and stakeholders using written and verbal communication.
- ***Customer Service Skills:*** includes the ability to provide a good service to clients and customers, including skills related to problem-solving and conflict resolution.

- ***Management Skills:*** includes the ability to effectively plan, organize, and manage projects, people, and resources.
- ***Professionalism:*** it refers to the ability to exhibit an ethical behavior, maintain confidentiality, and adhere to professional standards and best practices.
- ***Leadership:*** regards the ability to inspire and motivate other people and teams, to build and maintain relationships, and to facilitate teamwork and collaboration.

In addition to technical competencies, the model also includes three core competencies that are necessary for success in any IT role: Business, Industry, Technology.

This competency model provides a detailed description of the knowledge, skills, and abilities required for each skill area at different levels of proficiency. It also identifies core competencies that are essential for all IT professionals, regardless of their job roles or specializations. It is widely used by IT employers, educators, and training providers to design and evaluate job requirements, curricula, and training programs. It is also used by job seekers and career changers to assess their skills and identify areas for development.

2.2.3 Skills Framework for the Information Age (SFIA 8)

The SFIA (Skills Framework for the Information Age) (SFIA Foundation, 2021) is a globally-recognized framework that defines the skills and competencies needed for professionals in the information and communication technology (ICT) sector. It was developed by the British Computer Society (BCS) in collaboration with industry experts and leading organizations in the ICT industry.

The SFIA framework consists of a set of skills and competencies that are organized into a hierarchical structure, with each skill or competency classified into one of seven levels of responsibility and autonomy.

The SFIA framework is widely used by organizations to manage their human capital, including skills assessment, skills development, and skills gap analysis. The SFIA framework is updated regularly to reflect changes in the ICT industry and to ensure its relevance to the evolving needs of organizations and professionals in the field.

The SFIA framework includes over 100 skills and competencies, which are organized into six categories:

- ***Strategy and Architecture:*** This category includes skills related to the development of technology strategies, enterprise architectures, and technical roadmaps. Some of the skills in this category include IT strategy and planning, enterprise and business architecture, and solution architecture.
- ***Business Change:*** This category includes skills related to the management of technology-enabled change within an organization. Some of the skills in this category include business analysis, project and programme management, and portfolio management.
- ***Development and Implementation:*** This category includes skills related to the design, development, and implementation of technology solutions. Some of the skills in this category include software development, testing, deployment and integration, and technical support.
- ***Delivery and Operation:*** This category includes skills related to the delivery and operation of technology solutions, including service management, operations management, and technical support.
- ***Cybersecurity:*** This category includes skills related to cybersecurity, including risk management, security architecture, and security operations.
- ***Emerging Technology:*** This category includes skills related to emerging technologies, including artificial intelligence, machine learning, and blockchain.

Each skill or competency is classified into one of seven levels of responsibility and autonomy, ranging from "follow" (level 1) to "set strategy" (level 7). This hierarchical structure enables organizations to assess the skills and competencies of their employees, plan career paths, and identify skills gaps that need to be addressed.

2.2.4 P-CMM

The People Capability Maturity Model (P-CMM) (Curtis et al., 2001) is a conceptual framework that helps companies and organizations in general to improve their people management practices. This framework was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University and is based on the recognized Capability Maturity Model Integration (CMMI) framework.

The P-CMM is structured around five maturity levels, each representing a different stage in an organization's ability to effectively manage its people. The P-CMM levels are:

- ***Initial Level:*** where an organization's people management practices are ad hoc and unstructured. These are actions for a specific purpose, with no automatic replicability.
- ***Repeatable level:*** where an organization has made an effort to establish and formalize basic people management processes, while being able to repeat them consistently.
- ***Defined level:*** at this level an organization has a well-defined set of people management processes that are followed consistently throughout the organization. There is no ambiguity about what the processes look like, and the guidelines for moving them forward are known.
- ***Managed level:*** where an organization has established metrics and measures to monitor its people management processes, leading it to be able to use this information to make data-driven decisions, giving critical support to senior management and how decisions are made.
- ***Optimizing level:*** at this level, an organization continuously improves its people management processes based on data and feedback. It breaks with the paradigm of staticity and recognizes that the environment is changing, as are the needs to be addressed, so it is important to develop the capacity for adaptability and continuous improvement.

The P-CMM covers a wide range of people management practices, including staffing, training and development, performance management, compensation and benefits, career development and succession planning. By following the P-CMM framework, organizations can improve their ability to attract, develop, motivate and retain the talent they need to achieve their business objectives.

3. Research Approach

In this report, the comparison will be described in the direction of from DevSecOps critical sectors to the reference models, standards, best practices, and technologies to guide DevSecOps deployment. In this work, a mapping is made to identify how DevSecOps critical sectors can be correctly enabled from the specifications of the reference models, standards and best practices. A “mapping” is understood here as a modification of the process proposed by [3]. In this work, the mapping is a step-by step theoretical comparison process in which for each critical DevSecOps sector, and its deployment’s requirements, the set of skills from industry standards is compared, so that the skills of the industry considered may become enablers of the DevSecOps, so they are explicitly mapped.

The goal of the work can be summarized as follows:

The objective is to analyze the correlation between IT professional competency standards and critical sectors of DevSecOps, aiming to propose a comprehensive skillset that stakeholders driving DevSecOps must possess for effective deployment, management, and continuous improvement. This analysis will primarily focus on decision makers' perspectives within the software development context and the broader software industry. By bridging the gap between competency standards and DevSecOps requirements, this study aims to provide decision makers with actionable insights for enhancing their proficiency in driving successful DevSecOps initiatives.

Two main research questions will be addressed through this study:

- **RQ1:** What are the reference models, standards, best practices, and technologies to guide DevSecOps deployment?
- **RQ2:** What is the skillset required to successfully deploy DevSecOps?

The methodological approach for the mapping is illustrated in Figure 1.

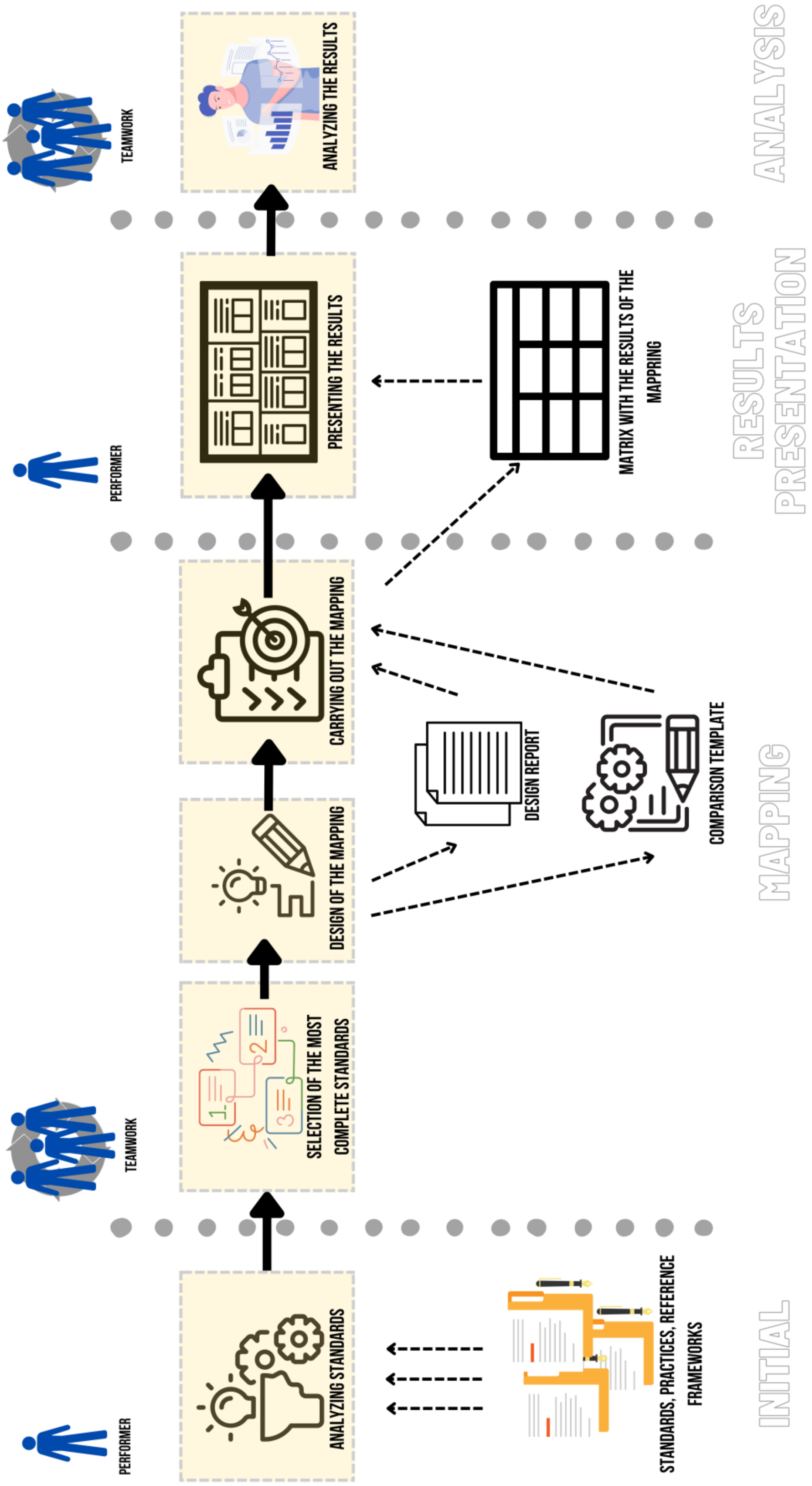


FIGURE 2. Research approach

Initial Phase. In this phase, the focus is on identifying standards, best practices, and frameworks that would allow the identification of skills needed to play a necessary role in the performance of DevSecOps. To do so, a review of the set of reference frameworks used in the software industry should be made.

Mapping phase. In this phase, once the available standards are known, those that are to be reviewed in depth are selected for their alignment with the interests of DevSecOps and for the degree of knowledge that software engineers and professionals in the sector may have about them. Next, the mapping design must be done, which means identifying how the comparison process between the critical areas of DevSecOps and the structural elements of each of the standards to be reviewed will be carried out. A comparison template should be created to facilitate the work, and a summary report of the designed mapping should be generated to be used as a reference when making the comparison with each selected standard. The critical part of this phase is the iterative realization of the comparative process of the DevSecOps area - Standard's element pair. After doing as many iterations as necessary, you can proceed to the results presentation phase.

Results presentation phase. In this phase, all the information from the comparisons made in the previous phase is collected, and everything is synthesized in a matrix that in the first column contains the critical areas of the DevSecOps, and then in the first row contains a column for each of the architectural elements of the standards to be considered. The crossing between rows (DevSecOps areas) and columns (elements of the standards), allows to synthesize the skills necessary for the DevSecOps area to be correctly carried out, i.e., the capabilities and skills that could make it possible for the area to function correctly are identified.

Analysis phase. In this phase, with the synthesis matrix already elaborated, the work is limited to making explicit the set of skills identified as critical for the functioning of DevSecOps, which could be a referential framework for the optimal deployment of DevSecOps.

In the following section, the two research questions, RQ1: What are the reference models, standards, best practices, and technologies to guide DevSecOps deployment? and RQ2: What is the skillset required to successfully deploy DevSecOps? are addressed.

4. Identifying the DevSecOps Competency skillset.

4.1 Initial phase.

In this phase, special attention has been paid to what is the supply of standards that address the identification of skills needed to address the activities related to software development and the direction and management of the software process in general. As a result, the following standards were found:

The U.S. IT Competency Model (*Information Technology Competency Model*, 2021), known as the "US IT Competency Model," created by the U.S. Department of Labor. This model was developed in collaboration with experts in the information technology industry and with input from professionals in the field. The objective of the model is to provide a framework for understanding the skills and competencies needed in the IT industry, and to serve as a reference for the development of education, training and professional development programs in the IT sector in the United States.

The Software Engineering Competency Model (SWECOM) (Burgess et al., 2014), which describes the competencies that software engineers must have in order to participate in the development and modification of software-intensive systems. Work areas and activities are specified for each competency.

The "Skills Framework for the Information Age" (SFIA) (SFIA Foundation, 2021), developed by the British organization SFIA Foundation. The SFIA Foundation is a non-profit organization dedicated to promoting and developing the SFIA skills framework. The SFIA framework provides a framework for describing and assessing the skills and competencies required for information and communication technology (ICT) professionals at different levels of experience and responsibility. SFIA is widely used around the world by organizations and IT professionals to assess and develop skills, manage careers and plan skills development in the ICT sector.

The People Capability Maturity Model (P-CMM) (Curtis et al., 2001), created by the Software Engineering Institute (SEI) at Carnegie Mellon University in the United States. The SEI developed the P-CMM as a framework to help organizations improve the capability and maturity of their human resource management, especially in the field of software development and information technology. The model focuses on talent development and management, career planning, competency management and continuous improvement of staff capabilities. The P-CMM is based on the principles and approaches of the Process Capability Maturity Model (CMM) and has been used by several organizations to improve the management and development of their human capital in the field of information technology.

Specific details about the mentioned models of competencies are given in section 2. A matrix summarizing the coverage that each model has over the critical sectors of DevSecOps was created following the structure shown in Figure 2. The full matrix is available at promiseinnovatech.com.

4.2 Mapping phase

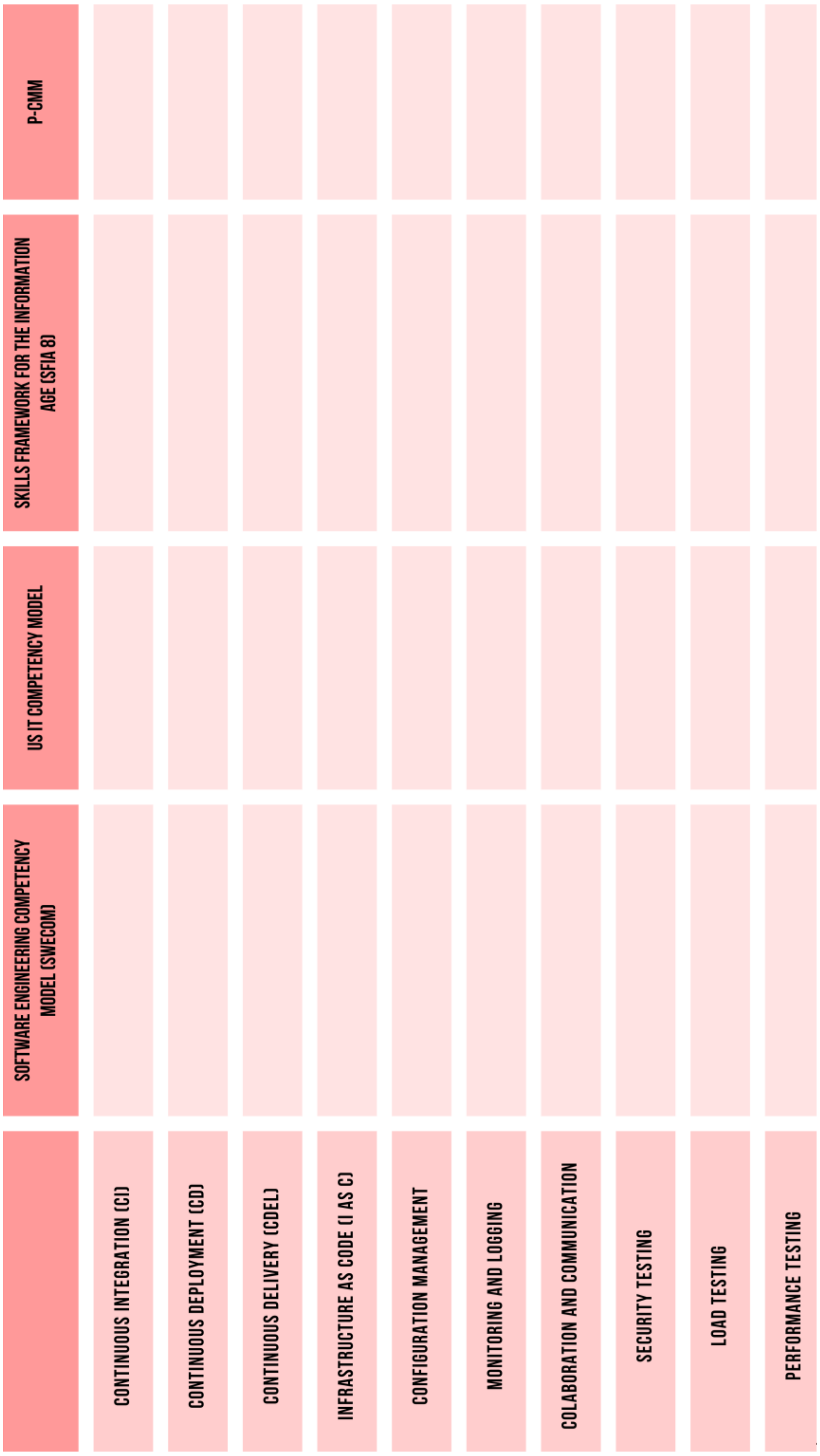
Following with the research approach, it was decided to go ahead and design the mapping considering the SFIA vs. the critical areas of DevSecOps. SFIA was selected since it is the wider competency model, so it covers a broader range of skills for a wider amount of sectors and professions in the software industry. Besides being general, SFIA contemplates a very well defined set of skills that may be appropriate to be compared to to the needs of DevSecOps areas.

The mapping design consist of an iterative process described below.

1. Select the pair SFIA's kill – DevSecOps area.
    a. Select the SFIA's skill to map.
    b. State the SFIA skill description.
    c. Select the DevSecOps Area to map.
    d. State the DevSecOps' area description.
2. For each selected skill, answer yes/no to the following questions.
    a. Q1. Is this skill needed to guarantee this DevSecOps area's functioning?

    b. Q2. Can this DevSecOps' area function even if this skill is not present?

3. For each selected skill, categorize the skill according to:
    a. If Q1= Yes, and Q2=No, then mark the skill as "essential".
    b. If Q1=Yes, and Q2=Yes, then mark the skill as "desirable".
    c. If Q1=No, and Q2=Yes|No, then mark the skill as "dispensable".
4. Double-check the skill categorization.
5. Generate a document with all the essential skills identified.
6. Prioritize the identified skills for each DevSecOps area.

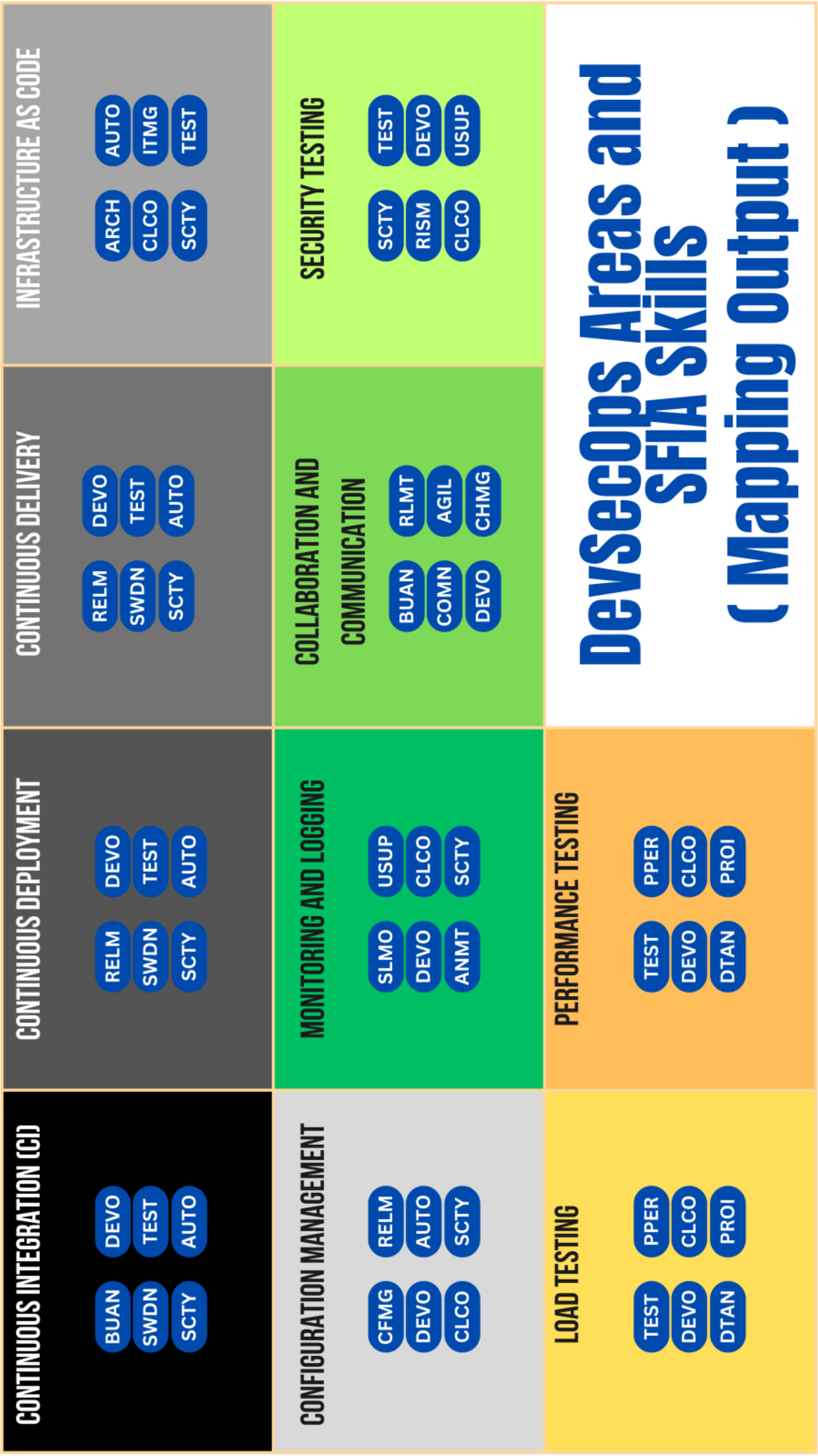


	SOFTWARE ENGINEERING COMPETENCY MODEL (SWECOM)	US IT COMPETENCY MODEL	SKILLS FRAMEWORK FOR THE INFORMATION AGE (SFIA 8)	P-CMM
CONTINUOUS INTEGRATION (CI)				
CONTINUOUS DEPLOYMENT (CD)				
CONTINUOUS DELIVERY (CDEL)				
INFRASTRUCTURE AS CODE (I AS C)				
CONFIGURATION MANAGEMENT				
MONITORING AND LOGGING				
COLABORATION AND COMMUNICATION				
SECURITY TESTING				
LOAD TESTING				
PERFORMANCE TESTING				

FIGURE 3. Matrix mapping DevSecOps vs. Skills reference frameworks.

4.3 Results presentation phase

As a result of the mapping, it has been possible to identify in a prioritized way the critical skills of the SFIA standard to ensure that each of the DevSecOps areas are working correctly. In the following graphic (Figure 3), the mapping results are summarized.



DevSecOps Areas and SFIA Skills (Mapping Output)

CONTINUOUS INTEGRATION (CI)
BUAN SWDN SCTY DEVO TEST AUTO

CONTINUOUS DEPLOYMENT
RELM SWDN SCTY DEVO TEST AUTO

CONTINUOUS DELIVERY
RELM SWDN SCTY DEVO TEST AUTO

INFRASTRUCTURE AS CODE
ARCH CLCO SCTY AUTO ITMG TEST

CONFIGURATION MANAGEMENT
CFMG DEVO CLCO RELM AUTO SCTY

MONITORING AND LOGGING
SLMO DEVO ANMT USUP CLCO SCTY

COLLABORATION AND COMMUNICATION
BUAN COMN DEVO RLMT AGIL CHMG

SECURITY TESTING
SCTY RISM CLCO TEST DEVO USUP

LOAD TESTING
TEST DEVO DTAN PPER CLCO PROI

PERFORMANCE TESTING
TEST DEVO DTAN PPER CLCO PROI

FIGURE 4. Mapping Output with skills for each DevSecOps area.

The following subsection justifies the relevance of each skill mapped to each of the DevSecOps areas.

4.4 Analysis phase: The standardized skills of professionals in DevSecOps

It is critical to identify “what to observe” while trying to identify the required skillset to enable DevSecOps successful deployment. In this regard, we adopt the view of (Clear, 2017), who argue that it is important to consider: capabilities, cultural enablers and technological enablers. Beyond this, it is important to adopt also the term “disposition” as it concerns “*not what abilities people have, but how people are disposed to use those abilities*” (Clear, 2017). Following several attempts to identify the skills are given, considering mainly whether organizational or people-related skills are mentioned.

In this subsection, the mapping of skills based on the largest skills framework (SFIA) is given, considering the relevance for the essential aspects of DevSecOps. In Figure 4, the summarized mapping is shown, while details are given in the following paragraphs.

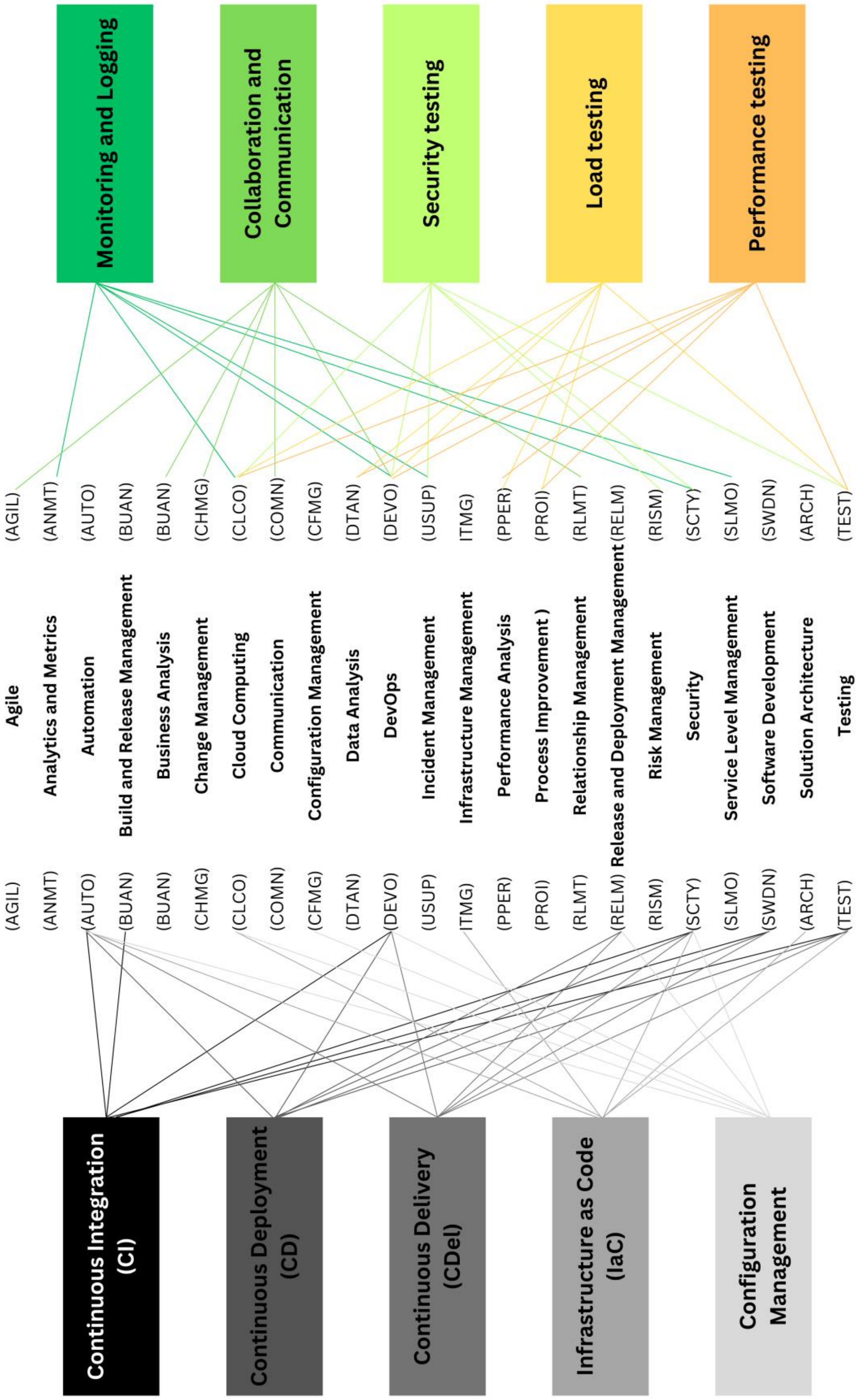

Monitoring and Logging
Collaboration and Communication
Security testing
Load testing
Performance testing
Agile (AGIL)
Analytics and Metrics (ANMT)
Automation (AUTO)
Build and Release Management (BUAN)
Business Analysis (BUAN)
Change Management (CHMG)
Cloud Computing (CLCO)
Communication (COMN)
Configuration Management (CFMG)
Data Analysis (DTAN)
DevOps (DEVO)
Incident Management (USUP)
Infrastructure Management ITMG)
Performance Analysis (PPER)
Process Improvement) (PROI)
Relationship Management (RLMT)
Release and Deployment Management (RELM)
Risk Management (RISM)
Security (SCTY)
Service Level Management (SLMO)
Software Development (SWDN)
Solution Architecture (ARCH)
Testing (TEST)
Continuous Integration (CI)
Continuous Deployment (CD)
Continuous Delivery (CDel)
Infrastructure as Code (IaC)
Configuration Management

FIGURE 5. Mapping the SFIA skills for DevSecOps.

4.4.1 Continuous Integration.

SFIA (Skills Framework for the Information Age) is a widely used framework for identifying and managing IT skills. In the context of DevSecOps and Continuous Integration (CI), the following SFIA skills would be particularly relevant:

Build and Release Management (BUAN): This skill involves managing the process of building and releasing software code and ensuring that it is deployed correctly in different environments. This is a critical skill for CI, as it involves automating the building, testing, and deployment of software code in a continuous and streamlined manner.

DevOps (DEVO): This skill, (considered in SFIA without the SEC yet) involves integrating development and operations processes to improve software delivery and maintenance. DevOps skills are particularly important for implementing CI in a DevSecOps environment, as they involve coordinating the development, testing, and deployment of software across different teams and environments.

Software Development (SWDN): This skill involves developing software code using programming languages, tools, and frameworks. CI requires the ability to write high-quality, testable code that can be easily integrated with other codebases and tested quickly and efficiently.

Testing (TEST): This skill involves designing and executing tests to validate software code and ensure that it meets functional and non-functional requirements. Testing skills are essential for implementing CI, as they involve automating the testing process and ensuring that tests are executed at every stage of the software development lifecycle.

Security (SCTY): This skill involves identifying and mitigating security risks in software systems. In a DevSecOps environment, security skills are essential for ensuring that security is built into every stage of the software development process, from code development to deployment and maintenance.

Automation (AUTO): This skill involves automating repetitive or manual tasks using tools and scripts. Automation skills are particularly important for implementing CI, as they involve automating the building, testing, and deployment of software code in a continuous and streamlined manner.

Overall, a strong understanding of these SFIA skills is essential for supporting Continuous Integration in a DevSecOps environment, as they involve coordinating the development, testing, and deployment of software across different teams and environments while ensuring that security is built into every stage of the process.

4.4.2 Continuous Deployment

Continuous Deployment (CD) is the next step after Continuous Integration (CI) in DevSecOps, where the code changes are automatically deployed to production after successful testing in a pre-production environment. The following SFIA skills are essential to support Continuous Deployment in DevSecOps:

Release and Deployment Management (RELM): This skill involves planning, coordinating, and deploying software releases, including managing the configuration of the production environment. In a DevSecOps environment, this skill is crucial for automating the deployment of software to production, ensuring that all the necessary approvals and tests are completed before deployment.

DevOps (DEVO): DevOps skills are also essential for supporting Continuous Deployment in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments.

Software Development (SWDN): The ability to develop high-quality code is critical for Continuous Deployment, as the software changes need to be deployed to production automatically without any manual intervention. Developers need to write code that is testable, modular, and maintainable, with proper version control.

Testing (TEST): Testing skills are essential to ensure that the code changes deployed to production are of high quality and do not impact the availability or security of the system. Test automation is a critical skill that is necessary to enable continuous deployment, as it involves automating the testing of the software changes.

Security (SCTY): Security is a critical consideration in DevSecOps, and security skills are essential to support Continuous Deployment. In a Continuous Deployment environment, security checks need to be automated to ensure that the code changes do not introduce any security vulnerabilities.

Automation (AUTO): Automation is an essential skill for Continuous Deployment, as it involves automating the deployment of software to production, automating the testing process, and automating the security checks.

Overall, a strong understanding of these SFIA skills is essential to support Continuous Deployment in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments while ensuring that security is built into every stage of the process.

4.4.3 Continuous Delivery

Continuous Delivery (CD) is an approach to software development and deployment that involves continuously delivering software changes to production. Unlike Continuous Deployment, CD requires a manual approval process before changes are deployed to production. The following SFIA skills are essential to support Continuous Delivery in DevSecOps:

Release and Deployment Management (RELM): This skill involves planning, coordinating, and deploying software releases, including managing the configuration of the production environment. In a DevSecOps environment, this skill is crucial for automating the deployment of software to pre-production environments and managing the manual approval process before deploying to production.

DevOps (DEVO): DevOps skills are also essential for supporting Continuous Delivery in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments.

Software Development (SWDN): The ability to develop high-quality code is critical for Continuous Delivery, as the software changes need to be deployed to pre-production environments automatically without any manual intervention. Developers need to write code that is testable, modular, and maintainable, with proper version control.

Testing (TEST): Testing skills are essential to ensure that the code changes deployed to pre-production environments are of high quality and do not impact the availability or security of the system. Test automation is a critical skill that is necessary to enable continuous delivery, as it involves automating the testing of the software changes.

Security (SCTY): Security is a critical consideration in DevSecOps, and security skills are essential to support Continuous Delivery. In a Continuous Delivery environment, security checks need to be automated to ensure that the code changes do not introduce any security vulnerabilities.

Automation (AUTO): Automation is an essential skill for Continuous Delivery, as it involves automating the deployment of software to pre-production environments, automating the testing process, and automating the security checks.

Overall, a strong understanding of these SFIA skills is essential to support Continuous Delivery in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments while ensuring that security is built into every stage of the process. The key difference between Continuous Delivery and Continuous Deployment is the manual approval process, which requires additional skills in release and deployment management.

4.4.4 Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is the practice of managing and provisioning infrastructure through code and automation tools. It involves using tools like Terraform, Ansible, Puppet, and Chef to define infrastructure configurations as code and then automate the provisioning and management of that infrastructure. The following SFIA skills are essential to support Infrastructure as Code in DevSecOps:

Solution Architecture (ARCH): This skill involves designing and implementing solutions that meet business requirements while considering scalability, maintainability, and security. In an IaC environment, this skill is essential for designing infrastructure configurations that are secure, scalable, and maintainable.

Automation (AUTO): Automation skills are essential for IaC, as it involves automating the provisioning, configuration, and management of infrastructure using code. Automation involves writing code that can deploy and manage infrastructure and automating the testing and deployment of infrastructure changes.

Cloud Computing (CLCO): Cloud computing skills are essential for IaC, as it involves provisioning and managing infrastructure in cloud environments like AWS, Azure, or GCP. Cloud computing skills include understanding cloud infrastructure components, networking, and security.

Infrastructure Management (ITMG): Infrastructure management skills are essential for IaC, as it involves managing infrastructure through code and automation tools. This skill involves understanding the various infrastructure components and tools, such as virtual machines, containers, databases, and networks.

Security (SCTY): Security is a critical consideration in DevSecOps, and security skills are essential to support IaC. In an IaC environment, security checks need to be automated to ensure that the infrastructure configurations are secure and meet compliance requirements.

Testing (TEST): Testing skills are essential to ensure that the infrastructure configurations deployed through IaC are of high quality and do not impact the availability or security of the system. Test automation is a critical skill that is necessary to enable continuous testing of the infrastructure.

Overall, a strong understanding of these SFIA skills is essential to support Infrastructure as Code in DevSecOps, as it involves designing, provisioning, configuring, and managing infrastructure through code and automation tools.

4.4.5 Configuration Management

Configuration management is an essential practice in DevSecOps that involves managing and tracking changes to the software and infrastructure configuration. The following SFIA skills are essential to support Configuration Management in DevSecOps:

Configuration Management (CFMG): This skill involves managing and tracking changes to the software and infrastructure configuration. In DevSecOps, this skill is crucial for automating the configuration of the software and infrastructure, ensuring that changes are tracked and auditable.

Release and Deployment Management (RELM): Release and deployment management skills are also essential for supporting Configuration Management in DevSecOps, as they involve planning, coordinating, and deploying software releases, including managing the configuration of the production environment.

DevOps (DEVO): DevOps skills are also essential for supporting Configuration Management in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments.

Automation (AUTO): Automation skills are critical for Configuration Management in DevSecOps, as they involve automating the configuration of the software and infrastructure, ensuring that changes are tracked and auditable.

Cloud Computing (CLCO): Cloud computing skills are also essential for supporting Configuration Management in DevSecOps, as they involve managing and provisioning infrastructure and services in the cloud.

Security (SCTY): Security is a critical consideration in DevSecOps, and security skills are essential to support Configuration Management. In a DevSecOps environment, security checks need to be automated to ensure that the configuration changes do not introduce any security vulnerabilities.

Overall, a strong understanding of these SFIA skills is essential to support Configuration Management in DevSecOps, as they involve managing and tracking changes to the software and infrastructure configuration, ensuring that changes are tracked and auditable while ensuring that security is built into every stage of the process.

4.4.6 Monitoring and logging

Monitoring and logging are critical practices in DevSecOps that involve monitoring the software and infrastructure for issues and logging events for auditing and troubleshooting. The following SFIA skills are essential to support Monitoring and Logging in DevSecOps:

Service Level Management (SLMO): This skill involves defining, measuring, and managing service level agreements (SLAs). In a DevSecOps environment, SLMO is crucial for ensuring that the monitoring and logging systems are meeting the required SLAs.

Incident Management (USUP): Incident management skills are also essential for supporting Monitoring and Logging in DevSecOps, as they involve identifying, tracking, and resolving incidents that impact the availability or security of the system.

DevOps (DEVO): DevOps skills are also essential for supporting Monitoring and Logging in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments.

Cloud Computing (CLCO): Cloud computing skills are also essential for supporting Monitoring and Logging in DevSecOps, as they involve managing and monitoring the infrastructure and services in the cloud.

Analytics and Metrics (ANMT): Analytics and metrics skills are essential for monitoring and logging in DevSecOps, as they involve collecting and analyzing data to identify trends and patterns in the system performance.

Security (SCTY): Security is a critical consideration in DevSecOps, and security skills are essential to support Monitoring and Logging. In a DevSecOps environment, security events need to be logged and monitored to detect any security threats or breaches.

Overall, a strong understanding of these SFIA skills is essential to support Monitoring and Logging in DevSecOps, as they involve monitoring the software and infrastructure for issues, logging events for auditing and troubleshooting, ensuring that the systems meet the required SLAs, and detecting any security threats or breaches.

4.4.7 Collaboration and Communication

Collaboration and communication are essential practices in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments. The following SFIA skills are essential to support Collaboration and Communication in DevSecOps:

Business Analysis (BUAN): This skill involves understanding and analyzing business requirements and translating them into technical requirements. In

DevSecOps, BUAN is crucial for understanding the business goals and ensuring that the software and infrastructure support those goals.

Relationship Management (RLMT): Relationship management skills are also essential for supporting Collaboration and Communication in DevSecOps, as they involve building and maintaining relationships with different teams and stakeholders, ensuring that everyone is aligned towards the same goals.

Communication (COMN): Communication skills are critical for Collaboration and Communication in DevSecOps, as they involve communicating technical information and updates to different teams and stakeholders, ensuring that everyone is informed and up-to-date.

Agile (AGIL): Agile skills are also essential for supporting Collaboration and Communication in DevSecOps, as they involve working collaboratively and iteratively to develop and deploy software, ensuring that everyone is working towards the same goals.

DevOps (DEVO): DevOps skills are also essential for supporting Collaboration and Communication in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments.

Change Management (CHMG): Change management skills are essential for supporting Collaboration and Communication in DevSecOps, as they involve managing and communicating changes to the software and infrastructure, ensuring that everyone is informed and prepared for the changes.

Overall, a strong understanding of these SFIA skills is essential to support Collaboration and Communication in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments, ensuring that everyone is aligned towards the same goals, informed and up-to-date, and prepared for changes.

4.4.8 Security Testing

Security testing is a crucial practice in DevSecOps that involves testing the software and infrastructure for vulnerabilities and security weaknesses. The following SFIA skills are essential to support Security Testing in DevSecOps:

Security (SCTY): Security skills are essential for supporting Security Testing in DevSecOps, as they involve understanding and mitigating security risks and vulnerabilities in the software and infrastructure.

Testing (TEST): Testing skills are also essential for supporting Security Testing in DevSecOps, as they involve planning, designing, and executing tests to identify security vulnerabilities and weaknesses in the software and infrastructure.

Risk Management (RISM): Risk management skills are also crucial for supporting Security Testing in DevSecOps, as they involve identifying, assessing, and mitigating risks to the security of the software and infrastructure.

DevOps (DEVO): DevOps skills are also essential for supporting Security Testing in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments, ensuring that security testing is integrated into the DevSecOps process.

Cloud Computing (CLCO): Cloud computing skills are also essential for supporting Security Testing in DevSecOps, as they involve managing and securing the infrastructure and services in the cloud.

Incident Management (USUP): Incident management skills are also essential for supporting Security Testing in DevSecOps, as they involve identifying and responding to security incidents and breaches.

Overall, a strong understanding of these SFIA skills is essential to support Security Testing in DevSecOps, as they involve identifying and mitigating security risks and vulnerabilities in the software and infrastructure, planning and executing tests to identify security weaknesses, integrating security testing into the DevSecOps process, managing and securing cloud infrastructure, and responding to security incidents and breaches.

4.4.9 Load testing

Load testing is a crucial practice in DevSecOps that involves testing the software and infrastructure to ensure that it can handle expected levels of traffic and usage. The following SFIA skills are essential to support Load Testing in DevSecOps:

Testing (TEST): Testing skills are essential for supporting Load Testing in DevSecOps, as they involve planning, designing, and executing tests to evaluate the performance and scalability of the software and infrastructure under different load conditions.

Performance Analysis (PPER): Performance analysis skills are also essential for supporting Load Testing in DevSecOps, as they involve analyzing and interpreting performance metrics and data to identify performance bottlenecks and areas for improvement.

DevOps (DEVO): DevOps skills are also essential for supporting Load Testing in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments, ensuring that load testing is integrated into the DevSecOps process.

Cloud Computing (CLCO): Cloud computing skills are also essential for supporting Load Testing in DevSecOps, as load testing is often performed on cloud-based infrastructure, and cloud technologies offer scalability and flexibility to support load testing.

Data Analysis (DTAN): Data analysis skills are also crucial for supporting Load Testing in DevSecOps, as they involve analyzing and interpreting performance data and metrics to identify patterns, trends, and areas for improvement.

Process Improvement (PROI): Process improvement skills are also essential for supporting Load Testing in DevSecOps, as they involve identifying and implementing improvements to the load testing process to enhance its efficiency and effectiveness.

Overall, a strong understanding of these SFIA skills is essential to support Load Testing in DevSecOps, as they involve planning and executing load tests to evaluate performance and scalability, analyzing and interpreting performance data, integrating load testing into the DevSecOps process, managing and securing cloud infrastructure, and continuously improving the load testing process.

4.4.10 Performance testing

Performance testing is a crucial practice in DevSecOps that involves testing the software and infrastructure to ensure that it meets the expected performance requirements. The following SFIA skills are essential to support Performance Testing in DevSecOps:

Testing (TEST): Testing skills are essential for supporting Performance Testing in DevSecOps, as they involve planning, designing, and executing tests to evaluate the performance of the software and infrastructure under different scenarios and conditions.

Performance Analysis (PPER): Performance analysis skills are also essential for supporting Performance Testing in DevSecOps, as they involve analyzing and interpreting performance metrics and data to identify performance bottlenecks and areas for improvement.

DevOps (DEVO): DevOps skills are also essential for supporting Performance Testing in DevSecOps, as they involve coordinating the development, testing, and deployment of software across different teams and environments, ensuring that performance testing is integrated into the DevSecOps process.

Cloud Computing (CLCO): Cloud computing skills are also essential for supporting Performance Testing in DevSecOps, as performance testing is often performed on cloud-based infrastructure, and cloud technologies offer scalability and flexibility to support performance testing.

Data Analysis (DTAN): Data analysis skills are also crucial for supporting Performance Testing in DevSecOps, as they involve analyzing and interpreting performance data and metrics to identify patterns, trends, and areas for improvement.

Process Improvement (PROI): Process improvement skills are also essential for supporting Performance Testing in DevSecOps, as they involve identifying and implementing improvements to the performance testing process to enhance its efficiency and effectiveness.

Overall, a strong understanding of these SFIA skills is essential to support Performance Testing in DevSecOps, as they involve planning and executing performance tests to evaluate performance and identify performance bottlenecks, analyzing and interpreting performance data, integrating performance testing into the DevSecOps process, managing and securing cloud infrastructure, and continuously improving the performance testing process.

5. First Discussion

As a result of the process followed in section 4, and in accordance to the research approach described in section 3, in summary, several skills were mentioned repeatedly as they were relevant to more than one activity in DevSecOps. The summarized list of relevant skills is:

- Agile (AGIL)
- Analytics and Metrics (ANMT)
- Automation (AUTO)
- Build and Release Management (BUAN)
- Business Analysis (BUAN)
- Change Management (CHMG)
- Cloud Computing (CLCO)
- Communication (COMN)
- Configuration Management (CFMG)
- Data Analysis (DTAN)
- DevOps (DEVO)
- Incident Management (USUP)
- Infrastructure Management (ITMG)
- Performance Analysis (PPER)
- Process Improvement (PROI)
- Relationship Management (RLMT)
- Release and Deployment Management (RELM)
- Risk Management (RISM)
- Security (SCTY)
- Service Level Management (SLMO)
- Software Development (SWDN)

- Solution Architecture (ARCH)
- Testing (TEST)

In this work we have been able to approximate the SFIA standard skills needed to support the proper functioning of the different areas of DevSecOps.

The purpose of the first research question has been satisfied by identifying the main industry competency frameworks detailing specific skills that we could use. The identified models of competencies are the Software Engineering Competency Model (SWECOM) (Impagliazzo et al., 2020), the US IT Competency Model (*Information Technology Competency Model*, 2021), the Skills Framework for the Information Age (SFIA 8) (SFIA Foundation, 2021), and the People Capability Maturity Model (P-CMM) (Curtis et al., 2001).

The mapping process that was done from the SFIA framework allowed us to identify the skills that each area of DevSecOps requires in order to function well, which was listed in the previous section, with a summarized list of 23 key skills, thus answering the second research question we posed.

Although we consider that we have made an important advance by making use of knowledge in the area of the software industry, it is important to continue with this work and expand it to take into account dimensions that are not being considered in the standards that were used as a frame of reference. Still we must mention some threats to the validity of the study and future work.

On the one hand, it is necessary to identify the organizational aspects that could allow DevSecOps to not only function correctly, but to be viable and sustainable over time. For this, it is necessary to continue this work by bringing in knowledge from organizational cybernetics, area which has long proven its usefulness in helping organizations to be intelligent, robust, and resilient. Organizational cybernetics, and specifically the Viable Systems Model (Beer, 1964, 1972a, 1984, 1985), provides an architectural structure in which five elementary functions are defined for organizations to survive in the long term: operation, coordination, control, intelligence, and politics or identity. According to organizational cybernetics, we may affirm that ensuring that these functions are present in DevSecOps provides them with organizational characteristics with a high impact on performance, productivity, and chances of survival over time.

Organizational cybernetics provides a whole architectural structure that if mapped with DevSecOps could help us identify how well established, from a cybernetics point of view, it is being. This is a work in progress and will be submitted for validation and publication soon.

On the other hand, addressing more deeply how the human factor impacts on DevSecOps performance is critical to be able to make a real improvement approach, since it is human beings who are behind the deployment of DevSecOps. There are already existing works that address this issue, mentioning general attributes of DevSecOps such as the ones described in (Sánchez-Gordón & Colomo-Palacios, 2020): collaboration, sharing knowledge, feedback, continuous improvement mindset, communication, responsibility, trust, experimentation, leadership, commitment and agreement, blameless, hiring new personnel, and transparency. There are other works revealing the need to consider more general cultural aspects, such as (Myrbakken & Colomo-Palacios, 2017), . As a continuation of this work, we want to expand the mapping that has been done to add human factors related skills to the skill set affecting DevSecOps. In the near future, we will present this work, in which in addition to this mapping we will identify the correspondence of human factors skills with those of the SFIA standard, allowing us to expand not only the impact on DevSecOps but potentially on the industry in general.

Finally, we would like to mention that we are in the process of designing an experiment in which we will invite experts with extensive experience, both from academia and industry, to validate the mapping done. To do so, we will make use of an interactive online questionnaire in which we will ask the experts to "crawl" the SFIA skills they consider essential for each area of DevSecOps. We will then compare the experts' opinions for two purposes: 1) To identify whether there is consensus, and all have a degree of agreement with the identification of the skills, and 2) To compare the experts' consensus selection with the proposal made in this work.

6. Skills from the cybernetics world

The Viable Systems Model (VSM) is a cybernetic theory developed by Stafford Beer in the 1970s (Beer, 1972b, 1984, 1985; Espejo & Harnden, 1990). It is a systems thinking approach aimed at understanding and designing organizations to effectively deal with complexity and maintain their viability in a rapidly changing environment (Andrade Sosa et al., 2007). The model draws inspiration from cybernetics, which is the study of communication and control in systems.

The VSM is particularly useful for identifying the essential features of a viable organization by providing a framework to analyze and assess an organization's structure, processes, and interactions. It helps in understanding how different components of an organization work together to achieve its purpose, adapt to changing circumstances, and remain viable over time.

The Viable Systems Model consists of five interrelated systems, or subsystems, which are as follows:

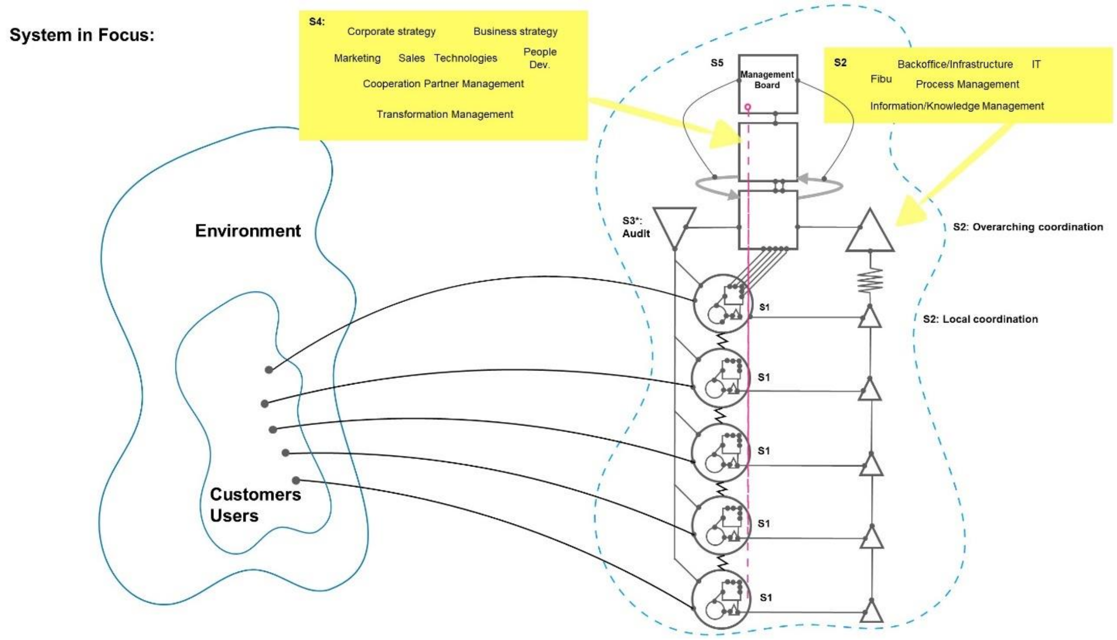


FIGURE 6. The viable systems model in focus.

6.1 System 1 – Operations

This subsystem represents the core activities of the organization. It is responsible for producing goods or delivering services, and it directly interacts with the external environment and stakeholders.

6.2 System 2 - Coordination

The coordination subsystem is responsible for integrating the activities of System 1. It sets the overall goals, defines policies, and ensures that the different parts of the organization are aligned towards the common purpose.

6.3 System 3 – Control

The control subsystem is responsible for monitoring the performance of System 1 in relation to the goals set by System 2. It gathers information, evaluates performance, and takes corrective actions when necessary to maintain alignment with the organization's purpose.

6.4 System 4 – Intelligence

The intelligence subsystem is concerned with the external environment and the challenges and opportunities it presents. It collects and processes information from the external environment and provides it to System 2 for decision-making and adaptation.

6.5 System 5 – Policy

The policy subsystem is responsible for setting the overall direction and guiding principles for the entire organization. It defines the boundaries within which Systems 1 to 4 operate and ensures that they are coherent and effective.

The key idea behind the Viable Systems Model is that **for an organization to be viable, it must have the capability to manage both internal complexity (within its own subsystems) and external complexity (in its environment)**. The model helps identify potential weaknesses in an organization's design and functioning, such as inadequate communication between subsystems, lack of adaptability, or inability to learn from the environment.

By using the VSM to analyze an organization, leaders can:

Improve **adaptability**: The model highlights how an organization can respond to changes in the environment effectively and modify its internal processes accordingly.

Enhance **communication** and **coordination**: It emphasizes the importance of clear communication and coordination between different parts of the organization to achieve its objectives.

Optimize **decision-making**: The model provides a structured approach to decision-making by delineating roles and responsibilities across the subsystems.

Identify **areas of improvement**: By examining the functioning of each subsystem, an organization can identify areas that need improvement and implement targeted changes.

Develop **resilience**: The VSM helps an organization become more robust in the face of unforeseen events or crises.

Foster **learning** and **innovation**: By monitoring the external environment through the intelligence subsystem, an organization can better adapt and innovate to stay competitive.

The Viable Systems Model is a valuable tool for understanding and designing organizations that can cope with complexity and maintain their viability (Andrade Sosa et al., 2007). By identifying the essential features of a viable organization, it offers a structured approach for leaders to optimize their organization's functioning, adaptability, and long-term success (Espinosa, 2023).

7. DevSecOps as viable systems

Conceptualizing DevSecOps as viable systems according to the Viable Systems Model (VSM) can provide valuable insights into how this approach can effectively address the challenges of integrating security practices into the software development and operations process. DevSecOps is an extension of the DevOps philosophy, which emphasizes collaboration and integration between development, IT operations, and quality assurance teams to deliver software more rapidly and reliably.

DevSecOps can be understood through the lens of the Viable Systems Model. A proposal on this follows:

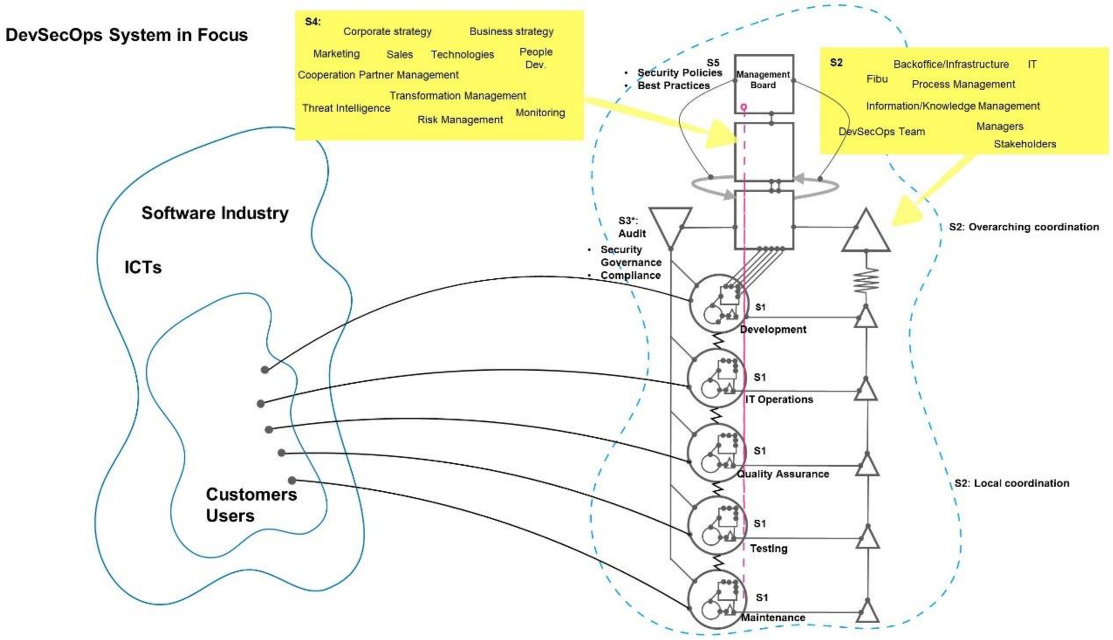


FIGURE 7. The DevSecOps viable system model in focus.

7.1 System 1 - Operations (Development, IT Operations, Quality Assurance):

System 1 in the context of DevSecOps includes the core activities of the development, IT operations, and quality assurance teams. These teams are responsible for developing, testing, deploying, and maintaining software applications. In a DevSecOps environment, these teams work collaboratively and iteratively to ensure continuous delivery of secure and high-quality software.

7.2 System 2 - Co-ordination (DevSecOps Team, Managers, Stakeholders):

System 2 represents the coordination subsystem responsible for integrating the activities of System 1. In the DevSecOps context, this includes the DevSecOps team, managers, and stakeholders who set the overall goals and vision for the software development process. They define the policies and procedures that promote the seamless integration of security practices into the development and operations workflows.

7.3 System 3 - Control (Security Governance, Compliance):

System 3 in the DevSecOps context involves security governance and compliance. This subsystem is responsible for monitoring and enforcing security policies and practices throughout the software development lifecycle. It ensures that security requirements are met, vulnerabilities are identified and addressed, and compliance with relevant regulations is maintained.

7.4 System 4 - Intelligence (Threat Intelligence, Risk Assessment):

System 4 focuses on intelligence gathering from the external environment. In DevSecOps, this includes activities such as threat intelligence and risk assessment. The intelligence subsystem continually monitors the ever-evolving security landscape, identifies potential threats, and assesses the risks posed to the software applications under development.

7.5 System 5 - Policy (Security Policies, Best Practices):

System 5 represents the policy subsystem, which sets the overall direction and guiding principles for security practices in DevSecOps. This includes defining security policies, best practices, and standards that all teams must adhere to during the development and operations process. System 5 ensures that security is embedded into the organizational culture and that security considerations are not an afterthought but a fundamental aspect of every stage of software delivery.

In this conceptualization, **DevSecOps is a viable system that allows organizations to build and deliver secure software efficiently. It fosters collaboration between development, IT operations, quality assurance, and security teams, leading to improved communication, quicker feedback loops, and more effective risk management**.

By applying the VSM to DevSecOps, organizations can:

- Improve **adaptability**: By continuously gathering intelligence on emerging security threats, organizations can adapt their security practices and respond proactively to new challenges.

- Enhance **communication** and **coordination**: The VSM framework highlights the importance of seamless communication and collaboration between different teams, which is a core principle of DevSecOps.
- Optimize **decision-making**: The model emphasizes the need for clear policies and best practices, which guide decision-making in security-related matters throughout the development lifecycle.
- Identify **areas of improvement**: By analyzing the functioning of each subsystem, organizations can identify gaps and weaknesses in their DevSecOps implementation and take corrective actions.
- Foster a **culture of security**: The VSM's policy subsystem ensures that security is ingrained in the organizational culture and becomes an integral part of software development and operations.

By conceptualizing DevSecOps within the Viable Systems Model, organizations can develop a more holistic understanding of the interplay between development, operations, and security functions. This can lead to more robust and resilient software delivery processes that are better equipped to handle the challenges of the modern cybersecurity landscape.

8. Connecting the Viable Systems Model (VSM) with the critical areas of DevSecOps:

System 1 - Operations (Development, IT Operations, Quality Assurance): the critical areas of DevSecOps involved are Continuous Integration (CI) and Continuous Deployment (CD). System 1 in VSM corresponds to the core activities of software development, IT operations, and quality assurance. In DevSecOps, the critical area of CI/CD focuses on automating the integration, testing, and deployment of code changes. This ensures that software updates are continuously delivered, allowing for rapid response to changing requirements and improvements in overall efficiency.

System 2 - Co-ordination (DevSecOps Team, Managers, Stakeholders): The critical area of DevSecOps involved is Cross-functional Collaboration. System 2 involves coordinating different functions within the organization. In DevSecOps, cross-functional collaboration is essential, as it brings together development, IT operations, security, and other stakeholders. Effective collaboration enhances communication, aligns goals, and ensures that security practices are integrated from the start of the development process.

System 3 - Control (Security Governance, Compliance): The critical areas of DevSecOps involved are Security Automation and Governance. System 3 in VSM deals with control and monitoring. In DevSecOps, the critical area of security automation and governance involves the use of automated security tools and processes to enforce security policies, identify vulnerabilities, and ensure compliance with security standards and regulations.

System 4 - Intelligence (Threat Intelligence, Risk Assessment): The critical areas of DevSecOps related are Threat Detection and Risk Management. System 4 focuses on gathering external intelligence to assess risks. In DevSecOps, the critical area of threat detection and risk management involves continuously monitoring the security landscape for potential threats and vulnerabilities. This helps in identifying and prioritizing security risks, allowing organizations to take proactive measures to mitigate them.

System 5 - Policy (Security Policies, Best Practices): The critical area of DevSecOps related is Secure Coding Practices. System 5 sets the policies and guiding principles. In DevSecOps, the critical area of secure coding practices involves establishing and enforcing coding standards that promote secure coding principles. This ensures that security is an integral part of the development process, reducing the likelihood of introducing vulnerabilities.

By connecting the Viable Systems Model to the critical areas of DevSecOps, organizations can gain a better understanding of how different aspects of their operations align with the principles of DevSecOps. This connection helps in identifying areas of improvement, enhancing communication and collaboration between teams, and establishing a more holistic approach to security and software delivery.

TABLE 2. Critical areas of DevSecOps vs. The Viable System Model

		The Viable System Model				
		System 1	**System 2**	**System 3**	**System 4**	**System 5**
		Operations	**Coordination**	**Control**	**Intelligence**	**Policy**
		Development, IT Operations, Quality Assurance	**DevSecOps Team, Managers, Stakeholders**	**Security Governance, Compliance**	**Threat Intelligence, Risk Assessment**	**Security Policies, Best Practices**
Critical Areas of DevSecOps	**Continuos Integration**	System 1 in VSM corresponds to the core activities of software development, IT operations, and quality assurance. In DevSecOps, the critical area of CI/CD focuses on automating the integration, testing, and deployment of code changes. This ensures that software updates are continuously delivered, allowing for rapid response to changing requirements and improvements in overall efficiency.				
	Continuous Deployment					
	Cross-functional Collaboration		System 2 involves coordinating different functions within the organization. In DevSecOps, cross-functional collaboration is essential, as it brings together development, IT operations, security, and other stakeholders. Effective collaboration enhances communication, aligns goals, and ensures that security practices are integrated from the start of the development process			
	Security Automation and Governance			System 3 in VSM deals with control and monitoring. In DevSecOps, the critical area of security automation and governance involves the use of automated security tools and processes to enforce security policies, identify vulnerabilities, and ensure compliance with security standards and regulations		
	Threat Detection and Risk Management				System 4 focuses on gathering external intelligence to assess risks. In DevSecOps, the critical area of threat detection and risk management involves continuously monitoring the security landscape for potential threats and vulnerabilities. This helps in identifying and prioritizing security risks, allowing organizations to take proactive measures to mitigate them.	
	Secure Coding Practices					System 5 sets the policies and guiding principles. In DevSecOps, the critical area of secure coding practices involves establishing and enforcing coding standards that promote secure coding principles. This ensures that security is an integral part of the development process, reducing the likelihood of introducing vulnerabilities

9. Second Discussion

Understanding DevSecOps through the lens of the Viable Systems Model (VSM) can bring several benefits to organizations that are seeking to integrate security practices seamlessly into their software development and operations processes. This approach offers a holistic perspective that considers the interconnections and interdependencies between different components of DevSecOps, leading to more effective security implementation and improved organizational viability. Somw of the benefits in more detail are described next:

Enhanced Communication and Collaboration: The VSM emphasizes the importance of communication and collaboration between different subsystems of an organization. By applying this model to DevSecOps, teams can better understand their roles and responsibilities in the security process. It encourages cross-functional collaboration between development, operations, quality assurance, and security teams, fostering a shared understanding of security requirements and objectives. This enhanced collaboration results in smoother workflows, quicker feedback loops, and ultimately, more secure software delivery.

Improved Adaptability to Changing Threat Landscapes: The VSM's focus on intelligence and risk assessment aligns well with the dynamic nature of cybersecurity threats. Understanding DevSecOps through this lens enables organizations to continually gather intelligence about emerging security threats and assess potential risks. By staying proactive and responsive to changing threat landscapes, DevSecOps teams can adjust their security practices accordingly, effectively mitigating risks and ensuring greater resilience against cyberattacks.

Optimal Resource Allocation: The VSM's control subsystem emphasizes monitoring and evaluating the performance of different subsystems. When applied to DevSecOps, this approach enables organizations to measure the effectiveness of their security practices and allocate resources more efficiently. By identifying potential bottlenecks or areas of improvement, organizations can make data-driven decisions to optimize their security efforts, ensuring that security is an integral part of the development process rather than an afterthought.

Consistent and Coherent Security Policies: System 5 in the VSM deals with policy and guiding principles. When applied to DevSecOps, this ensures that security policies and best practices are consistently applied throughout the software development lifecycle. By having a clear set of security policies, organizations can maintain a cohesive approach to security across different teams and projects, reducing the risk of overlooking critical security measures.

Enhanced Organizational Viability: The Viable Systems Model's primary objective is to ensure an organization's viability in a complex environment. When applied to DevSecOps, this approach helps organizations build a robust and secure software delivery pipeline. By integrating security early and continuously into the development and operations process, organizations can mitigate security risks and avoid potential threats. This, in turn, enhances the organization's overall viability by reducing the likelihood of security breaches, data leaks, or system downtime due to security vulnerabilities.

Holistic Approach to Security: Understanding DevSecOps through the VSM encourages a holistic approach to security. Rather than treating security as an isolated function, this perspective fosters a culture of security across all levels of the organization. It ensures that security is an inherent part of the organization's identity, leading to better-informed decision-making, increased security awareness, and a more resilient security posture.

Understanding DevSecOps through the lens of the Viable Systems Model brings numerous benefits to organizations seeking to enhance their security practices. By fostering collaboration, adaptability, and a holistic approach to security, organizations can effectively navigate the complexities of modern cybersecurity threats and maintain their viability in an ever-changing technological landscape. This approach empowers organizations to build and deliver secure software while simultaneously driving innovation and maintaining a competitive edge.

10. Conclusion

Based on the fact that knowing how to deploy and maintain DevSecOps, as a necessary evolution of DevOps, is an extremely important issue in the software development business today, in this work we proposed two things: 1) to identify what are the existing professional competency frameworks that could guide in the process of knowing the skills in the software industry, and 2) to identify what are the skills identified by the industry that can ensure the proper deployment of each area of DevSecOps.

Firstly, a literature review has been carried out to find out the status of this issue. We found that although there are great efforts to identify human and technological aspects that are critical for DevSecOps, no specific framework has been defined for this issue. To discover this, we reviewed four competency frameworks (the Software Engineering Competency Model (SWECOM), the US IT Competency Model, the Skills Framework for the Information Age (SFIA 8), and the People Capability Maturity Model (P-CMM).) and assessed their comprehensiveness in order to meet the needs of DevSecOps. Given that the SFIA framework appears to be the broadest, and encompasses the skills identified by the other competency frameworks, it was selected for further mapping to the critical areas of DevSecOps.

A formal and iterative mapping was made between the SFIA competencies and the critical areas of DevSecOps in order to identify those that have an impact on the functioning of each area. To do this, each possibly relevant skill was assessed as to whether it was essential, desirable, or dispensable, based on a series of questions that led to assessing it. From this mapping, the essential SFIA skills that would contribute to the functioning of DevSecOps were identified, namely:Agile (AGIL), Analytics and Metrics (ANMT), Automation (AUTO), Build and Release Management (BUAN), Business Analysis (BUAN), Change Management (CHMG), Cloud Computing (CLCO), Communication (COMN), Configuration Management (CFMG), Data Analysis (DTAN), DevOps (DEVO), Incident Management (USUP), Infrastructure Management (ITMG), Performance Analysis (PPER), Process Improvement (PROI), Relationship Management

(RLMT), Release and Deployment Management (RELM), Risk Management (RISM), Security (SCTY), Service Level Management (SLMO), Software Development (SWDN), Solution Architecture (ARCH), and Testing (TEST). It was subsequently justified why each of these skills is critical in the areas of DevSecOps in which it impacts.

Regarding the Viable System Model as a means to improve DevSecOps, understanding DevSecOps through the lens of the Viable Systems Model brings numerous benefits to organizations seeking to enhance their security practices. Among then identified and explained benefits, it could be mentioned: Enhanced Communication and Collaboration, Improved Adaptability to Changing Threat Landscapes, Optimal Resource Allocation, Consistent and Coherent Security Policies, Enhanced Organizational Viability, Holistic Approach to Security.

The need to continue this work to further consider the organizational, human and feasibility characteristics that are also necessary to ensure the successful deployment of DevSecOps is also raised, thus completing a general, systemic, referential framework of the competencies needed to support the successful deployment of DevSecOps and the evolving challenges it reveals.

11. References

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*. https://doi.org/10.1016/j.infsof.2022.106894

Alarifi, A., Zarour, M., Alomar, N., Alshaikh, Z., & Alsaleh, M. (2016). SECDEP: Software engineering curricula development and evaluation process using SWEBOK. *Information and Software Technology*, *74*, 114–126. https://doi.org/10.1016/j.infsof.2016.01.013

Andrade Sosa, H. H., Dyner R., I., Espinosa, A., López Garay, H., & Sotaquirá, R. (2007). *Pensamiento Sistémico: Diversidad en búsqueda de Unidad*. Ediciones Universidad Industrial de Santander.

Beer, S. (1964). *Cybernetics and Management*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Beer, S. (1972a). *Brain of the firm*. Allen lane The penguin press.

Beer, S. (1972b). *Brain of the firm*. Allen lane The penguin press.

Beer, S. (1984). The Viable System Model : its provenance , development , methodology and pathology. *Journal of the Operational Research Society*, *35*, 7–26.

Beer, S. (1985). *Diagnosing The System for Organizations*. John Wiley & Sons, Inc.

Burgess, A., Kelly, A. M., Butterfield, E. M., Keppler, J., McClenahan, D., Guillemette, K., & Phon, M. (2014). *Software Engineering Competency Model (SWECOM)*. IEEE Computer Society.

Capozucca, A., & Guelfi, N. (2020). Analysing the swecom standard for designing a devops education programme. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12271 LNCS*, 133–150. https://doi.org/10.1007/978-3-030-57663-9_10

CC2020 Task Force. (2020). Computing Curricula 2020. In *Computing Curricula 2020*. ACM. https://doi.org/10.1145/3467967

Clear, T. (2017). Meeting Employers Expectations of DevOps Roles: Can Dispositions Be Taught? *ACM Inroads*, *8*(2), 19–21. https://doi.org/10.1145/3078298

Curtis, B., Hefley, W. E., Sally, Q.-L., & Miller, A. (2001). *People Capability Maturity Model ® (P-CMM ®)*.

Espejo, R., & Harnden, R. (1990). The viable system model: interpretations of Stafford Beer's VSM: John Wiley & Sons, 1989, £24.95, xi + 472 pages. *European Journal of Operational Research*, *44*. https://doi.org/10.1016/0377-2217(90)90368-L

Espejo, R., & Reyes, A. (2011). Organizational Systems: Managing Complexity with the Viable System Model. In *Springer*. https://doi.org/10.1007/978-3-642-19109-1

Espinosa, A. (2023). *Sustainable Self-Governance in Business and Society*. Routledge.

Frezza, S., Daniels, M., Pears, A., Cajander, Å., Kann, V., Kapoor, A., McDermott, R., Peters, A. K., Sabin, M., & Wallace, C. (2018). Modelling competencies for computing education beyond 2020: A research based approach to defining competencies in the computing disciplines. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 148–174. https://doi.org/10.1145/3293881.3295782

IEEE-CS Professional & Educational Activities Board - SWEBOK Evolution Team. (2023). *Software Engineering Body of Knowledge (SWEBOK) - V4 Beta (v2022Dec31)*. https://waseda.box.com/s/elnhhnezdycn2q2zp4fe0f2t1fvse5rn

Impagliazzo, J., Bourque, P., & Mead, N. R. (2020). Incorporating CC2020 and SWECOM Competencies into Software Engineering Curricula: A Tutorial. *32nd IEEE Intl. Conference on Software Engineering Education & Training*.

Information Technology Competency Model. (2021). United States Department of Labor. https://www.careeronestop.org/competencymodel/competency-models/information-technology.aspx

Johnson, M. W., & Leydesdorff, L. (2013). Beer's Viable System Model and Luhmann's Communication Theory: Organizations from the Perspective of Meta-games. *Journal of Systems Research and Behavioural Science*, *282*(August 2013), 266–282. https://doi.org/10.1002/sres.2222

Lee, J. S. (2018). The DevSecOps and Agency Theory. *Proceedings - 29th IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2018*, 243–244. https://doi.org/10.1109/ISSREW.2018.00013

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. In *ACM Computing Surveys* (Vol. 52, Issue 6). Association for Computing Machinery. https://doi.org/10.1145/3359981

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: A multivocal literature review. *Communications in Computer and Information Science*, *770*, 17–29. https://doi.org/10.1007/978-3-319-67383-7_2

Ormond, J. (2021). *ACM AND IEEE-CS RELEASE COMPUTING CURRICULA 2020, GLOBAL GUIDELINES FOR BACCALAUREATE DEGREES IN COMPUTING*. https://www.acm.org/media-center/2021/march/computing-curricula-2020

Sánchez-Gordón, M., & Colomo-Palacios, R. (2020). Security as Culture: A Systematic Literature Review of DevSecOps. *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*, 266–269. https://doi.org/10.1145/3387940.3392233

SFIA Foundation. (2021). *Skills Framework for the Information Age - SFIA 8*. www.sfia-online.org

Yasar, H. (2020). *Overcoming DevSecOps Challenges: A Practical Guide for All Stakeholders*.

Yasar, H., & Yankel, J. (2023). *Top 5 Challenges to Overcome on Your DevSecOps Journey*. https://resources.sei.cmu.edu/asset_files/Webinar/2023_018_101_978710.pdf