Security Practices in Agile Development: An Industry Survey on Adoption, Perceived Impact, and Measurement in DevOps

Alejandra Selva-Mora[1], **Christian Quesada-López**[1], **Adrián Lara** [1], **Marcelo Jenkins**[1]

[1]Escuela de Ciencias de la Computación e Informática (ECCI)
Centro de Investigaciones en Tecnologías de la Información y Comunicación (CITIC)
Programa de Posgrado en Computación e Informática (PPCI)
Sistema de Estudios de Posgrado (SEP)
Universidad de Costa Rica
San Pedro de Montes de Oca – San José – Costa Rica

{alejandra.selvamora, cristian.quesadalopez, adrian.lara, marcelo.jenkins}@ucr.ac.cr

Abstract. *This paper presents an empirical survey study (N=24) examining the integration of security practices into agile and continuous development processes. The findings reveal that while the most adopted practices include Standards and Requirements (71%), Code Review (67%), and Compliance and Policy (63%), their implementation remains predominantly non-systematic. Despite moderate to high perceived impact, significant gaps exist between recognition and effective adoption, compounded by low security expert participation (37%) and a notable scarcity of metrics. The findings provide exploratory evidence on the current state of practice and key priorities for organizations looking to strengthen security integration while maintaining agility in their development processes.*

Keywords: *software security, agile development, DevOps, DevSecOps, BizDevOps, BSIMM, SDLC, shift-left, industry survey, effectiveness metrics*

1. Introduction

Agile software development facilitates rapid response to organizational changes, which has led to increased adoption of methodologies such as Scrum, XP, and DevOps that enable fast delivery while addressing emerging trends like Web 2.0, SaaS, IoT, and embedded systems. Consequently, software engineering has integrated concepts such as Continuous Integration and Continuous Deployment, which consider multiple contextual dimensions, including the **E**cosystem-**Strategy**-**A**rchitecture-**O**rganization model (ESAO).

Although prior research has proposed a variety of security practices to strengthen software development processes, empirical evidence shows that their adoption in real organizations is still limited and heavily influenced by organizational culture rather than technical constraints [Selva-Mora 2025]. Existing studies highlight obstacles such as lack of expertise, insufficient training, and inconsistent governance; however, there remains insufficient consolidated evidence on how frequently these practices are actually applied across the software development life cycle (SDLC) phases, how practitioners perceive their impact, and whether organizations measure their effectiveness—particularly in agile and DevOps settings.

To determine the actual use of security practices in software development environments, an industry survey was conducted with software professionals, primarily from Costa Rica. The questionnaire covered the level of involvement of information security professionals, the SDLC phases in which security practices are incorporated and which specific practices are used, as well as perceived benefits and challenges of using security practices. This study builds upon a previous mapping study by the authors [Selva-Mora and Quesada-López 2024], extending prior work by examining the real-world usage, perceived impact, and measurement maturity of security activities in industry contexts.

This study makes three main contributions: (1) characterizing how security activities are actually used across SDLC phases in agile and DevOps environments, providing updated organizational adoption patterns; (2) analyzing practitioners' perceived positive impact of these activities, identifying both recognized value and areas of uncertainty; and (3) examining measurement maturity by assessing whether and how organizations track security practice effectiveness. Together, these contributions explore the persistent gap between security's recognized importance and its effective adoption, a pattern consistently found in prior research and reinforced by this study's findings.

This report is structured as follows: Section II presents the related work; Section III includes the design and description of the survey; the results of the responses are presented in Section IV; subsequently, in Section V the obtained results are analyzed; then, the threats to validity are presented in Section VI; and, finally, the study's conclusions and future work are included in Section VII of the document. The questions included in the questionnaire, as well as the raw data and complimentary information are available at Appendices.

2. Related Work

The topic of security in software development has been relevant from both academic and research perspective. In this regard, the analysis of how computing professionals incorporate security practices has been the subject of several survey-type studies.

Several survey-based studies have examined how computing professionals incorporate security practices into software development. [Ur Rahman and Williams 2016] identified 4 main security practices in DevOps environments—task automation, team collaboration, security training, and non-automated reviews—finding that common DevOps activities effectively integrate security. [Wolden et al. 2015] found that the Cobit®5 framework strengthens security governance, while emphasizing that security awareness across the organization is essential. [Oyetoyan et al. 2016] highlighted that skills drive security activity adoption, that secure design is the most critical training need, and that systemic barriers, such as a lack of resources and plans, hinder security implementation. [Stewart and Jürjens 2017] demonstrated that the main causes of security incidents are organizational—lack of training, management directives, and compliance policies—and that a gap exists between security practice application and perceived impact. [Assal and Chiasson 2019] found that developers are self-motivated toward security but face organizational obstacles, concluding that a holistic approach addressing organizational issues is needed. [Rindell et al. 2021] verified that agile and security practices are used systematically together, with early lifecycle activities perceived as more impactful,

though a discrepancy exists between usage frequency and actual perceived impact.

Among these studies, [Ur Rahman and Williams 2016] and [Rindell et al. 2021] are the most closely aligned with the present study's focus on security practices and their impact. The present study extends this line of research by providing exploratory, practice-oriented evidence, analyzing security practices not only theoretically but also in real development environments, assessing both the frequency of use and perceived positive impact of security activities, organizational restrictions, and how effectiveness can be measured.

3. Survey Design

This section presents the research questions (RQs) and describes the measures taken to answer them. For the questionnaire design, best practices for surveys in the area of computer science have been considered, according to [Hui et al. 2019], [Genero Bocco et al. 2023], and [Caicedo Cavagnis and Zalazar Jaime 2018].

Research Questions. The RQs are aligned with that work and seek to analyze the adoption of security practices in agile software development environments. Although a distinction exists between the concepts of security practice and security activity according to [Ur Rahman and Williams 2016], both terms were used interchangeably in the survey to avoid confusion among respondents. However, a security activity is considered a specific action focused on achieving a small, well-defined objective with a tangible result, while a security practice refers to a collection of activities grouped by similarities [Ur Rahman and Williams 2016]. The RQs this study seeks to answer are as follows:

RQ1: To what extent are security activities used during software development? This research question was addressed by analyzing the extent to which both the 39 security activities proposed by the Building Security In Maturity Model [Black Duck 2025] and the security activities per SDLC phase proposed by [Rindell et al. 2021] are used.

RQ2: What is the positive impact perceived by software professionals of security activities in software development? This RQ complements RQ1 by identifying whether security activities are used because they are genuinely perceived as useful and contribute to improving software security, or because they are mandated by organizational guidelines without clear perceived benefits.

RQ3: How is the effectiveness of security activities applied in software development measured? This RQ aims to understand how organizations have defined procedures to verify whether the incorporation of security activities actually improves software security levels, as well as what the perceived benefits of applying these activities are, and what challenges persist for improving security levels in software development.

Questionnaire. The survey was designed with the specific objective of finding answers to RQs. Surveys related to the topic under study were analyzed, such as the research conducted by [Stewart and Jürjens 2017], [Rindell et al. 2021], [Ur Rahman and Williams 2016], [Wolden et al. 2015], [Oyetoyan et al. 2016], and [Assal and Chiasson 2019], as well as the academic articles analyzed in [Selva-Mora and Quesada-López 2024] to compile benefits and challenges to be included in the survey. Questions were proposed and submitted for review by the entire

research team, and then three cognitive interviews were conducted, where item clarity, redundancy, and conceptual alignment were evaluated, but no psychometric validation was performed due to sample size. The *think-aloud* technique was used to increase the reliability and validity of the questions [Caicedo Cavagnis and Zalazar Jaime 2018]. The final questionnaire comprises a total of 32 questions divided as follows: 8 questions on general information; 6 questions for the identification of the software development environment; 5 questions with specific security activities recommended for each SDLC phase according to [Rindell et al. 2021]; 10 questions focused on identifying benefits, challenges, and methods for assessing the effectiveness of performing security activities, based on results from [Selva-Mora and Quesada-López 2024]; and 3 closing questions.

Sampling and Survey Execution. We used invitation-based convenience sampling targeting software professionals within the authors' networks (ECCI, CITIC, and industry contacts), receiving 24 completed responses (completion ratio of 27.3%). Partially completed responses were excluded; no analysis of dropout bias was performed due to insufficient partial data. The survey was administered online using LimeSurvey and disseminated via invitation-based convenience sampling to computing professionals in the authors' academic and industry networks. The questionnaire remained open for a multi-month period (12/2024-09/2025) to ensure adequate participation. Participation was voluntary and anonymous. The results should be interpreted considering the small convenience sample (N=24) and the regional context.

Scales. There were used multiple-choice questions according to the topics related to the ongoing research, as well as single-choice questions where there were used 5-point or 6-point Likert scales.

Since positive impact level is a perception assessment, a weighting system was decided to help identify the relevance of each activity, so it is calculated as follows: $Impact\_Level = \sum_{i=1}^{5} weight_i \times Qty\_of\_Responses_i$ where *Impact_Level* is calculated for each security activity, according to the *weight* assign to each response options and the *Qty_of_Responses* received for each option. For these computations, Likert categories were mapped to numeric weights as follows: Very low=1, Low=2, Moderate=3, High=4, Very high=5. To compare impact level of each category, the result of the sum was normalized to 100, eliminating *Don't know/No answer* responses. This weighting scheme is an analytical device for this study and does not represent a validated psychometric scale.

For usage reports, a 5-point Likert scale is used: *Never*, *Rarely*, *Sometimes*, *Mostly used*, and *Systematically used*. These values will be used to analyze both usage frequency and the number of times each activity is reported. For the latter, only mid-range and higher values are considered (from *Sometimes* to *Systematically*).

Threats to Validity. *External validity*. Results reflect a convenience sample (primarily Costa Rica) and a small number of complete responses (N=24), limiting generalization. We report the sampling frame, invitations, and completion ratio to support interpretation. *Construct validity*. Impact weights follow a specified Likert mapping; "Don't know/No answer" was treated as uncertainty and reported separately. No gold-standard scales exist for several constructs; internal consistency is reported as a descriptive indicator. *Internal validity*. Self-reported usage and impact may be affected by recall and social desirability biases. To mitigate ambiguity, we included cognitive interviews and provided examples

for key activities. *Conclusion validity*. Given N=24, we did not perform inferential statistics; results are descriptive. We provide all item-level distributions to enable independent assessment.

4. Analysis of Results

Demographics and Experience: The survey profiled software development professionals with an average age of 36 years. Respondents were predominantly male, with men accounting for 96% of participants and women for only 4%, reflecting a significant gender imbalance. All respondents hold university degrees, indicating a highly educated group. In terms of professional background, 12% came from fields outside of software development, including health sciences, business and economics, and other engineering disciplines.

Organizational Information: The respondents work across various industry sectors, with the largest concentration in software development and IT services (46%), followed by education (17%), and banking and insurance/financial services (13% each). Regarding organization size, nearly half of respondents (46%) work in organizations with over 1,000 employees, while 29% work in mid-sized organizations (between 250 and 1,000 people). Overall, 75% of respondents work in organizations with more than 250 employees. This suggests that most are working in substantial organizations that likely rely heavily on software to support business processes, enhance collaboration, and improve decision-making. When examining IT department sizes, half of the respondents reported working in IT departments with over 100 people, while 25% have IT departments with 6 to 20 people, and 17% work in IT departments with 5 or fewer staff members. Interestingly, there appears to be a notable disconnect between overall organization size and IT department size. This finding aligns with results from [Selva-Mora and Quesada-López 2024], which suggests that many organizations' IT focus remains on maintaining operations than engaging in adequate security planning.

Software Development Environment. This section of the survey aimed to gain an understanding of the software development environment within the organizations where respondents work.

Regarding software development methodologies, the results showed that Scrum is the most widely used methodology (17 responses), followed by both Kanban and hybrid methodologies (11 responses each), and then waterfall and incremental methodologies (7 respondents each). Other participants mentioned prototyping, extreme programming, Microsoft's Security Development Lifecycle (SDL), and Rapid Application Development. Overall, the results reveal that agile methodologies are the most commonly used for software development, which is consistent with the current trend of organizations seeking to deliver software into production more quickly.

In terms of the frequency with which security experts are involved in the software development process, 63% of respondents reported that experts *Never*, *Rarely*, or *Sometimes* get involved, while only 37% reported *Most of the time* or *Always*. Further analysis would be needed to understand this low level of involvement, but this situation confirms that one of the main challenges in incorporating security practices is eliminating the lack of knowledge and awareness about security, due to the absence of experts participating in software development [Selva-Mora and Quesada-López 2024].

Regarding whether development teams have sufficient knowledge about information security, 8 responses indicated *Neither disagree nor agree*, 7 people responded *Disagree*, 6 people responded *Agree*, 3 people said *Strongly agree*, and none *Strongly disagree*. Although these results present balanced opinions, 33.33% neither agree nor disagree, making it difficult to draw a conclusive finding.

Results from next question indicate that security practices are mainly integrated during technical phases of the SDLC, particularly *Implementation* (18 responses) and *Testing* (14), followed by *Design* (11) and *Release* (10). *Pre-project* security activities are also reported. In contrast, *Requirements* shows minimal security integration, diverging from academic recommendations that advocate shift-left security, which emphasizes embedding security requirements early in development rather than applying countermeasures late. Regarding development approaches, 63% responded using *DevOps* to accelerate delivery and enhance quality through automation. Only 3 respondents identify with *DevSecOps*, which explicitly integrates security practices, while 5 report no defined approach. The relationship between the phases in which security practices are integrated and the software development approach can be visualized in Figure 1.

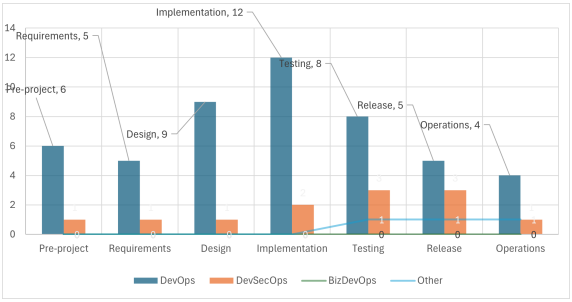

Figure 1. Security practices per SDLC phase and development approach

Results on BSIMM security practices are presented in Table 1. According to the responses received, the most widely adopted activities are *Standards and requirements* (71%), associated with pre-project phases and focused on defining security requirements, followed closely by *Code review*, with 67% of responses, reflecting strong integration of security during the implementation phase. Governance-related practices are also prominent, with 63% of respondents implementing *Compliance and policy* activities, and 54% reporting *Training* initiatives. *Architecture analysis* is implemented by 54% of participants, highlighting attention to early design-stage security. Deployment-phase practices are less frequent, with 50% addressing *Software environment* security, 42% integrating *Security testing* and *Configuration and vulnerability management*, and 29% conducting *Penetration testing*. Lower adoption rates are observed for *Security features and design* (38%) and *Attack models* (25%). Strategic practices show the weakest integration, as only 13% implement *Strategy and metrics*, indicating limited emphasis on formal security planning and measurement.

4.1. Security Activities Utilization and Positive Impact

The third section of the survey aims to identify the actual use of specific security activities and gather respondents' opinions on whether their integration represents a positive impact on the organization.

Table 1. BSIMM security practices reported by participants

Domain	*Security Practice*	*Qty Reported*
Governance	Strategy and metrics	3
	Compliance and policy	15
	Training	13
Intelligence	Attack models	6
	Security features and design	9
	Standards and requirements	17
SSDL Touchpoints	Architecture analysis	13
	Code review	16
	Security testing	10
Deployment	Penetration testing	7
	Software environment	12
	Configuration management and vulnerability management	10
	Other	2

Requirements Analysis Phase (Figure 2): Most security activities in this phase are performed at least *Sometimes*, with few being *Rarely* or *Never* applied. Three activities stand out for *Systematically used*, each reported by 6 professionals: *Security architecture review*, the creation of the *Data classification scheme and inventory*, and *Define application goal and criticality*. Regarding perceived positive impact, it is predominantly *Moderate* to *High*, with 59 responses each (27%), followed by *Very high* (36 responses, 17%) and *Low* impact (20 responses). Only one response (1%) indicates *Very low* impact, and 41 responses (19%) correspond to *Don't know/No answer*. Across 216 assessments covering the nine activities, the most frequently used practice is *Application of security and privacy risk analysis* (22), although its adoption is largely informal. This is followed by *Security architecture review* and *Define application goal and criticality*, each with 21 reports and the highest levels of systematic use. The least adopted activities, with 17 reports each, are creating the *Data classification scheme and inventory*, translating Compliance constraints to requirements, and *Conduct a business impact analysis*.

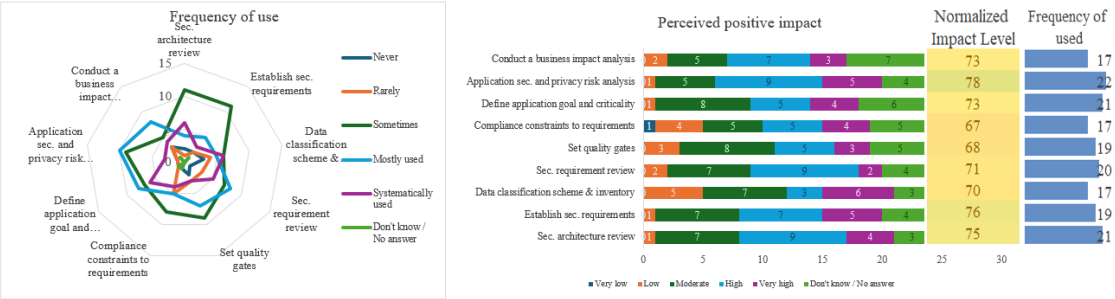

Figure 2. Results for security activities in the requirements analysis phase

Design Phase (Figure 3): Results show that most security activities are applied at least *Sometimes*, although several respondents report *Rarely* or *Never*. The most *Systematically used* activity is *Design requirements*, reported by 6 professionals, followed by *Architecture and application development guidelines* and *Application security settings definition*, each with 4 *Systematically used* reports. Perceived positive impact follows a similar pattern to the requirements phase, with *High* (26%) and *Moderate* (22%) impact dominating, followed by *Very high* impact (20%). Both *Low* (9%) and *Very low* (3%) impact assessments are limited, while 20% of responses correspond to *Don't know/No answer*, reinforcing concerns about limited confidence in assessing the effectiveness of design-phase security activities. Overall, the most frequently used activities are *Design requirements* (21), *Architecture and application development guidelines* (18), and *Abuse*

or misuse cases together with *Attack surface analysis and reduction* (17 each). Lower adoption is observed for *Threat modeling* (16) and *Application security settings definition* (15), with 8 respondents indicating *Rarely* or *Never*. Of particular concern is the limited use of *Threat modeling* (only 20%) despite its recognition as a fundamental secure design practice in frameworks such as BSIMM, the NIST Cybersecurity Framework, and OWASP's Application Security Verification Standard. This gap suggests barriers related to the high level of specialization required, the steep learning curve associated with attack pattern knowledge, and the time investment needed, which may conflict with agile development practices. Consistent with the previous phase, activities related to documentation, standardization, and compliance show higher adoption than those requiring deeper contextual analysis.

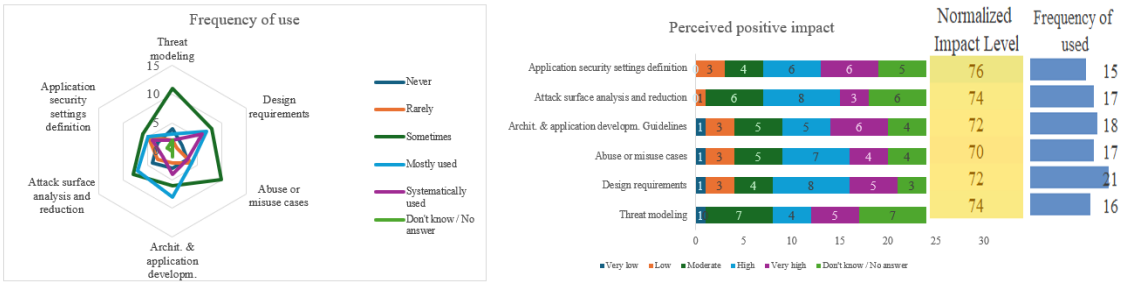

Figure 3. Results for security activities in the design phase

Implementation Phase (Figure 4): Like the previous phases, these results indicate that most security activities are applied *Sometimes* or *Systematically used*, with none reported as *Mostly used* and several reported as *Rarely* or *Never* applied. A notable increase in *Don't know/No answer* responses is observed (22) compared to only 4 in both the requirements and design phases. Regarding perceived positive impact, it differs from earlier SDLC phases. Among 192 assessments, *Moderate* positive impact predominates (30%), followed by *High* (28%) and *Very high* impact (16%). *Low* (5%) and *Very low* (1%) impact assessments remain limited. However, *Don't know/No answer* responses account for 21% of assessments (40), reinforcing concerns raised by the high level of uncertainty in activity usage. This is particularly concerning given that this phase involves code development, integration of external components, and unit testing, and may reflect insufficient security knowledge, consistent with the second most frequently reported challenge in [Selva-Mora and Quesada-López 2024]. The activities most frequently reported as systematic are *Approved tools* (13), *Coding standards* (11), and *Static analysis* (7). Overall, the most widely used activities are *Coding standards* and *Approved tools* (17 each), followed by *Security reviews with automated tools* (14). The least adopted activities are *Security specific hardening sprints* (7), *Deprecation of unsafe functions*, and *Documentation of security solutions* (11 each). An important finding is that none of the proposed activities is reported as being used *Mostly*, suggesting that implementation phase security practices are applied primarily when mandated by formal processes. This indicates that even though these practices are widely recommended, they are not consistently integrated into developers' daily workflows unless explicitly required, aligning with findings by [Selva-Mora and Quesada-López 2024] that identify a lack of security awareness as a major challenge.

Verification and Validation Phase (Figure 5): The results show that most proposed se-

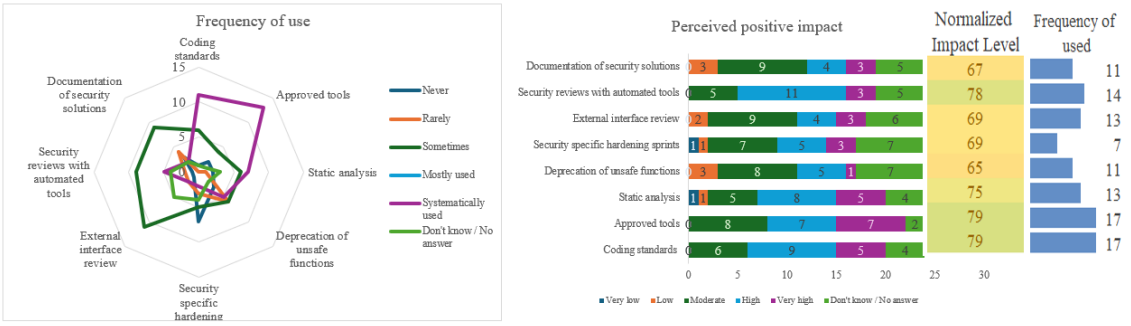

Figure 4. Results for security activities in the implementation phase

curity activities are applied *Sometimes*, followed by *Mostly used*, with similar levels of *Rarely* and *Systematically used*, and a higher incidence of *Never* responses than in earlier SDLC phases. As in the implementation phase, a high proportion of *Don't know/No answer* responses persists, indicating that respondents are often unsure whether security activities are applied during testing. This is particularly concerning given that implementation and verification phases are complementary: vulnerabilities introduced during development are expected to be identified during testing. This raises the possibility that some vulnerabilities may reach production and be discovered through incidents rather than preventive controls. Perceived positive impact aligns with the impact observed in the implementation phase. Among 216 assessments, *Moderate* positive impact dominates (27%), followed by *High impact* (24%) and *Very high* impact (13%). *Low* and *Very low* impact assessments each represent 6%. *Don't know/No answer* responses account for 23% of assessments (49), reinforcing concerns about limited visibility, awareness, or confidence in evaluating the effectiveness of security activities during this critical SDLC phase. The activities most frequently reported as systematic are *Code reviews during testing* (8), followed by *Dynamic analysis* (5), and *Fuzz testing* and *Automated testing tools* (4 each). In terms of overall use, *Code review during testing* is the most adopted activity (17), followed by *Automated testing tools* (16) and *Security specific test cases* (15). The least adopted activities are *Review security testing plans* (10, with equal numbers of *Rarely* and *Never* responses), *Attack surface review* (11, including 7 *never* responses), and *Fuzz testing* and *Penetration testing* (12 each).

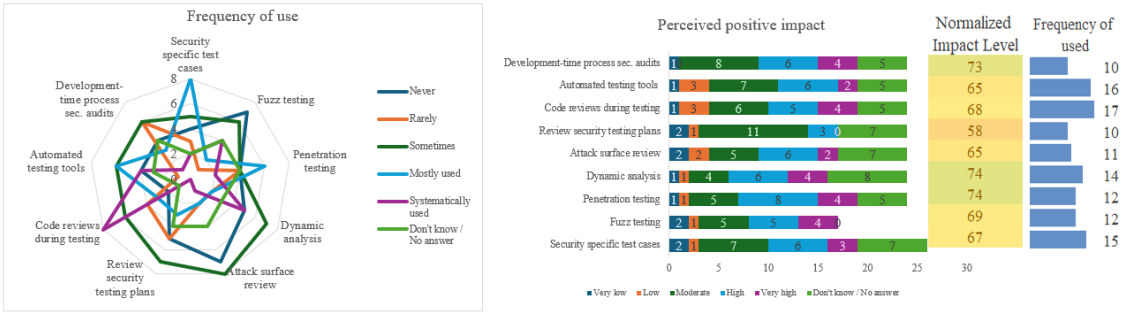

Figure 5. Results for security activities in the verification and validation phase

Release Phase (Figure 6): Results show that, as in other SDLC phases, most security activities are applied *Sometimes*, followed by *Mostly used*. Notably, *Never* is the third most frequent response, representing 21% of all responses, the highest among all phases.

This is particularly concerning given that the release phase prepares software for operation in production environments, where inadequate security practices can directly lead to exploitable vulnerabilities. Although the number of *Don't know/No answer* responses decreases compared to previous phases, it remains high. This is especially striking given that 75% of respondents report adopting DevOps or DevSecOps approaches, which emphasize close collaboration between development and operations, yet the results suggest limited awareness or application of release phase security activities. Regarding perceived positive impact, among 168 assessments, *High* impact dominates (30%), followed by *Very high* impact (18%) and *Moderate* impact (18%). *Low* (4%) and *Very low* (3%) impact assessments remain limited. However, this phase records the highest proportion of *Don't know/No answer* responses across the SDLC (27%, 45 responses), indicating persistent uncertainty about the value and consequences of release phase security activities. The activities most frequently reported as *Systematically used* are *Ensure host and network security basic* (8), *Incident response plan created* (7), and *Security patch planning* (5). In terms of overall use, the most frequently reported activities within this sample are *Ensure host and network security basic*, *Incident response plan created*, *Internal security audits*, and *Security patch planning*, each with 18 usage reports. The least adopted activities are *Formal certification* (10, including seven *Never* responses), *External security audits* (11, eight never), and *Documentation required by regulations* (15, six *Never*).

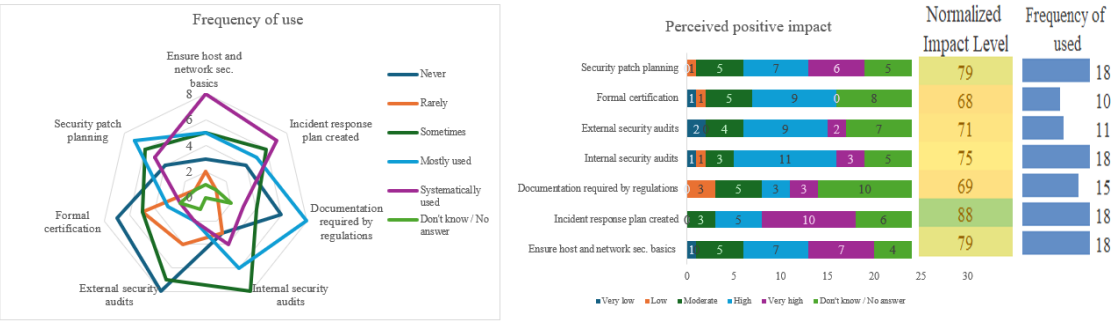

Figure 6. Results for security activities in the release phase

4.2. Benefits, Challenges and Effectiveness

This survey section examines three aspects: the perceived benefits of integrating security activities within the SDLC, the challenges that hinder achieving those benefits, and the methods organizations use to measure the effectiveness of security activities.

Benefits: The survey examines perceived benefits based on advantages identified in prior literature and survey design analysis [Selva-Mora and Quesada-López 2024]. The most frequently reported benefit is *Early identification of security requirements* (16 reports, 67% of respondents), which aligns with recommendations to address security in the earliest SDLC phases [Williams 2019] and supports the DevOps principle of shifting security left ([Lohrasbinasab et al. 2020]). This is followed by *Reduction in the number of security incidents* and *Enhanced security awareness across the organization*, each reported by 15 respondents, indicating improvements in system stability, planning, and shared responsibility. A third commonly reported benefit is *Well-informed decisions regarding the integration of security at all levels* (13), reflecting improved understanding of threats

and exploitation paths. The least frequently observed benefits (8) include *Most appropriate documentation* and *Improved communication with users and stakeholders*. While reduced documentation may be expected in agile environments, the lack of structured security documentation is problematic given its recognized importance in agile contexts ([Rindell et al. 2021]; [Wolden et al. 2015]). Limited improvement in business and stakeholder communication also suggests that security integration remains primarily technical, with insufficient involvement from non-technical domains.

Benefits reported by 9 respondents include *Reducing the criticality of security incidents*, *Effective alignment between agility and security*, and *Better customer satisfaction*. These low adoption levels raise concerns about the real impact of security practices, particularly in agile environments where security is often deprioritized [Jaatun and Cruzes 2021]. Finally, only 11 respondents (46%) report *Better understanding of actual and potential attack patterns*, highlighting a significant gap, as such understanding is essential for timely vulnerability identification and mitigation. This finding reinforces the need for a coherent security model that clearly defines organizational security objectives and guides security integration across development, architecture, and operations [Villalón-Fonseca 2022].

Challenges: The survey identifies challenges based on categories derived from prior literature [Selva-Mora and Quesada-López 2024]. The most frequently reported challenge is *Difficult to adapt software developed to changes* (46%), likely due to security mechanisms being tightly coupled with functionality. This finding supports the need for asset-oriented security strategies, as proposed by [Villalón-Fonseca 2022]. *Maintain a steady pace of work* follows closely (42%), reflecting conflicting stakeholder priorities that favor functionality and delivery speed over security [Terpstra et al. 2017]. *Adequate compliance with policies and standards* is reported by 33% of respondents, suggesting limited regulatory awareness or weak organizational security posture. Additional challenges, each reported by 29% of respondents, include *Decrease in software delivery frequency* and *Insufficient collaboration between the business and developers*. Less frequently cited, but still relevant (21%), are *Lack of focus on the customer* and *Surplus workload attributed to authorizations and documentation*, both associated with low organizational security awareness.

Regarding organizational limitations, the most significant is *Complexity in the software environment arising from evolving requirements* (58%), reinforcing the need for intrinsic security approaches that reduce rework [Villalón-Fonseca 2022]. *Lack of security knowledge by users and developer teams* follows (46%), consistent with prior findings that key stakeholders often lack sufficient security understanding [Terpstra et al. 2017]. *Documentation does not provide sufficient information to effectively implement subsequent changes* and *Insufficient security awareness across the organization* are each reported by 42%, echoing concerns identified in [Selva-Mora and Quesada-López 2024] and supported by evidence that security decisions are often undocumented in agile environments [Cruzes et al. 2018]. *Expenditures associated with the implementation of security practices* (38%), *Lack of security personnel in the software development process* (33%), and *Development teams fail to deliver expected performance* (21%) are less frequently reported but remain relevant. These constraints are closely linked to organizational security awareness and leadership commitment, suggesting that increased awareness could miti-

gate cost concerns, staffing gaps, and performance issues.

Effectiveness Measurement: Results indicate limited maturity in measuring the effectiveness of security practice implementation. A total of 38% of respondents reported having no related metrics, while 33% selected *Don't know/No answer*, indicating low visibility or use of metrics even when they exist. Only 29% confirm the existence of security-related metrics. Reported metrics mainly focus on technical indicators (e.g., outdated packages, public IP exposure, use of secrets and certificates) rather than on evaluating the quality or effectiveness of security activities, suggesting the absence of formal measurement processes in most organizations. Responses regarding metric management further reinforce this finding. While some respondents demonstrate higher maturity through references to security posture measurement, use of industry standards, alert systems, and continuous improvement processes, these represent a minority. Only 25% of respondents describe practices consistent with a formal information security management structure, including defined responsibilities, periodic review of metrics, and corrective actions. Overall, the lack of systematic measurement limits organizations' ability to evaluate, adjust, and continuously improve the integration of security practices across the SDLC.

5. Discussion

The survey suggests a recognition–adoption gap: practices are widely perceived as beneficial, yet most are applied only *Sometimes*, not as a formal procedure. Combined with infrequent expert involvement (37% *Most of the time/Always*) and limited measurement maturity (38% no metrics; 33% don't know), the results indicate that organizational and process factors constrain systematic security integration.

The study posed three research questions. The first investigates to what extent security activities are used during software development. Respondents report that the most integrated BSIMM security practices are *Standards and requirements* with 71%, followed by *Code review* with 67%, *Compliance and policy* with 63%, *Training* with 54%, and *Architecture analysis* with 54%. In more detail, the security activities used show a predominance of informal implementation, with the most reported frequency being *Sometimes*, with few activities reported as *Systematically used*, suggesting more reactive than planned use. The high percentage of *Don't know/No answer* responses across all SDLC phases is concerning, both in use and perceived positive impact, indicating that security knowledge is not as high as required. Although the most used activities are utilized in requirements analysis and design phases, this occasional use reveals that a more genuine shift-left strategy is required where security is considered from early stages through formalized processes. The most frequently used activities are those responding to more established and generic processes, tending to perform those requiring more case-by-case analysis like threat modeling less frequently.

The second research question examines the perceived positive impact of security activities. Generally, they are perceived as having a moderate to high positive impact, implying that developers are aware that security integration can bring benefits. Few activities are reported with low or very low impact. although many *Don't know/no answer* responses were received, suggesting that more security training is needed. A discrepancy must be recognized between perceived positive impact and actual use, since activities are recognized with moderate to high impact, but their use is not systematic but occasional,

with a gap existing between recognizing these practices' value and their actual adoption that does not necessarily depend on each developer, but also on organizational formal policies and processes. The greatest reported benefits are early identification of security requirements, reduction in number of security incidents, increased security awareness, and well-founded decisions. However, no benefit is reported by more than 67% of respondents, implying that one-third do not perceive these values as results of incorporating more activities.

The third question investigates how the effectiveness level of applied security activities is measured. Results show a broad absence regarding effectiveness metrics, with 71% of respondents indicating they have no related metrics or are unaware of them, which goes against good process management practices where stakeholders should know each metric, how it is calculated, where data comes from, and what expected values are.

Implications for industry include the urgent need to formally integrate security into software development contexts. The study shows low participation of security experts during various SDLC phases with only 37% consistently, representing a potential organizational risk. Organizations require establishing formal mechanisms to involve specialists, defining criteria on when to request security support, and creating multidisciplinary teams. Real transitions from DevOps to DevSecOps approaches that integrate business needs are necessary, meaning a BizDevOps approach. This lack of experts must be accompanied by security training and awareness plans, not only for IT personnel but across all organizational areas, through structured training programs, certification promotion, and designing a plan that improves organizational culture so security is a priority from any business perspective. Organizations should strive to maintain balance between software delivery speed and meeting adequate security levels through strategic integration of security practices without harming continuous delivery. As a complement, organizations must establish a culture of measuring security activities effectiveness, since absence of metrics and evaluation methods hinders adequate decision-making.

Implications for research include proposing specific models allowing security integration in agile software development environments, primarily based on BizDevOps approaches, so security activities are included at strategic workflow points integrating all organizational aspects. Existing academic and industry frameworks could be analyzed, identifying their use and effectiveness levels to identify gaps that could be resolved with new validated proposals. Regarding effectiveness, establishing a method to measure security practices integration effectiveness becomes essential, as it is one of the least covered topics according to results and knowing effectiveness helps improve not only the method but results. Designing security metrics that do not represent significant additional workload is necessary, but rather are embedded in development and operations to collect clear data showing how security levels improve. Three priorities emerge for practice into the SDLC: (1) Promote a shift-left practice, focus on integrating security lightweight practices from the beginning rather than ad hoc tasks; (2) Institutionalize expertise and clear responsibilities for involving security specialists; and (3) Formalize an organizational process to manage security metrics that provides feedback and allows improvement on security practices.

6. Conclusions

This survey provides exploratory evidence of how security practices are integrated into agile and DevOps environments. While practitioners report moderate-to-high perceived impact, adoption remains largely non-systematic, expert participation is infrequent, and measurement maturity is low, revealing a persistent recognition–adoption gap. The results show that, although many organizations report using DevOps or DevSecOps approaches, security integration often appears limited within this sample and requires greater operational effort to gain real strength in day-to-day development. Three priorities emerge as potential areas for organizations to explore: (i) operationalizing shift-left through lightweight, requirements-phase activities; (ii) institutionalizing security expertise via clear engagement criteria and responsibilities; and (iii) adopting minimal, actionable metrics to support continuous monitoring and improvement. At the same time, developers and organizations acknowledge the importance of strengthening security integration and the need to invest time and resources to make adoption more consistent and effective. Future work includes broadening the sample, refining the instrument's reliability, and evaluating targeted interventions—such as security champions and metrics dashboards—to improve adoption and strengthen security outcomes.

A key element for achieving successful security incorporation is awareness, not only in information technologies areas, but throughout the entire organizational structure. If the security posture is improved and all parties involved prioritize it, efforts are more likely to yield better results, as there will be more perspectives to help identify vulnerabilities, threats, and scenarios that could be addressed in a timely manner.

As previously mentioned, this study is part of ongoing research; therefore, the research team will take the obtained results to continue the investigation and seek to obtain an outcome that supports organizations in improving their approach and posture regarding information security.

7. Acknowledgements

This work has been developed with the support of CITIC, ECCI, PPCI, SEP, and Research Project 834-C4-157 of the University of Costa Rica.

8. References

Assal, H. and Chiasson, S. (2019). Think secure from the beginning: A survey with software developers. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13.

Black Duck (2025). BSIMM15 Report 2025: Building Security In Maturity Model. Technical report, Black Duck Software, Inc. Fifteenth edition of the BSIMM study.

Caicedo Cavagnis, E. E. and Zalazar Jaime, M. F. (2018). Entrevistas cognitivas: Revisión, directrices de uso y aplicación en investigaciones psicológicas.

Cruzes, D. S., Jaatun, M. G., Bernsmed, K., and Tøndel, I. A. (2018). Challenges and experiences with applying microsoft threat modeling in agile development projects. In *2018 25th Australasian Software Engineering Conference (ASWEC)*, pages 111–120. IEEE.

Genero Bocco, M., Piattini Velthius, M., Cruz-Lemus, J., and Díaz García, O. (2023). *Métodos de Investigación en Informática*. AQCLab, Ciudad Real, España, 1 edition.

Hui, W., Lui, S. M., and Lau, W. K. (2019). A reporting guideline for is survey research. *Decision Support Systems*, 126:113136.

Jaatun, M. G. and Cruzes, D. S. (2021). Care and feeding of your security champion. In *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–7. IEEE.

Lohrasbinasab, I., Acharya, P. B., and Colomo-Palacios, R. (2020). Bizdevops: A multivocal literature review. In *International Conference on Computational Science and Its Applications*, pages 698–713. Springer.

Oyetoyan, T. D., Cruzes, D. S., and Jaatun, M. G. (2016). An empirical study on the relationship between software security skills, usage and training needs in agile settings. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pages 548–555. IEEE.

Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., and Leppänen, V. (2021). Security in agile software development: A practitioner survey. *Information and Software Technology*, 131:106488.

Selva-Mora, A. (2025). A bizdevops-aligned framework for integrating security practices in agile software development. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 68–70. IEEE.

Selva-Mora, A. and Quesada-López, C. (2024). Security practices in agile software development: A mapping study. In *Proceedings of the 7th ACM/IEEE International Workshop on Software-intensive Business*, pages 56–63.

Stewart, H. and Jürjens, J. (2017). Information security management and the human aspect in organizations. *Information & Computer Security*, 25(5):494–534.

Terpstra, E., Daneva, M., and Wang, C. (2017). Agile practitioners' understanding of security requirements: Insights from a grounded theory analysis. In *2017 IEEE 25th international requirements engineering conference workshops (REW)*, pages 439–442. IEEE.

Ur Rahman, A. A. and Williams, L. (2016). Security practices in devops. In *Proceedings of the Symposium and Bootcamp on the Science of Security*, pages 109–111.

Villalón-Fonseca, R. (2022). The nature of security: A conceptual framework for integral-comprehensive modeling of it security and cybersecurity. *Computers & Security*, 120:102805.

Williams, L. (2019). Secure software lifecycle knowledge area issue. *The National Cyber Security Center*.

Wolden, M., Valverde, R., and Talla, M. (2015). The effectiveness of cobit 5 information security framework for reducing cyber attacks on supply chain management system. *IFAC-PapersOnLine*, 48(3):1846–1852.