Universiteit
Leiden
The Netherlands

“Have heard of it”: a study with practitioners on adoption of secure software development frameworks

Wee, A.; Kudriavtseva, A.; Gadyatskaya, O.

Citation
Wee, A., Kudriavtseva, A., & Gadyatskaya, O. (2024). “Have heard of it”: a study with practitioners on adoption of secure software development frameworks. *2024 Ieee European Symposium On Security And Privacy Workshops (Euros&pw)*, 626-633. doi:10.1109/EuroSPW61312.2024.00076

Version:	Publisher's Version
License:	Licensed under Article 25fa Copyright Act/Law (Amendment Taverne)
Downloaded from:	https://hdl.handle.net/1887/4213013

Note: To cite this publication please use the final published version (if applicable).

"*I have heard of it*": A study with practitioners on adoption of secure software development frameworks

1st Alex Wee
LIACS, Leiden University
Leiden, The Netherlands
t.s.a.wee@umail.leidenuniv.nl

2nd Arina Kudriavtseva
LIACS, Leiden University
Leiden, The Netherlands
a.kudriavtseva@liacs.leidenuniv.nl

3rd Olga Gadyatskaya
LIACS, Leiden University
Leiden, The Netherlands
o.gadyatskaya@liacs.leidenuniv.nl

2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) | 979-8-3503-6729-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/EUROSPW61312.2024.00076

***Abstract*—The ever-growing incidence of software vulnerabilities giving rise to devastating cyber attacks pushes organizations and governments to take software security more seriously. To avoid vulnerabilities, organizations producing software strive to adopt secure software development frameworks (SSDFs). Our mixed-methods study with software development practitioners focuses on the SSDFs' adoption trends and examines the key factors influencing organizational decisions regarding secure software development. Our findings from a survey (n=37) and interviews (n=8) with software development practitioners indicate that, while being aware of the existing SSDFs, organizations mostly use custom-made frameworks that afford more flexibility and align better with the development processes.**

***Index Terms*—Secure software development; software security practices; mixed-methods research; study with practitioners.**

1. Introduction

Cybersecurity threats have been on a rising trend in the past decade. Besides leading to an increasing number of breaches, they have also been of increasing impact, as seen in two high-profile cases in recent years – the WannaCry ransomware attack in 2017 and the SolarWinds attack in 2020. To reduce the impact of cyber attacks, organizations invest in multi-pronged strategies to improve their security posture. These include infrastructure-related measures, such as strengthening network defenses and hardening systems, and process controls, such as user account validity reviews and vetting of personnel. In addition to that, organizations started adopting secure software development activities to detect and eliminate vulnerabilities [3]–[5] and, ultimately, ensure the security of the software they deploy [9], [11].

Secure software development frameworks (SSDFs) imply the integration of secure development practices throughout the software development lifecycle (SDLC) to address software vulnerabilities more proactively and holistically. It has been shown that developers might only actively consider security during coding, while implicitly disregarding other app development-related activities as security-irrelevant [24]. SSDFs, such as the Microsoft Security Development Lifecycle [10] and the Comprehensive, Lightweight Application Security Process (CLASP) [19] released by the Open Web Application Security Project (OWASP), help practitioners to think about security early and throughout the software development process.

While deciding on which framework or activities to adopt, organizations are often faced with competing demands on their finite resources and are therefore looking for the most cost-effective and least resource-intensive activities to adopt that can offer the most secure outcomes [9]. In this research, we look into the SSDFs and their component activities along with the factors that organizations consider when deciding upon which framework/practices to adopt. More specifically, the goal of our study is to answer the following research questions:

- **RQ1:** What are the current practices regarding the adoption of SSDFs?
- **RQ2:** What are the key considerations for organizations in deciding on the SSDFs and practices to adopt?
- **RQ3:** For organizations that use custom SSDF, what are the factors influencing this decision?
- **RQ4:** What are the drivers, changes, and trends in the adoption of SSDFs?

2. Related Work

There are many established SSDFs published by companies, academia, and government organizations that have been studied in several surveys [7], [13], [17], [20], [27]. They provide structured approaches to integrating security practices, such as threat modeling, penetration testing or security code review, into the software development process, aiming to mitigate vulnerabilities early and enhance the overall security posture. In the scope of this study, we refer to these established frameworks as *formal frameworks*.

2.1. Adoption studies

Several previous studies investigated the adoption of formal SSDFs or the adoption of (sets of) security activities prescribed by different frameworks.

Weir et al. [25] identified 12 of the most used secure development activities that are adopted by more than 50% of the participating developers in their projects. Assal and Chiasson [4] found that security-aware developers

This research was partially supported by the Dutch Research Council (NWO) under the project NWA.1215.18.008 Cyber Security by Integrated Design (C-SIDe).

© 2024, Alex Wee. Under license to IEEE.

DOI 10.1109/EuroSPW61312.2024.00076
Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.

adopt many more practices across the SDLC compared to security-inattentive developers, and the degree of attention to security is largely determined by the team culture.

Still, even security-aware developers do not pay enough attention to security activities during the design stage. Surveying developers, Assal and Chiasson [5] also found that, on average, software projects dedicate 19% of time to security-related activities. They reported that company-wide security pushes, such as the adoption of company-wide best practices and baseline security standards, team support, and in-house tools and frameworks are some factors strongly influencing the adoption of security activities.

Morrison et al. [16] identified 16 core security practices that commonly feature in SSDFs; most of those practices were used daily by at least one developer participating in the study. Ryan et al. [22] surveyed the adoption of the 12 most common security practices from Weir et al. [25] and measured the correlation of the security practices adoption with the security culture in the studied organizations. They found that while a strong security culture was conducive to the usage of security practices, overall, there was only a weak correlation. They also reported that frequently the reported usage of security practices did not result in much time being spent on doing them, indicating that such organizations might focus on compliance rather than security [22].

Focusing on the adoption of formal SSDFs, the study by Geer [9] found that 81% of the surveyed organizations were aware of SSDFs, while only 30% used one. Yet, times change, and more and more organizations adopt secure software programs [26]. It is thus opportune to revisit the question of the adoption of SSDFs in organizations.

Recently Kanei et al. [11] reported that, based on a large survey with developers from the US and Japan, in-house security development programs and guidelines are very popular in organizations. They call to look more in-depth into the in-house frameworks to better understand their usage trends. This is one of the focuses of our study.

2.2. Challenges in adoption

Although there is substantial information available on secure development activities and frameworks, several studies have shown that organizations are facing challenges in adopting them. Assal et al. [5] observed that many software developers are motivated to develop secure software but are deterred when they have to deal with competing priorities, the lack of resources, or insufficient security knowledge. Gasiba et al. [8] found that, while software developers were intent on complying with secure coding guidelines and were aware of their importance, knowledge of the company's secure software development guidelines was lacking. Kudriavtseva and Gadyatskaya [14] reported that while SSDFs prescribe security activities throughout the SDLC, security measurement tends to occur at the later stages, and organizations do not invest in assessing security early in the development process.

Several other studies have also observed challenges faced and suggested some form of adaptation of security interventions to better fit the context of the organization. Kirlappos et al. [12] identified a perceived conflict of security with productive activities as the key driver for employees' non-compliance to security-related policies in organizations. They concluded that an effective resolution to this problem requires security measure adaptation and de-centralization of security-related decisions to empower developers to make context-related choices. This is why in our research we also focus on self-tailored, *custom frameworks* that organizations develop and adapt to their own needs.

3. Methodology

We use a mixed-methods two-stage study design (first survey and then interviews) as this allows us to collect information from potentially a richer variety of participants, and then deepen the insights obtained in the survey study using semi-structured interviews. All participants were informed about the study objectives, the type of information being collected, and our data management plan, and consented to participation. The study design was approved by the relevant Ethics Review Committee at our university.

3.1. Baseline SSDFs

Based on the studies and other literature that compared and analyzed different frameworks [1], [6], [13], [27], we have ascertained the more prominent frameworks to be:

1) Microsoft Security Development Lifecycle (SDL) [10]
2) OWASP Comprehensive, Lightweight Application Security Process (CLASP) [19]
3) McGraw's Touchpoints [15]
4) NIST Special Publication 800-160 on Systems Security Engineering [21]
5) Software Assurance Forum for Excellence in Code (SAFECode) [23]

In addition to that, the Supply-Chain Levels for Software Artifacts (SLSA) [18], a security framework from Google to prevent tampering and improve the integrity of software packages, has been growing in prominence in the light of software supply chain attacks and so we included it in our baseline.

We selected these six established and prominent frameworks as the baseline to discuss with our participants, as we expected that at least some of them would be familiar. We asked the participants if their organizations had adopted one of these baseline frameworks or any other formal, established framework, or if they had their own custom framework.

3.2. Survey

The survey was first conducted as a quantitative method to measure the adoption rate of secure development frameworks and the various security practices during software development. The survey also aimed to understand the reasons behind the adoption decisions. Most of the questions were multiple-choice questions, with an option for the "Other" input field that allowed entering free text.

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.

The survey was published online on the Qualitrics platform[1]. It was open to responses from 29 March 2022 to 30 April 2022 and required approximately 15 minutes to complete. There were a total of 27 questions split into 4 sections: (1) information about the current/recent software project that is the context of these questions; (2) questions on secure software development; (3) demographic information about the respondent and their company; (4) concluding questions that provide an opportunity to stay in touch with the research team. To mitigate any potential issues, the survey was first piloted with several practitioners to ensure its clarity and to find possible defects. Based on these pilots, the survey was fine-tuned before its publication. The complete list of final survey questions is provided in [2].

For the survey, participants were chosen opportunistically (convenience sampling) based on the personal connections of the researchers with people involved in IT projects in roles such as developers, reviewers, architects, team leaders, or project managers. Attempts were also made to reach out to more potential participants in focus groups and communities on the LinkedIn[2] and Reddit[3] platforms.

3.3. Interviews

The interview questions were designed based on an initial analysis of the survey data. We formulated the interview questionnaire to collect more detailed information related to the study research questions where more detailed elaboration was required or where the verbatim responses from the survey needed further clarification. A semi-structured interview was employed to ensure some form of uniformity and to prevent the conversation from veering off the intent to provide insights into the research questions.

The first few questions focused on the demographic information of the interviewees. After that, the interview moved on to the security-related questions about the adoption of SSDFs, security practices adopted in the participants' projects, and the reasons for their selection. The interviewees were also asked if there have been any recent additions or changes to their practices in the last couple of years. Finally, they were asked to share their perspective on the trends they see in the adoption of security frameworks and practices. Our complete interview questionnaire is provided in [2].

The interviewees were recruited from 2 main sources. The first source was the participants of the survey, who were asked if they were willing to take part in the interviews. The rest of the interviewees were approached based on personal connections and the relevance of their current role in software development projects. After the participants were recruited, an email was sent informing them of the objectives of the research and to give them a rough idea of the interview questions. All interviews were conducted online and recorded for subsequent transcription.

1. https://www.qualtrics.com/
2. https://www.linkedin.com/
3. https://www.reddit.com/

4. Survey results

4.1. Survey participants

A total of 67 individuals responded to the online survey, from which 37 valid respondents were obtained. We consider responses valid if the participants answered all the key security questions in the survey. The key security questions that are necessary for a meaningful interpretation of the results are questions Q7–Q10 (see [2]) related to the adoption of SSDF in the context of the participant's project.

4.2. Survey findings

Although the order of the sections was different when answered by the respondents, we present the results in the following order: (1) project information; (2) demographic information; (3) answers to the security-related questions. Note that due to the lack of space, we only provide summary details for some of the questions.

4.2.1. Project information. Most of the participants were involved in projects that included the development phase of SDLC. Quite a number also indicated the acquisition and retirement phases. Free text responses for the 2 respondents who selected "Others" were "*Mainly on DevOps for application software*" and "*Regulatory compliance evaluation*". The majority of participants' projects had very small team sizes of 2 to 5 people. A small minority had large teams (more than 50 people in the project).

4.2.2. Demographic information. Regarding participants' roles in their projects, there were 93 responses from the 37 respondents, giving an average of 2-3 different roles per respondent. The most common role assumed by the respondents is the developer role. More than half (22 out of 37) of the respondents have more than 10 years of IT experience, indicating quite an experienced pool of respondents. The majority of the respondents work in large organizations. Most of the participants work in (private) multi-national companies (MNCs) and the next largest group of respondents work for the government/public sector. Regarding the sector, the majority of the respondents work in the IT industry and defense/national security. Our respondents were based in 10 countries, with the majority coming from Singapore and The Netherlands. 7 respondents did not specify their country of residence.

4.2.3. Security questions. About 27% of the respondents indicated that they are aware of their company or project having encountered a cyber-attack on software that was developed or deployed. Most companies have a security team or a chief information security officer (CISO) who is responsible for the security of the developed software. However, about 14% of the responses indicated that there was no specific person responsible for software security.

All of the respondents think that adopting secure practices or an SSDF helps to prevent or reduce the occurrence of cyber-attacks. Notably, most respondents demonstrated familiarity with the baseline SSDFs. Only 4 participants indicated that they were unaware of any such existing

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.

framework. 2 participants selected the "Other" option, and one of them entered "*Sheltered Harbour*" in the free text box, which is not an SSDF but a cyber resilience standard for US financial organizations[4]. Despite the high level of awareness, most participants (21 out of 37) noted that their projects used custom frameworks, highlighting the flexible approaches to software security among organizations.

Most respondents indicated that their projects carry out several secure development activities. Figures 1a and 1b summarize the SSDF activities that the participants' projects follow. Design review, code review, and security risk assessment are the most common ones. These activities are also the top ones among custom frameworks.

Figures 2a and 2b list the reasons for adopting an SSDF. The most cited reason was that the framework selected was comprehensive enough for their project size. Also frequently cited are reasons such as alignment with their development lifecycle processes or being simple to understand and apply. The most cited benefits observed were that more secure software has been developed or that there is a reduced impact of security incidents. The given reasons were similar for both formal and custom SSDFs.

Figure 3 shows the benefits that the participants observed after adopting an SSDF. Among the 37 valid respondents, 7 did not select any option, and one of them indicated "*N/A*" in the free text input. Thus, most of the respondents felt that it was necessary to adopt an SSDF. Still, a small minority felt that it was unnecessary. This could be due to a perception that adopting some individual secure development activities will suffice instead of a complete framework.

Figure 4 summarizes the degrees to which the participants felt that a framework is implemented (as a fraction of the prescribed security activities that are repeated and regularly checked for quality). Our notion of SSDF enforcement is thus related to maturity models, but is weaker and is easier to report in a survey. Most of the respondents (22) indicated that their projects either fully or strictly enforced the security process, while a small minority indicated that they were not or only lightly enforced. The main reasons given for not fully enforcing the processes were insufficient time catered, the lack of guidance, and the associated costs.

Regarding the adoption timeline of SSDFs, a significant number of the respondents (13) reported that their company started adopting a secure development framework more than three years ago, and 9 participants indicated that this occurred 1-3 years ago.

Participants were also asked if there were any changes or additions to their SSDFs introduced in the last 2 years. If there were, participants were asked to briefly describe the changes. Among the 10 respondents who indicated that there was some change to their adopted framework, 7 provided details about the changes that are summarized in the following few points:

- The framework was changed to account for increased compliance audit and stricter checks.
- The framework was changed for specific vulnerability mitigation (e.g. ransomware and Log4J).
- The framework was changed to refine the activities involved.
- The framework was updated due to an evolution of technology but the core of it did not change.

Figure 5 shows the reasons that prompted changes or additions. Based on the responses, the bulk of the changes or additions were due to new legal/government regulations. Many respondents also indicated that they felt there was a greater need to adopt a secure development framework as companies increasingly rely on remote working.

4. https://shelteredharbor.org/

ID	Project role	Based in	Org. size	Industry	Org. type
P1	Product owner	The Netherlands	Small	Healthcare	Private local
P2	Project manager	Singapore	Large	Defense	Public local
P3	Developer, project lead	Curacao	Micro	Finance	Private local
P4	Reviewer	Singapore	Large	Government	Public local
P5A P5B	Developer SecDevOps	Luxembourg	Small	Finance	Private local
P6	Requirements manager	Singapore	Large	IT	Private MNC
P7	Developer	Luxembourg	Large	IT	Private MNC
P8	Developer, project lead	The Netherlands	Large	IT	Non-profit local

TABLE 1: Demographic information of the interviewees. MNC stands for multi-national corporation.

5. Interview results

5.1. Demographic information

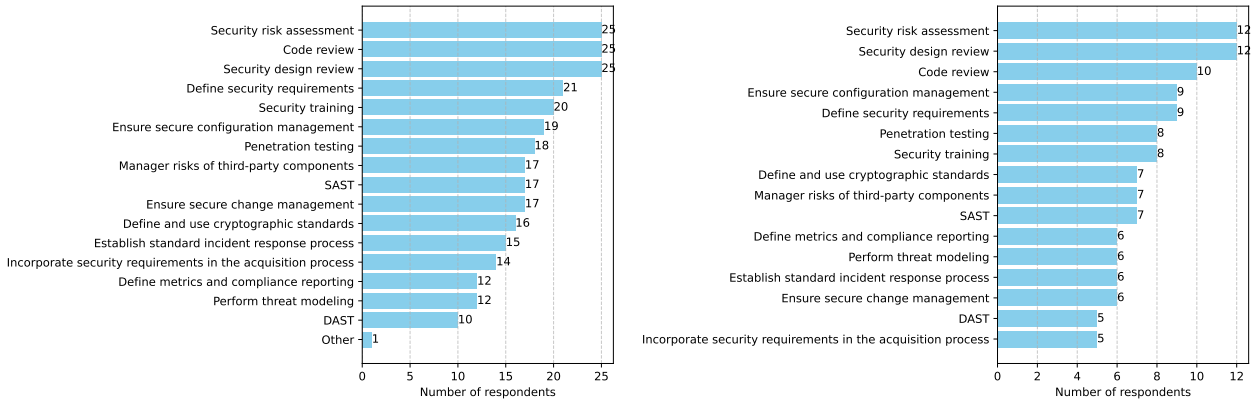
In total, we interviewed 8 IT professionals. Three interviewees were based in Singapore, 2 in The Netherlands, 2 in Luxembourg, and 1 in Curacao. Their demographics are described in Table 1, which assigns an identifier (ID), P1 to P8, to each. The 5th interviewee is assigned 2 IDs (P5A and P5B) as they had recently joined a new employer and described the 2 recent projects from this former and current organization. To preserve participants' anonymity, "they/their" is used, regardless of gender, when referring to the interviewees individually.

5.2. Coding

The interviews were recorded, transcribed, and analyzed. First, open coding was done to label the quotes of potential interest mentioned during the interviews. This has the effect of breaking up the interview transcripts into discrete parts for subsequent analysis. During coding, which was done by the main study author, besides being open and unbiased in reading through the transcripts, software development and cybersecurity activities were specially looked out for.

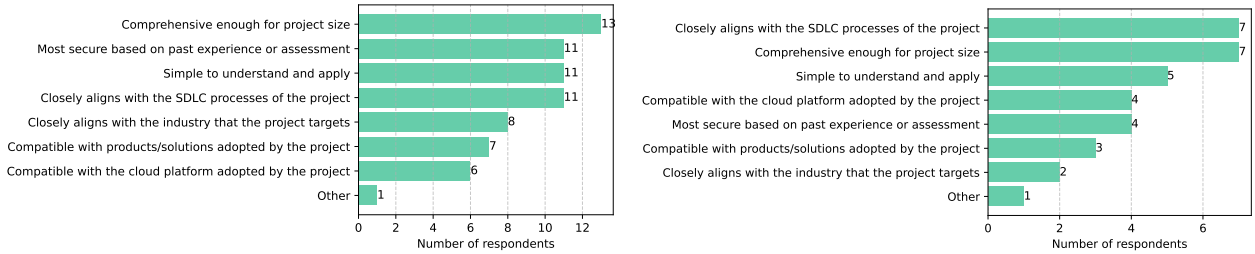
To ensure that the analysis of the data was reliable, the transcript for one of the interviews was coded by an independent coder (another author), and the inter-coder agreement was measured. A test coding was first done to

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.



(a) All responses (b) Responses only related to custom frameworks

Figure 1: What SSDLC stages/activities does your project follow?



(a) All responses (b) Responses only related to custom frameworks

Figure 2: Can you explain the reason(s) for your project adopting the mentioned framework?

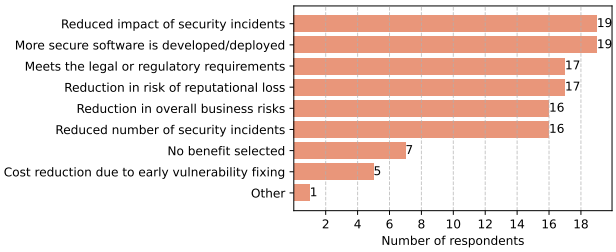


Figure 3: Which of the following benefit(s) have you observed after adopting the mentioned framework?

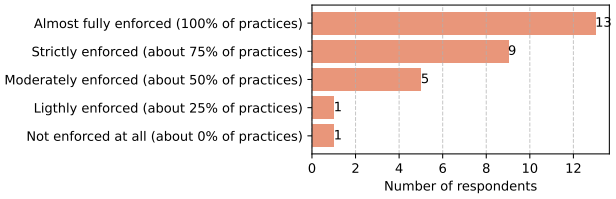


Figure 4: How strictly does your project enforce the security processes of the framework?

ensure the stability of the semantic domains and codes defined and to improve the familiarity of the independent coder with the codebook. After the initial test coding, a final independent coding was done on the sample and the Krippendorff alpha measured was 0.778. As this value is close to the recommended value ($\alpha \geq 0.800$), we conclude that the coding is reasonably reliable. The two coders further discussed the differences discovered and reached an agreement, which further improved the codebook. Finally, a total of 189 unique codes were derived and applied to 268 quotations in the 8 interview transcripts.

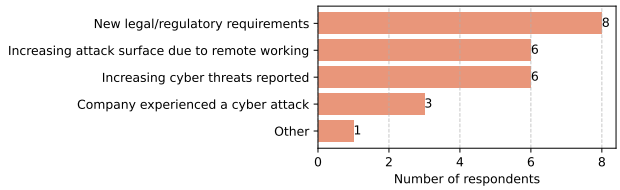


Figure 5: What are the reasons for the changes?

5.3. Interview findings

Analysis of the codes and quotations in the interview transcripts yielded the findings described in the next few sections. The findings are presented grouped by research questions. The full interview questionnaire is provided in [2].

5.3.1. Knowledge and attitudes toward SSDFs. The participants were first asked what they knew about SSDFs. Four of the interviewees (P1, P2, P6, P8) could not describe or name any specific framework, although interviewee P8 mentioned having heard of it.

"*I have heard of it. But to be honest, I've never looked at it.*" – P8

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.

Two interviewees (P3, P7) managed to describe what SSDF is but could not name any specific framework.

"*I think it's a framework, which ensures the development of our software will be secure. It will prevent the development from being attacked by other people.*" – P7

Two interviewees (P4, P5) were able to describe what the SSDF is and managed to name some of the frameworks such as MS SDL, OWASP CLASP and NIST 800-160.

"*They are security activities that fortify the application throughout the normal software development lifecycle.*" – P4

"*I think in general, secure software development is kind of a practice that tries to make sure that security of a piece of software is integrated at every level.*" – P5

5.3.2. Current practices and reasons for SSDF adoption. All of the interviewees indicated that their projects either follow a custom development framework or some secure development activities. None of them adopted a published framework in its entirety. Seven of the interviewees (P1, P2, P3, P6, P5B, P7, P8) expressed that their projects do not follow any specific secure development framework or they are unaware of it. Two of the interviewees (P3 and P7) mentioned that their projects do not follow any specific framework and provided no reasons for that. P1 mentioned that their project does not follow any specific framework, however, they did adopt security practices as they were compelled to follow laws and regulations because of the patient data within their project's systems. P2 mentioned that their project, which uses Commercial off-the-shelf (COTS) products, only adopts some secure activities to ensure that the software is safe to use. However, other projects in their company that develop in-house applications actually adopt an SSDF as a reference and customize it to make it more robust. They customize the framework as it is dependent on their business needs and which component of the framework fits into their criteria. Also due to increasing cyber-attacks, more activities have been added by the different industries as well as their company to ensure that the software is in a better position to be used. P5B explained that they have no policy in place currently because they are writing the policy and determining the rules for the next iteration of releases for their development. P6 pointed out that they believed the technical members of his team follow a secure development framework, although they were not aware of which framework nor the reasons for its adoption. P8 indicated that their project did not adopt any framework, explaining:

"*No, this one (project) doesn't (adopt any framework). However, cyber-security is of the utmost importance in this project. So from all the experience we have in the team with cybersecurity, we try to apply everything to the highest standards.*" – P8

Two interviewees (P4, P5A) revealed that their projects utilize a custom SSDF. P4's rationale was to select and integrate the most suitable activities from standard frameworks. Similarly, P5A emphasized the importance of security in their project, with practices aligned with SSDF principles, despite the absence of a formal software security process.

Security Practice	Freq.
Testing (generic, penetration, black-box, manual, white-box, automated)	10
Code review (generic, automated, manual, static code analysis)	6
Vulnerability scanning & assessment	3
Authentication (generic, multi-factor)	3
Patching	2
Design review	2
OS hardening	2
Protect from insider threats	1
API specification	1
Security configuration and settings	1
Audit logging	1
Audits (by external auditors)	1
Code obfuscation	1
Implement secure architecture	1
Inaccessible from the internet	1
Secure coding	1
Secure data deletion	1
Test-driven development	1
Threat/risk assessment	1
Use VPN software	1

TABLE 2: Secure development practices mentioned in interviews.

All of the interviewees described varying methods and degrees of enforcement of the security processes that their project adopted. Specifically, four interviewees (P2, P4, P6, P8) underscored robust enforcement, encompassing many stringent measures such as a thorough declaration of security status, automated scanning coupled with independent reviews, milestone assessments, dedicated cybersecurity team evaluations, and penetration testing.

Two interviewees (P1, P7) indicated some enforcement by internal reviews/control. Two interviewees (P3, P5A&P5B) noted the absence of enforcement measures.

5.3.3. Common practices adopted. All interviewees gave several secure development practices that their projects adopted, with each respondent naming 3 to 7 activities. Table 2 lists the activities mentioned and their frequency.

5.3.4. Drivers, changes and trends in SSDF adoption. The majority of interviewees discussed recent enhancements to their SSDF aimed at bolstering security measures, with examples including the adoption of tools like the OWASP dependency checker, and the implementation of additional standards and tests to accommodate the rise of mobile apps. Others emphasized practices such as high-fidelity testing, centralized configurations, and transitioning to more secure VPN providers. Looking ahead, several interviewees anticipated a surge in SSDF adoption driven by escalating cyber-threats and regulatory compliance requirements, while others predicted a future where security is increasingly integrated into development tools. Nonetheless, caution was expressed against prioritizing business considerations over security until a breach occurs, emphasizing the need for broader adoption of effective cybersecurity solutions.

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.

6. Discussion

This section discusses the results and findings from both the survey and interviews and compares them with earlier research, where applicable.

6.1. Adoption of frameworks

Our research highlights an increasing adoption of cybersecurity practices among projects and companies, although many do not embrace a formal SSDF. Survey results showed 44% adopting formal frameworks, 44% using custom frameworks, and 12% having no formally defined framework, while interview findings revealed 0% adopting formal frameworks, 22% utilizing custom frameworks, and 78% having nothing they would call a secure software development framework. The difference between survey and interview percentages may stem from the survey's multiple-choice format facilitating framework recall. Notably, while project teams often use security practices, identification of specific frameworks can be challenging. We can conclude that practitioners have a higher recognition of baseline SSDFs when they are presented with their names compared to their ability to recall well-known frameworks without any prompts.

Overall, our results align with earlier research [9] indicating that only about 30% of respondents use formal SSDFs, but our participants demonstrated higher awareness of practitioners compared to results in [9], which might reflect better security awareness among developers today. Additionally, aside from NIST 800-160, OWASP CLASP, and Microsoft SDL, no other formal frameworks (dedicated to secure software engineering) were adopted or recognized by respondents.

6.1.1. Enforcement of processes. The other important finding is that although almost half of the survey respondents indicated that their projects fully enforced the security processes, the slight majority (55%) indicated that they were not fully enforced. The interview responses show a roughly similar trend. This aligns with the findings by Ryan et al. [22] who found that while some organizations adopt certain security practices, they might not spend enough time actually doing them.

6.1.2. Common activities adopted. We can conclude that design review, code review and penetration testing are the 3 most commonly used security activities that were frequently mentioned in both the survey and the interviews. The survey participants also identified the security risk assessment, security training, and defining security requirements as important practices. Notably, these are practices traditionally done earlier in the SDLC. The interviewees, in contrast, were more keen to mention practices related to testing code, fixing vulnerabilities, and system hardening.

RQ1: Most organizations adopt a self-tailored, custom SSDF. Formal frameworks are adopted less frequently. The enforcement degree of the adopted framework varies, with almost half organizations stringently enforcing the chosen set of practices. The adopted SSDFs usually include at least design review, code review and penetration testing as key security practices.

6.2. Adoption considerations

The 4 most common reasons cited by the survey respondents for adopting the frameworks (formal or custom) are: (1) they are comprehensive enough for their project size, (2) they closely align with the development lifecycle processes of their project, (3) they are simple to understand and apply, and (4) based on past experience, they deliver most security. None of the interview participants reported about the adoption of a formal framework (see the reasons they gave behind the adoption of a custom framework in the next subsection).

RQ2: When considering which SSDF to adopt, organizations prioritize approaches that are comprehensive yet simple enough to apply, and that align with the already established development process.

6.3. Custom frameworks

For custom frameworks, the survey respondents prioritized it being comprehensive enough for the project, and its close alignment with the SDLC of the project. Being simple enough to apply, compatibility with the cloud platform adopted by the project and being most secure based on past experience were the other reasons mentioned relatively frequently. In addition, two interviewees mentioned that the reasons for their projects using custom frameworks were the flexibility to select reasonable activities from standard frameworks and also incorporate their own, and the overall importance of security in their project.

RQ3: Organizations using custom frameworks prioritize its comprehensiveness and alignment with the development process. They also pay attention to the cloud infrastructure requirements, their own past experience with this framework, and the flexibility afforded by a tailor-made solution.

6.4. Changes and trends in SSDF adoption

Survey responses highlighted framework amendments driven by technology evolution, compliance audit requirements, and process refinements. Interviewees cited changes such as new tools for scanning software libraries, increased standards for mobile apps and cloud migration, and stricter privacy settings. While responses differed in format, there were consistent themes across the survey and interview answers: new laws and regulations prompt stricter security and privacy measures; escalating cyberthreats drive the adoption of new security activities such as vulnerability scanning tools and incident detection systems; and the technology evolution calls for focus on new platforms and systems.

Assal and Chiasson [4] reported that the common challenges hindering the adoption of best security practices are the division of roles in the project team that does not give developers security responsibilities and the lack of security knowledge and available resources. At the same time, a strong security culture, pressure from regulators, and experiencing a security incident were found to be security-enhancing factors. Gasiba et al. [8] found that

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.

developers while being willing to comply with security coding guidelines, require freedom, knowledge and skills, and time and resources to be able to do it successfully. Broadly, our findings align with these previous results. However, our results have less focus on developers, as our study looked at software development practitioners, which is a broader category.

RQ4: New legal and regulatory requirements and the technology evolution are the main drivers behind the changes in the adopted SSDFs.

6.5. Limitations

We now summarize the main limitations of our study. Firstly, convenience sampling was utilized to gather survey and interview participants, limiting the representativeness of our sample. Most respondents were from Singapore and The Netherlands, reflecting the first author's personal connections. Additionally, our sample is relatively small (37 valid survey responses and 8 interviews). Consequently, the external validity and generalizability of findings need to be further established in subsequent studies.

Additionally, our survey might suffer from priming bias, as using multiple-choice questions might limit the participants' recall of, e.g., other, non-listed frameworks or reasons behind some of their actions. We have partially mitigated this by also conducting interviews that provided additional information.

7. Conclusions

Our study focused on the adoption of SSDFs and utilized a mixed-methods research methodology to unveil contemporary practices and underlying considerations. Despite the availability of formal SSDFs and relatively high recognition of these established frameworks, organizations often opt for custom approaches or select specific security activities, which align more flexibly with the development processes. In our future research, we plan to look more closely into what the adopted custom frameworks look like and how effective they are.

References

[1] Secure software development framework (SSDF). Version 1.1, National Institute of Standards and Technology, 2022. Available at https://doi.org/10.6028/NIST.SP.800-218. Accessed in March 2024.

[2] Survey and interview questionnaire, 2024. Available at https://drive.google.com/file/d/1fB4UnUPautjf7o7PyH1tfhbfdtm0tN5D/.

[3] J. Akhoundali, S. R. Nouri, K. Rietveld, and O. Gadyatskaya. MoreFixes: A large-scale dataset of CVE fix commits mined through enhanced repository discovery. In *Proc. of PROMISE*. ACM, 2024.

[4] H. Assal and S. Chiasson. Security in the software development lifecycle. In *Proc. of SOUPS*, pages 281–296, 2018.

[5] H. Assal and S. Chiasson. 'Think secure from the beginning' A survey with software developers. In *Proc. of CHI*, pages 1–13, 2019.

[6] B. De Win, R. Scandariato, K. Buyens, J. Grégoire, and W. Joosen. On the secure software development process: CLASP, SDL and Touchpoints compared. *Inf. and Soft. Technology*, 51(7):1152–1171, 2009.

[7] J. Fonseca and M. Vieira. A survey on secure software development lifecycles. In *Software Development Techniques for Constructive Information Systems Design*, pages 57–73. IGI Global, 2013.

[8] T. E. Gasiba, U. Lechner, M. Pinto-Albuquerque, and D. M. Fernandez. Awareness of secure coding guidelines in the industry – A first data analysis. In *Proc. of TrustCom*, pages 345–352. IEEE, 2020.

[9] D. Geer. Are companies actually using secure development life cycles? *Computer*, 43(6):12–16, 2010.

[10] M. Howard and S. Lipner. *The security development lifecycle*, volume 8. Microsoft Press Redmond, 2006.

[11] F. Kanei, A. A. Hasegawa, E. Shioji, and M. Akiyama. Analyzing the use of public and in-house secure development guidelines in US and Japanese industries. In *Proc. of CHI*, pages 1–17, 2023.

[12] I. Kirlappos, A. Beautement, and A. Sasse. "Comply or die" is dead: Long live security-aware principal agents. In *Proc. of FC*, pages 70–82. Springer, 2013.

[13] A. Kudriavtseva and O. Gadyatskaya. Secure software development methodologies: A multivocal literature review. *arXiv preprint arXiv:2211.16987*, 2022.

[14] A. Kudriavtseva and O. Gadyatskaya. You cannot improve what you do not measure: A triangulation study of software security metrics. In *Proc. of SAC*. ACM, 2024.

[15] G. Mcgraw. *Software Security: Building Security In*. Addison-Wesley, 2006.

[16] P. Morrison, B. H. Smith, and L. Williams. Surveying security practice adherence in software development. In *Proc. of HoTSoS*, pages 85–94, 2017.

[17] J. C. S. Núñez, A. C. Lindo, and P. G. Rodríguez. A preventive secure software development model for a software factory: a case study. *IEEE Access*, 8:77653–77665, 2020.

[18] OpenSSF. Safeguarding artifact integrity across any software supply chain, https://slsa.dev/, 2023.

[19] OWASP. Comprehensive, lightweight application security process, 2006. Available at https://owasp.org/www-pdf-archive/Us_owasp-clasp-v12-for-print-lulu.pdf. Accessed in March 2024.

[20] A. Ramirez, A. Aiello, and S. Lincke. A survey and comparison of secure software development standards. In *CMI Conference on Cybersecurity and Privacy (CMI)-Digital Transformation-Potentials and Challenges (51275)*, pages 1–6. IEEE, 2020.

[21] R. Ross, M. Winstead, and M. McEvilley. Engineering trustworthy secure systems. Technical report, National Institute of Standards and Technology, 2022. Available at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1r1.pdf. Accessed in March 2024.

[22] I. Ryan, U. Roedig, and K.-J. Stol. Measuring secure coding practice and culture: A finger pointing at the moon is not the moon. In *Proc. of ICSE*, 2023.

[23] SAFECode. Fundamental practices for secure software development, 2018. Available at https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf. Accessed in March 2024.

[24] D. Van Der Linden, P. Anthonysamy, B. Nuseibeh, T. T. Tun, M. Petre, M. Levine, J. Towse, and A. Rashid. Schrödinger's security: Opening the box on app developers' security rationale. In *Proc. of ICSE*, pages 149–160, 2020.

[25] C. Weir, S. Migues, M. Ware, and L. Williams. Infiltrating security into development: Exploring the world's largest software security study. In *Proc. of ESEC/FSE*, pages 1326–1336, 2021.

[26] C. Weir, S. Migues, and L. Williams. Exploring the shift in security responsibility. *IEEE Security and Privacy Magazine*, 2022.

[27] L. Williams. *Cyber Security Body of Knowledge (CyBOK): Secure Software Lifecycle Knowledge Area*. 2019. Available at https://www.cybok.org/media/downloads/Secure_Software_Lifecycle_issue_1.0.pdf. Accessed in March 2024.

Authorized licensed use limited to: Universiteit Leiden. Downloaded on February 12,2025 at 11:27:29 UTC from IEEE Xplore. Restrictions apply.