Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2025, № 4 (120) https://doi.org/10.33108/visnyk_tntu2025.04
ISSN 2522-4433. Web: ***visnyk.tntu.edu.ua***

UDC 004.056.55:004.4.233

APPLYING THE SAFE METHODOLOGY TO INTEGRATE CYBERSECURITY IN LARGE-SCALE IT PROJECTS

Mariia Stadnyk

Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

Abstract. *This study examines the integration of cybersecurity practices into the Scaled Agile Framework (SAFe) as a structured approach for securing large-scale IT projects. The research analyzes how DevSecOps activities–such as SAST, DAST, SCA, container scanning, and configuration control–enhance the security of the Continuous Delivery Pipeline by enabling continuous vulnerability detection and reducing human-factor risks. Threat modeling methods, including STRIDE, PASTA, and LINDDUN, are evaluated for their effectiveness in identifying security risks at early design stages and informing architectural decisions. The study also highlights the role of Zero Trust principles, Architecture Decision Records, and Security Enablers in ensuring resilient system architecture. Additional mechanisms, such as Security Backlog Items, enhanced Definition of Done criteria, and compliance tasks aligned with ISO/IEC 27001, GDPR, PCI DSS, and HIPAA, were shown to support regulatory adherence. The involvement of Security Champions significantly improves communication between development teams and security experts, fostering a stronger security culture. Overall, the findings demonstrate that SAFe provides a comprehensive foundation for integrating cybersecurity across organizational levels, thereby improving product reliability and operational resilience.*

Key words: *Agile, SAFe, cybersecurity, DevSecOps, Built-In Quality.*

Submitted 01.09.2025
Revised 28.11.2025
https://doi.org/10.33108/visnyk_tntu2025.04. 099 *Published 27.01.2026*

1. INTRODUCTION

In the modern era of digital transformation, cybersecurity has become one of the key factors determining the success of IT projects regardless of organizational scale. According to the global Cybersecurity Ventures Report (2023), the number of cyber incidents worldwide increases by 15% annually, and the total damage from cyberattacks is expected to reach USD 10.5 trillion by 2025 [1]. Furthermore, Gartner analysts (2024) report that more than 72% of large-scale IT projects using agile methodologies include a dedicated cybersecurity component: DevSecOps, Secure SDLC, or integrated Security-by-Design practices [2].

In 2015, only 16% of Agile projects included mandatory security activities, in 2019 this figure reached 32%, and by 2023 the number increased to over 58%, according to the State of Agile Report by VersionOne [3]. In large-scale frameworks such as SAFe, the proportion of projects involving cybersecurity exceeds 75%, as large program-level and platform solutions require comprehensive protection of architecture, APIs, data, and infrastructure.

Since 2022, the importance of cybersecurity has risen sharply for the Ukrainian IT sector. According to the State Service of Special Communications and Information Protection of Ukraine, the number of cyberattacks targeting Ukrainian companies and government institutions increased more than fivefold between 2022 and 2024 [4]. Additionally, the *ESET Threat Report 2024* indicates that Ukraine is among the top five countries worldwide in terms of targeted cyberattacks against IT systems, which significantly amplifies the need for security measures embedded at the development stage [5].

For Ukrainian IT companies, cybersecurity is not only a technical requirement but also a strategic business factor. More than 68% of Ukrainian outsourcing vendors work with clients

in industries with strict security standards – financial services, telecommunications, defense technologies, e-commerce, and healthcare [6]. Therefore, ensuring compliance with international regulations, such as ISO/IEC 27001, NIST 800-series, GDPR, and PCI DSS, is essential. This makes the integration of Security-by-Design and DevSecOps into scalable development frameworks, particularly SAFe, critically important.

According to Deloitte Cyber 2024, 62% of global organizations face challenges in implementing security in Agile processes due to the lack of centralized methodology and unified standards [7]. As a highly structured and hierarchically scalable Agile framework, SAFe provides an optimal platform for integrating cybersecurity at all levels – from Portfolio to Team. It enables a systematic approach to security, including DevSecOps, automation, threat modeling, and compliance management.

Thus, in the context of rapidly growing cyber threats, active digitalization, wartime conditions, and increasing compliance demands from international stakeholders, integrating cybersecurity into scalable development models becomes a strategic necessity for the Ukrainian IT industry. The use of SAFe enables organizations to adopt a structured Security-by-Design approach, implement security practices at every stage of the product lifecycle, and strengthen the global competitiveness of Ukrainian IT vendors.

2. ANALYSIS OF EXISTING RESEARCH

Within the study of integrating cybersecurity into scalable Agile methodologies, particularly SAFe, a range of recent scientific and practical works has been reviewed. The following section presents a structured analysis of key studies that form the theoretical foundation of the topic.

The study «Integrating Security into Agile Software Development Processes» [8] highlights the inconsistency between traditional cybersecurity approaches and the rapid iteration cycles of Agile. The authors propose a model for early inclusion of security activities into the backlog, including threat modeling and continuous risk assessment. The findings show that early integration of security significantly reduces the number of vulnerabilities at later development stages.

The work «How to Integrate Security Compliance Requirements with Agile Software Engineering at Scale?» [9] analyzes the issue of integrating compliance requirements (GDPR, ISO/IEC 27001) into large-scale Agile environments. The authors demonstrate that in SAFe, regulatory requirements should be incorporated at the Portfolio and ART levels, and that automating compliance checks significantly reduces release delays. The authors proposed the S2C-SAFe model that describes how the requirements of the IEC 62443-4-1 standard can be integrated into the SAFe methodology by aligning the roles, processes, and artifacts of both approaches. The article shows that security practices, such as defining security requirements, secure implementation, and security verification, can be naturally embedded into standard SAFe workflows and positioned within the Continuous Delivery Pipeline. The authors emphasize that this integration provides development teams and security experts with a shared operational model that enables coordinated risk management and ensures regulatory compliance in large-scale Agile environments.

The practical aspect of implementing security within SAFe is presented in «Using SAFe to Align Cyber Security and Executive Goals» [10], which describes the real-world experience of a large international organization. The study shows that involving Security Architects and Security Champions in PI Planning improves the alignment between business objectives and security requirements.

The article «Scaled Agile Framework (SAFe) for Cybersecurity Teams» [11] focuses on adapting SAFe to the needs of cybersecurity teams. The authors describe the structure of

roles, including Security Engineer and Product Owner Security, and provide examples of security stories and Definition of Done criteria for security-related tasks.

In the study «DevSecOps: Integrating Security into DevOps» [12], an in-depth review of DevSecOps models and automated security-testing tools such as SAST, DAST, IAST, and dependency scanning is presented. The work demonstrates the importance of automation as a key element of the SAFe Continuous Delivery Pipeline. The systematic review «Secure DevOps: A Systematic Literature Review» [13] covers more than 150 publications and presents a consolidated Secure DevOps model that includes monitoring, secrets management, and automated control processes. The findings correlate with the principles of SAFe DevSecOps.

The article «Built-in Security in Agile Projects: Challenges and Solutions» [14] examines the principle of Built-in Quality, which is one of the foundational components of SAFe. The authors propose incorporating security acceptance criteria into the Definition of Ready and Definition of Done, as well as using regular threat modeling in every iteration.

The empirical study «Security Activities in Scaled Agile: An Empirical Study» [15] analyzes the practices of 12 large companies using SAFe. The results show that the most effective mechanisms include involving security specialists during backlog refinement and having a Security Champion embedded in each Scrum team within the ART. The work «Threat Modeling in Agile and DevOps» [16] provides an overview of STRIDE, LINDDUN, and PASTA approaches and highlights methods for automating threat modeling. The article demonstrates how to adapt these approaches to the rapid development cycles typical of SAFe and DevOps workflows.

The study «Continuous Security: Automating Secure Software Delivery» [17] describes a continuous security model within CI/CD, including dependency control, container security, and secrets management. The authors emphasize the relevance of automation for large SAFe organizations.

The article «Security Challenges in Large-Scale Agile Development» [18] examines common security problems in large Agile programs, including difficulties in coordinating security requirements among multiple teams. The study proposes mechanisms for synchronizing security activities at the ART and Solution Train levels. The official document NIST SP 800-207 «Zero Trust Architecture» [19] outlines the core principles of Zero Trust, which are highly relevant for designing secure architecture in large-scale SAFe systems. It emphasizes continuous access control, network segmentation, and minimal trust.

The study «Security-by-Design: A Comprehensive Survey» [20] provides a detailed analysis of Security-by-Design models and design patterns. The authors show that embedded security must be applied from the earliest planning phases, which aligns with the SAFe Portfolio and Solution levels. The article «Agile DevSecOps for Cloud-Native Applications» [21] examines the integration of DevSecOps into cloud-native environments. It addresses specific risks related to containerization, Kubernetes, secrets management, and microservices, and offers architectural solutions compatible with SAFe DevOps practices. The work «Security Automation in CI/CD Pipelines» [22] analyzes methods of automated vulnerability detection, dependency scanning, and configuration control. The authors propose integrating security automation into the SAFe Continuous Delivery Pipeline to reduce human error and accelerate incident response.

This study by Stadnyk and Palamar [23] provides valuable insights into the specific project management practices required in cybersecurity-oriented environments, highlighting the need for structured processes, risk-driven planning, and compliance-focused coordination. Their findings complement the principles discussed in this article, reinforcing the importance of integrating security governance and systematic management approaches into large-scale Agile frameworks such as SAFe.

Overall, the analysis of the reviewed sources shows that the integration of cybersecurity into Agile, DevOps, and SAFe is an active area of scientific and practical research. A general trend is observed: a shift from «post-development security verification» toward Security-by-Design, implemented through DevSecOps, automation, threat modeling, and architectural governance. This aligns fully with the principles of SAFe and confirms the relevance of the chosen research topic.

3. SAFe FRAMEWORK IN A NUTSHELL

The Scaled Agile Framework (SAFe) is a comprehensive, hierarchical model designed to coordinate Agile practices across enterprises developing large, complex software systems. It extends traditional Agile principles with Lean thinking, system-level governance, synchronized planning cycles, and multi-team collaboration mechanisms. SAFe addresses the limitations of single-team Agile by providing structure, alignment, and architectural integrity across hundreds of engineers working in distributed environments.

At its core, SAFe integrates Scrum, Kanban, Lean Product Development, enterprise architecture, and DevOps into a unified system of roles, events, and artifacts. It organizes work into four interconnected levels: Team, Program (Agile Release Train), Large Solution, and Portfolio (Fig.1). Each level is responsible for specific decision-making, planning horizons, and delivery outcomes.

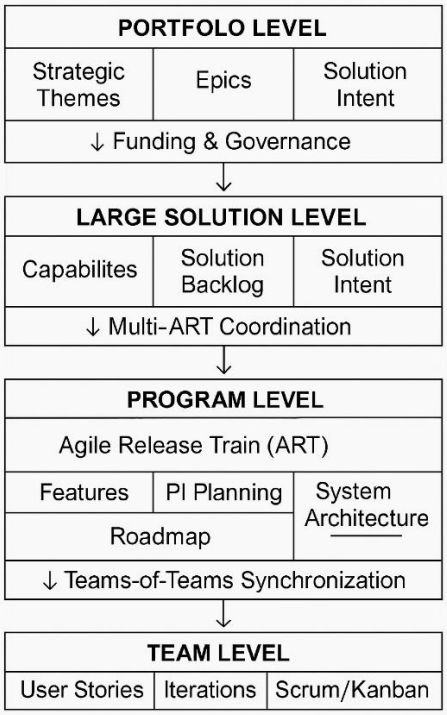

Figure 1. The main levels of the Scaled Agile Framework (SAFe) and their key artifacts

The Scaled Agile Framework (SAFe) is structured across several interconnected levels, each of which plays a distinct role in coordinating large-scale Agile development. At the Portfolio Level**,** SAFe links the organization's long-term business strategy with the development portfolio. The central concept at this level is the Strategic Theme, which represents the high-level business priorities that guide decisions on what types of solutions should be developed. Another key term is the Epic, a large-scale business initiative that requires substantial funding, cross-organizational coordination, and multiple iterations to implement. To

ensure effective financial governance, SAFe uses Lean Budgets, a system that allocates funding to value streams rather than individual projects, allowing teams to innovate without bureaucratic delays. The primary purpose of the Portfolio Level is to maintain alignment between enterprise strategy and the work delivered by development teams, ensuring that all initiatives contribute to measurable business outcomes.

The Large Solution Level appears when a product or system is so complex that it requires coordination across multiple Agile Release Trains (ARTs). This level is responsible for managing large-scale system development that cannot be handled by a single ART. Here, work is organized around Capabilities, which are high-level solution behaviors that span several teams and release trains. These capabilities are stored in the Solution Backlog, a repository of upcoming work items for large solutions. A vital concept at this level is the Solution Intent, which contains the evolving understanding of the system's requirements, architecture, constraints, and compliance needs. The purpose of the Large Solution Level is to synchronize the work of multiple trains, maintain a consistent architectural direction, and ensure that every component of a large system integrates correctly and safely.

The Program Level, built around the Agile Release Train (ART), forms the operational backbone of SAFe. An ART is a long-lived, cross-functional team-of-teams typically consisting of 50 to 125 people who plan, develop, test, and deliver value together. At this level, work is defined in terms of Features, which represent customer-centric functionalities that can be delivered within a single Program Increment. These increments, known as PIs, are fixed time periods (usually 8–12 weeks) during which all teams within the ART plan and commit to delivering specific outcomes. A crucial planning event here is PI Planning, where all members of the ART synchronize their goals, identify dependencies, and create an integrated plan. Another key role is the System Architect, who ensures technical alignment across teams and maintains a coherent architectural direction. The purpose of the Program Level is to establish predictable delivery cycles, foster cross-team collaboration, and ensure that all teams work in a synchronized cadence.

Finally, the Team Level represents the smallest operational unit in SAFe, where individual Scrum or Kanban teams carry out iterative development. Teams work with User Stories, which break down features into small, actionable units of customer value that can be implemented within short timeboxed iterations. Through Iteration Planning, teams decide what work they can complete during the upcoming cycle, typically lasting one or two weeks. Depending on the preferred workflow, teams may operate using Scrum, which emphasizes timeboxed sprints, or Kanban, which focuses on continuous flow and limiting work in progress. The purpose of the Team Level is to transform high-level program objectives into working product increments while maintaining quality and responding quickly to feedback.

Across all four levels, SAFe establishes a coherent and scalable structure that ensures alignment between business objectives and technical execution. Each level contributes to transparency, faster delivery, and reduced complexity, ultimately enabling enterprises to develop solutions that are both strategically relevant and technically sound.

4. THEORETICAL FOUNDATIONS OF CYBERSECURITY INTEGRATION IN SAFe

The integration of cybersecurity into the Scaled Agile Framework (SAFe) is based on the combination of Lean, Agile, DevOps, and systems thinking concepts, which makes it possible to incorporate security requirements into all stages of the software development lifecycle. SAFe does not view security as a separate phase or an external process but rather as an integral component of the value created by the team. This approach reflects the principle of Security-by-Design, which involves addressing potential threats, vulnerabilities, and protective mechanisms in the early stages of development.

At the theoretical foundation of SAFe lies the concept of Built-In Quality, a fundamental principle stating that quality, including cybersecurity, must be embedded directly into the process of product creation. In this context, security becomes an inherent attribute of each product increment rather than the result of an isolated audit. Development teams, analysts, DevOps engineers, and architects share collective responsibility for ensuring the security of the solution.

The Built-In Quality component in SAFe encompasses a set of interconnected aspects that ensure the integration of quality and cybersecurity directly into the software development process (Table 1). One of its key elements is Flow, which defines the ability of teams and the Agile Release Train (ART) to maintain a stable, continuous, and predictable stream of value delivery. In the context of cybersecurity, this means that all security checks are incorporated into every stage of the CI/CD pipeline, and the pipeline itself operates in a manner that prevents vulnerabilities from entering the production environment. Security in SAFe is viewed not as a separate manual activity but as an inherent part of the automated delivery process, where systems continuously perform code scanning, dependency analysis, container security checks, and secret leakage detection. Without such an approach, security activities are shifted to the end of the development cycle, which leads to significant release delays and substantially increases financial and operational risks.

Table 1

Integrated Cybersecurity activity into Built-In Quality Component

Built-In Quality Component	Integrated Cybersecurity activity
Flow	Automated code scanning at the commit stage, automated dependency checking (SCA), container scanning (for Docker/Kubernetes environments), secret scanning (detecting exposed keys), secure build environments (no secrets stored in the pipeline), rapid feedback for developers
Architecture and Design Quality	Threat modeling (STRIDE, PASTA, LINDDUN), zero trust architecture (NIST 800-207), secure design patterns such as data encryption and service isolation, network segmentation, the principle of least privilege, architecture runway (preparation of architectural elements in advance), compliance constraints (GDPR, PCI DSS, ISO 27001)
Code Quality	Static application security testing, dynamic application security testing, interactive application security testing, unit tests and security unit tests, peer review or code review, linting and coding standards
System Quality	Penetration testing, integration testing, performance and load testing, security regression testing, chaos engineering

The second aspect of Built-In Quality is Architecture and Design Quality, which refers to the system's ability to withstand potential cyberattacks and ensure security by default. Architectural decisions in SAFe must be modular, predictable, and resilient, which is achieved through the application of threat modeling techniques such as STRIDE, PASTA, and LINDDUN, as well as the adoption of Zero Trust Architecture principles in accordance with NIST guidelines. Secure design patterns, including data encryption, network segmentation, and the principle of least privilege, also play an important role. In complex distributed systems, such as microservice architectures, these measures ensure that even if one component is compromised, it does not lead to a breach of the entire system. Consequently, weaknesses in architectural decisions may introduce systemic vulnerabilities that cannot be resolved through simple technical fixes.

Another essential element is Code Quality, which determines the degree of safety and reliability of the software code. In SAFe, this aspect requires that errors, defects, and vulnerabilities be identified before the code is integrated into the main branch. To achieve this, static (SAST) and dynamic (DAST) testing methods are applied, along with interactive analysis (IAST), which combines the advantages of both approaches. In addition to automated tools, expert evaluation in the form of code review and peer review plays a significant role, enabling the timely detection of logical flaws and design weaknesses. This is complemented by modular and security-related unit tests, as well as adherence to corporate coding standards. Since the overwhelming majority of vulnerabilities originate at the code level, their early detection substantially reduces remediation costs and minimizes the likelihood of threats propagating to higher layers of the system.

The final aspect, System Quality, focuses on the comprehensive evaluation of the system as a whole, including its security, performance, and reliability. In the context of cybersecurity, this involves conducting penetration testing, which enables the simulation of real-world attacks in order to identify weaknesses that may have been overlooked by automated static or dynamic analysis tools. Additional activities include integration testing to verify the correct interaction of system components; performance and load testing to prevent failures caused by overload; and security regression testing to ensure that newly introduced changes do not compromise existing protective mechanisms. In more complex environments, methods of Chaos Engineering are also applied to assess the system's resilience to unexpected failures. The importance of this aspect lies in the fact that even if individual components are secure, the system as a whole may remain vulnerable due to incorrect integration or misconfiguration.

Another important theoretical aspect is Lean Thinking, which aims to eliminate waste and maximize value. In the context of security, this means reducing rework associated with identifying vulnerabilities at later stages and creating conditions for early detection of risks. The Lean-oriented approach involves the use of short inspection and adaptation cycles (Inspect & Adapt), enabling teams to regularly reassess and refine the security aspects of the product.

SAFe also relies on systems thinking, which considers a product as a complex system composed of interconnected components. In the cybersecurity context, systems thinking makes it possible to understand how changes in one module can influence the overall architectural security, as well as to develop threat models that account for the interactions between subsystems.

Furthermore, SAFe incorporates the requirements of regulatory standards such as ISO/IEC 27001, GDPR, and NIST SP 800-207 (Zero Trust), enabling organizations to ensure both technical and regulatory compliance. At the portfolio level, these requirements are integrated into the Strategic Themes, while at the program level they are reflected in specific features, architecture enablers, and PI objectives.

5. TOOLS FOR INTEGRATING CYBERSECURITY INTO SAFe

The integration of cybersecurity into SAFe relies on a structured toolkit that supports secure development, architecture, deployment, and compliance across all organizational levels. These tools enable continuous verification of security requirements, automated vulnerability detection, and systematic risk mitigation. As a result, SAFe teams can deliver secure and trustworthy solutions at scale.

DevSecOps in SAFe is one of the core mechanisms that enables continuous integration of cybersecurity across the entire product lifecycle. Unlike traditional security models, where security checks are performed only at the final stages, DevSecOps embeds security activities throughout development, testing, deployment, and operations. In SAFe, DevSecOps includes automated security testing such as SAST (Static Application Security Testing) for source-code

analysis, DAST (Dynamic Application Security Testing) for assessing the behavior of running applications, and SCA (Software Composition Analysis) for monitoring third-party dependencies and identifying vulnerabilities in external libraries.

A critical component of this approach is container scanning and configuration control, which help detect insecure Docker image settings and misconfigured Kubernetes environments. Integrated secret management mechanisms, covering API keys, tokens, certificates, and other sensitive assets, ensure that confidential information is protected from leakage within the CI/CD pipeline.

The SAFe Continuous Delivery Pipeline consists of four interdependent stages: Continuous Exploration, Continuous Integration, Continuous Deployment, and Release on Demand. Each of these serves as a security integration point. During Continuous Exploration, security requirements are identified and analyzed; Continuous Integration incorporates automated security checks and artifact validation; Continuous Deployment verifies configuration policies and enforces secure deployment practices; and Release on Demand ensures that releases are executed only when all security criteria have been satisfied. This continuous model reduces human error and enables early detection of vulnerabilities, improving both efficiency and system resilience.

Threat Modeling in SAFe is a fundamental technique for anticipating and mitigating risks early in the development process. Using structured methodologies such as STRIDE, PASTA, and LINDDUN, teams systematically identify threats ranging from confidentiality breaches to complex business-logic attacks. Early threat modeling allows teams to evaluate the criticality of potential attack scenarios and define appropriate mitigation strategies. The results of this analysis are documented in the Solution Intent, an artifact that captures functional and non-functional requirements along with associated security constraints.

The security architecture within SAFe is grounded in concepts such as Zero Trust, which assumes no implicit trust between system components and requires continuous verification of all interactions. Architects apply instruments such as Architecture Decision Records (ADRs) to document strategic decisions, Security Enablers to introduce architectural improvements, Compliance Constraints to ensure alignment with regulatory obligations, and Architecture Epics to manage large-scale security initiatives. At the Program and Large Solution levels, these tools establish a unified governance structure that ensures architectural consistency and minimizes systemic vulnerabilities.

SAFe incorporates dedicated Security Backlog Items as part of a structured approach to managing security-related work. These items may include security user stories focused on safeguarding system and user interactions, threat-mitigation tasks aimed at addressing identified risks, architectural enablers required to strengthen the security posture, and compliance tasks mapped to regulatory frameworks such as GDPR, ISO/IEC 27001, or PCI DSS.

The Definition of Done (DoD) is expanded with specific security criteria to ensure that each product increment meets established security requirements. These include mandatory automated security tests, verification of third-party dependencies for vulnerabilities, updating libraries to secure versions, conforming to coding standards, and validating cryptographic policies. This guarantees that no increment can be considered complete without meeting all required security conditions.

A Security Champion is a designated team member who acts as a local security expert and facilitator between the development team and cybersecurity specialists. This individual understands core security standards, performs initial threat modeling, monitors adherence to secure coding practices, supports the integration of DevSecOps tools into the pipeline, and provides ongoing training and guidance to team members.

The role is particularly crucial in large organizations, where an Agile Release Train (ART) may include numerous teams, making it impractical for security experts to be directly involved in

every process. Security Champions help bridge this gap, ensuring timely risk escalation, faster remediation, and the cultivation of a strong security culture within development teams.

SAFe incorporates regulatory compliance processes through the Compliance Backlog – a centralized repository of requirements derived from standards such as GDPR, PCI DSS, HIPAA, and ISO/IEC 27001. At the Portfolio level, Lean Governance ensures systematic oversight through regular audits, policy enforcement, and risk management. At the Program level, compliance enablers facilitate technical implementation of regulatory requirements into the product architecture.

Teams leverage specialized automation tools to support these activities, including Open Policy Agent (OPA) and Kyverno for enforcing security policies in Kubernetes environments, audit trails for recording all critical system operations, and security dashboards for real-time monitoring of security posture and incidents. Consequently, compliance is not treated as a separate manual process but is integrated into daily development and operational workflows.

The table 2 below provides a structured overview of all key tools used to integrate cybersecurity into SAFe, SAFe level and cybersecurity alignment.

Table 2

Alignment of SAFe Cybersecurity Practices with International Standards

SAFe Security Aspect	Industry Standard	SAFe Level	Alignment / Explanation
DevSecOps in the CI/CD Pipeline	ISO/IEC 27001, NIST SP 800-53, OWASP SAMM	Team, Program	Ensures secure development, change control, and automated vulnerability detection
Threat Modeling in SAFe	NIST SP 800-154, ISO/IEC 27005, OWASP ASVS	Program, Large Solution	Provides systematic threat analysis and risk documentation
Security Architecture (Zero Trust)	NIST SP 800-207, CSA CCM	Large Solution, Portfolio	Based on the Zero Trust principle («never trust, always verify»)
Security Backlog Items	ISO/IEC 27034-1	Team, Program	Integrates security requirements into functional development activities
Definition of Done (security criteria)	OWASP ASVS, NIST Secure Coding Guidelines	Team	Ensures that each increment meets secure coding and testing requirements
Security Champions	NIST NICE Framework, BSIMM	Team	Enhances team security competence and accelerates issue escalation
Compliance Backlog and Audits	GDPR, PCI DSS, HIPAA, ISO/IEC 27001	Program, Portfolio	Ensures organizational alignment with regulatory standards
Monitoring and Observability	NIST SP 800-137	Program, Large Solution	Enables early incident detection and real-time visibility into security posture

The table below systematizes the key cybersecurity practices embedded within the SAFe framework and demonstrates their alignment with leading international security standards. It also indicates the specific SAFe levels at which each practice is applied, providing a clear understanding of how cybersecurity is integrated across the organizational structure.

6. CONCLUSIONS

1. The research confirms that effective cybersecurity integration in SAFe requires embedding security practices across all hierarchical levels, ensuring that protection mechanisms become an inherent part of the development lifecycle rather than a separate stage.

2. DevSecOps practices, including automated security testing, dependency scanning, configuration control, and secret management, significantly improve the security posture of large-scale IT projects by enabling early detection and mitigation of vulnerabilities.

3. Threat modeling, Zero Trust–based architectural governance, and structured security enablers are essential for identifying systemic risks at early stages and maintaining architectural resilience in complex distributed environments.

4. The use of Security Backlogs, enhanced Definition of Done criteria, and the involvement of Security Champions ensures consistent implementation of security requirements, regulatory compliance, and rapid response to emerging risks within Agile Release Trains.

References

1. Cybersecurity Ventures (2023). 2023 Official Cybercrime Report: Cybersecurity market data, insights & statistics. https://cybersecurityventures.com/cybercrime-report/.
2. Gartner (2024). Gartner forecast: Security and risk management trends in agile and DevSecOps. Gartner Research. https://www.gartner.com.
3. VersionOne (2023). 17th Annual State of Agile Report. Digital.ai. https://digital.ai/resources/state-of-agile-report.
4. State Service of Special Communications and Information Protection of Ukraine (2024). Annual cybersecurity report of Ukraine 2022–2024. https://cip.gov.ua.
5. ESET (2024). ESET Threat Report 2024: Global trends in cyberattacks. ESET Research. https://www.eset.com/int/security-report.
6. IT Ukraine Association (2023). Ukrainian IT industry report: Outsourcing market overview and security compliance requirements. https://itukraine.org.ua.
7. Deloitte (2024). Deloitte Cyber Report 2024: Global challenges in integrating security into Agile and DevSecOps. Deloitte Insights. https://www2.deloitte.com.
8. Shahid J., Hameed M. K., Javed I. T., Qureshi K. N., Ali M., & Crespi N. (2020). Integrating security into agile software development processes. IEEE. https://doi.org/10.1109/ACCESS.2020.2968524
9. Moyon F., Mendez Fernandez D., Beckers K., Klepper S. (2021). How to integrate security compliance requirements with agile software engineering at scale? arXiv preprint arXiv:2105.13404. https://doi.org/10.48550/arXiv.2105.13404.
10. WithSecure (2022). Using SAFe to align cyber security and executive goals. https://www.withsecure.com/content/dam/withsecure/global/en/white-papers/using-safe-to-align-cyber-security-and-executive-goals.pdf.
11. LarkSuite (n.d.). Scaled Agile Framework (SAFe) for cybersecurity teams. https://www.larksuite.com/en_us/static/docs/safe_cybersecurity.pdf.
12. Aljuneidi A., et al. (2021). DevSecOps: Integrating security into DevOps. ACM Computing Surveys. https://doi.org/10.1145/3453151.
13. Kaur A., & Chatterjee I. (2020) Secure DevOps: A systematic literature review. Information and Software Technology, 130. https://doi.org/10.1016/j.infsof.2020.106412
14. Chehaba A., et al. (2019). Built-in security in agile projects: Challenges and solutions. Springer. https://doi.org/10.1007/978-3-030-06019-0_18.
15. Sabaliauskaite G., et al. (2022). Security activities in scaled agile: An empirical study. ICSOB Conference. https://doi.org/10.1007/978-3-031-07245-3_10.
16. Shostack A. (2021). Threat modeling in Agile and DevOps. Microsoft Research. https://doi.org/10.48550/arXiv.2106.13353.
17. DevOps Institute (2020). Continuous security: Automating secure software delivery. https://www.devopsinstitute.com/wp-content/uploads/Continuous-Security.pdf.
18. Bass J. M. (2019). Security challenges in large-scale agile development. ACIS 2019. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1006&context=acis2019.
19. National Institute of Standards and Technology (2020). Zero Trust Architecture (NIST SP 800-207). https://doi.org/10.6028/NIST.SP.800-207

20. Ali S., et al. (2021) Security-by-design: A comprehensive survey. Computers & Security, 111. https://doi.org/10.1016/j.cose.2021.102357.
21. Ahmad I., et al. (2021). Agile DevSecOps for cloud-native applications. IEEE. https://doi.org/10.1109/ACCESS.2021.3054527.
22. Martins R., et al. (2020). Security automation in CI/CD pipelines. Software Engineering Journal. https://doi.org/10.1109/MSEC.2020.3014683.
23 Stadnyk M., Palamar A. (2022). Project management features in the cybersecurity area. Scientific Journal of the Ternopil National Technical University, 2(106), 54–62. https://doi.org/10.33108/visnyk_tntu2022.02.054

УДК 004.056.55:004.4.233

ЗАСТОСУВАННЯ МЕТОДОЛОГІЇ SAFE ДЛЯ ІНТЕГРАЦІЇ КІБЕРБЕЗПЕКИ У ВЕЛИКОМАСШТАБНИХ ІТ-ПРОЄКТАХ

Марія Стадник

Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, Україна

Резюме. *Здійснено всебічний аналіз особливостей поєднання кібербезпекових практик із методологією Scaled Agile Framework (SAFe), що розглядається як цілісний механізм захисту великомасштабних ІТ-рішень. Показано, як упровадження компонентів DevSecOps – таких, як статичне й динамічне тестування безпеки (SAST і DAST), аналіз програмних залежностей (SCA), сканування контейнерних середовищ та контроль конфігурацій – забезпечує підвищення безпечності процесів Continuous Delivery та сприяє своєчасному виявленню вразливостей. Окрема увага приділена оцінюванні можливостей сучасних методів моделювання загроз, зокрема STRIDE, PASTA, які дозволяють ідентифікувати критичні ризики ще під час формування архітектурних концепцій.*

Проведено комплексний аналіз інструментів, за допомогою яких SAFe інтегрує кібербезпеку на всіх рівнях організації. Розглянуто роль DevSecOps у забезпеченні безперервного сканування вразливостей у CI/CD-конвеєрі, оцінено значення моделювання загроз та Zero Trust-архітектури для раннього виявлення ризиків і посилення архітектурної стійкості. Детально проаналізовано застосування Security Backlogs, критеріїв Definition of Done, ролі Security Champions та механізмів нормативної відповідності. Окремо наведено узагальнювальну таблицю, що демонструє відповідність інструментів SAFe міжнародним стандартам кібербезпеки та їхнє застосування на різних рівнях SAFe. Підкреслено роль Security Champions у посиленні комунікації між технічними командами та експертами з кіберзахисту, що сприяє формуванню сталої культури безпеки в середині організації.

Узагальнення результатів підтверджує, що SAFe створює повноцінну методологічну базу для вбудовування кібербезпеки на всіх рівнях управління та розроблення, що, у свою чергу, підвищує надійність, стійкість і якість ІТ-продуктів.

Ключові слова: *Agile, SAFe, кібербезпека, DevSecOps, Built-In Quality.*