Relationship Status: *"It's complicated"* Developer-Security Expert Dynamics in Scrum

Houda Naji
Ruhr University Bochum
Bochum, Germany
houda.naji@rub.de

Marco Gutfleisch
Ruhr University Bochum
Bochum, Germany
marco.gutfleisch@rub.de

Alena Naiakshina
Ruhr University Bochum
Bochum, Germany
alena.naiakshina@rub.de

***Abstract*—The high number of cyber threats poses significant challenges, with impactful software exploits ranging from data theft to ransomware deployment. Unfortunately, past research highlighted limited security expertise within development teams. Collaboration between developers and security experts, therefore, emerges as one of the few workable means to address this gap. In this paper, we explore the complex interplay between developers and security experts within Scrum, one of the most widely adopted frameworks which actively promotes collaboration, to shed light on their working relationship, challenges, and potential avenues for improvement. To this end, we conducted a qualitative interview study with 14 developers and 13 security experts. Our qualitative results reveal three communication patterns and five shared challenges between the groups affecting the develop-security expert collaboration. Top challenges include consistent interaction difficulties and the lack of workable means to balance business and security needs. As a result, we found that three core Scrum values (openness, respect, courage) are missing from this relationship. Based on our results, we propose recommendations for fostering a healthy collaboration between developers and security experts, both within and beyond Scrum.**

***Index Terms*—Agile, Scrum, Security, Developers, Security Experts, Collaboration, Relationship.**

I. Introduction

The escalating number of cyber threats poses significant challenges in the digital society, such as data theft or ransomware deployment [20], [46]. In fact, reported software vulnerabilities have quintupled from 5 186 in 2013 to 28 830 in 2023, with over 220 000 vulnerabilities recorded in the NIST database to date [32], likely representing only a fraction of undisclosed vulnerabilities. To further improve structured and secure project management, companies often adopt software development frameworks to increase efficient collaboration among team members and, thus, the delivery of high-quality software products.

While developers should tackle security holistically, there is a limited availability of security expertise within the development community [5], [12]. For instance, in Scrum, one of the most widely adopted Agile frameworks [14], [41], teams are often supported by security experts involving *software security groups* [29], [47], *security champions* [1], [8], [18] or *DevSecOps* [3], [37]. However, past research indicated that their poor integration into the software development process is challenging for both developers [2], [17], [23], [31], [35] and security experts [28], [47], [51]. While previous studies [38], [51] investigated integrating security engineering methods into agile software development, they primarily focused on technical and organizational challenges. By emphasizing the human factor, this study explores the underlying causes of collaboration difficulties and potential areas for improving communication.

We explored how these dynamics unfold within a framework that actively promotes collaboration as one of its core elements, Scrum, the most widely used agile development framework worldwide. In particular, we conducted qualitative interviews with 14 developers and 13 security experts, each lasting an average of 60 minutes, to address the following research questions:

RQ1: *How do developers and security experts in a Scrum setup communicate?*

RQ2: *What challenges arise from the collaboration between developers and security experts within a Scrum setup?*

Despite Scrum's focus on collaboration, our study found a surprisingly complex relationship between the groups. We identified three main communication patterns (see IV-A1a, IV-A1b IV-A1c) and five challenges (see IV-B1, IV-B2, IV-B3, IV-B4, IV-B5) that overburden the collaboration among developers and security experts. The top two challenges relate to interaction difficulties and the inability to reconcile business and security goals. Our results also reveal that companies often play a big role in shaping this relationship and affecting how security is perceived. For instance, the omnipresent focus in Scrum on speed and flexibility often results in security measures being dismissed. In this culture, developers eventually view security as a burden, and receive security expertise and feedback as unwelcome news. Moreover, our findings indicate that fostering a positive working relationship demands deliberate effort, as it is often hindered by various factors. This necessitates assistance from Scrum but also requires approaches beyond Scrum's capabilities. Based on our findings, we offer recommendations to enhance developer-security expert collaboration and suggest directions for future security research. Our key insights are summarized as follows:

- **Scrum Approach:** Utilizing Scrum's values fosters seamless communication and mutual respect, addressing security concerns through continuous learning and collaboration. Our results indicate that product owners should

1558-1225/25/$31.00 ©2025 IEEE
DOI 10.1109/ICSE55347.2025.00120

prioritize security alongside functionality, integrating it into backlog and planning, while Scrum Masters should facilitate collaboration and address challenges within the framework, enhancing product security iteratively.

- **Approaches beyond Scrum:** Addressing collaboration gaps extends well beyond Scrum, with the scarcity of security experts presenting a significant hurdle. Solutions like adopting DevSecOps principles and fostering personal connections between experts and developers can enhance collaboration and mutual understanding.

II. Related Work & Background

A. Scrum

The Scrum framework [22], routed in the agile mindset, focuses on collaboration, learning, and flexibility for high-performance outcomes. Scrum events provide opportunities for inspecting and adapting Scrum artifacts, including the sprint, sprint planning, daily Scrum, Sprint Review, and Sprint Retrospective. The Scrum phases, based on the Scrum Body of Knowledge (SBOK® Guide) [42], consist of initiation, planning and estimation, implementation, review and retrospect, and release. The Scrum team, comprising the Scrum Master, Product Owner, and Developer Team, collaborates cross-functionally to achieve product and sprint goals. Developers are held accountable for delivering a usable product increment each sprint, adhering to the Definition of Done (DOD). The Product Owner maximizes product value, and the Scrum Master ensures adherence to Scrum principles and values while removing impediments. Stakeholders play a significant role, contributing to defining goals regarding quality, regulations, and security. These goals influence the Scrum process. Scrum artifacts, including the Product Backlog, Sprint Backlog, and Increment, provide information about the product, required work, and completed tasks. Scrum's values of commitment, courage, focus, openness, and respect foster collaboration and contribute to project success. Scrum's principles, as outlined in the SBOK [42], emphasize iterative development, self-organization, and continuous improvement. More details about the Scrum framework are included in the Supplementary Materials (Section VI, p. 3).

B. Organizational Security Culture

Past research identified organizational culture and processes impacting software security [6], [17]–[19], [23], [25], [35], [44], [45], [52]. Alnatheer et al. [4] proposed ways of defining proper measurements for information security culture, with the main factors identified being top management involvement, policy enforcement and security training. In an interview study, Assal and Chiasson [5] found that security compliance and best practices were often not followed and identified several factors, like the company's organizational culture, influencing overall security compliance. Veiga et al. [13] explored different perspectives academia and industry have regarding security culture and concluded that the involvement of management and mutual trust between employees play a big role in the success of security. Karlson et al. [21] investigated the security compliance of Swedish office workers and found a relation between compliance and security culture, suggesting that a more bureaucratic culture can lead to more compliance. Lopez et al. [26] conducted an ethnographic study with software developers and found that while security can be increased through integrating secure behavior in the usual workflow, proper processes, and context-driven training, it still requires integration into the organizational context.

C. Developers and Security Experts

While consulting security experts might improve developers' security practices in their daily work, previous work suggested that they often do not interchange. For example, Weir et al. [12] found a high level of requested support regarding security, while only 14–22 % of Android developers had access to security experts. Tøndel et al. [51] investigated factors impacting the prioritization of security in agile software development. The lack of security experts, security awareness, and security routines in the SDP negatively impacted the prioritization of security. Rindell et al. [38] investigated software security engineering activities in agile software development in a survey with 61 professionals. They found that most security activities take place in the SDP's later stages, although the effectiveness of the security activities was perceived to be most effective in earlier stages. Mazurek [28] identified the problem of experts trying to maximize security while disregarding their organization's situation or culture. To improve secure software development, Palombo et al. [34] and Tuladhar et al. [49] suggested involving security experts in the development process, such as co-creation and situated learning. However, Thomas et al. [47] found that communication is a big challenge regarding the interaction between developers and security experts. Most communication occurred through the ticketing system or bug tracking, while direct communication was rare. To the best of our knowledge, this is the first exploratory interview study with developers and security experts investigating both perspectives on software security collaboration in a Scrum setting.

III. Methodology

We conducted 27 semi-structured interviews to gain an in-depth insight into the working relationship between security experts and developers within a Scrum setup. Participants had to complete a pre-questionnaire before our interviews. We embedded an option for scheduling an interview to reduce the complexity of coordinating across different time zones. We used an online conference tool supporting video communication and only recorded the audio track, Zoom [54]. We left participants free to choose whether to have a video chat or just audio. Researcher R1 conducted all interviews, while Researcher R2 assisted with taking notes. An overview of the study can be found in Figure 1. The pre-questionnaire and the interview guidelines can be found in the Supplementary Materials.

A. Interview Design

Instrument Development. We developed two interview guidelines for security experts and developers, respectively.

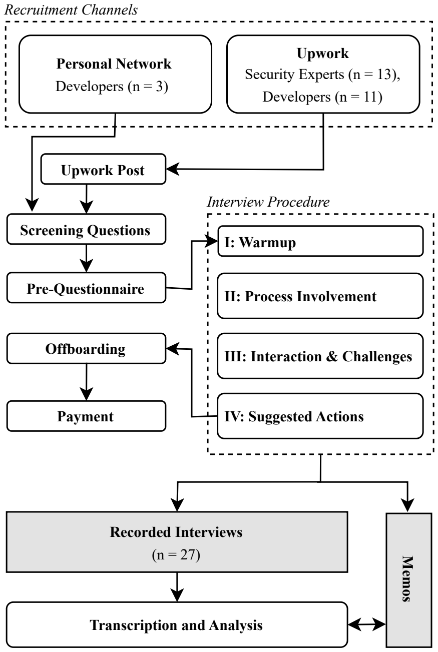

Fig. 1: Overview of the study

Both included the same questions adapted according to the participants' roles. However, we delved more deeply into retrospective-related questions with developers, given that security experts typically don't form a Scrum team. We started our interview guideline with a *(I) Warm Up*, asking about participants' initial thoughts on the research topic. *(II) Process Involvement*: We continued with the first significant part of our interview guide, which aimed to gain insights into the participants' experience of the level of involvement of security experts through the different Scrum phases. *(III) Interactions & Challenges*: In the next part of the interview guide, we asked participants about their working environment, general challenges and responsibilities, and experiences with the social interaction between security experts and developers. *(IV) Suggested Actions*: Afterward, we asked for participants' thoughts on improving the collaboration or addressing previously mentioned issues. Finally, we asked them what they wanted to share with the other party (security experts or developers).

B. Pilot Study

We internally piloted the interview questions with two security researchers and conducted two pilot studies with two developers recruited on Upwork.com. We did not change our interview design based on the two pilot studies. Further, we initially considered Scrum Masters independent observers of the relationship between developers and security experts within a Scrum framework. We conducted two pilot interviews with experienced Scrum Masters and found limited insights into the developer-security expert relationship by being directed to both roles for a more comprehensive understanding. Consequently, we decided to narrow this study's focus to developers and security experts to refine the scope of our investigations.

C. Participants

1) Recruitment.: We used our professional networks and Upwork for participant recruitment, targeting developers and security experts. Security experts were required to have prior experience collaborating with developers in a Scrum framework. Developers met the criteria if they had experience collaborating with security experts in a Scrum setup and had worked on software product security. The interviews were conducted in English. Thus, fluent English communication was a prerequisite for all participants. We reached out to 25 security experts on Upwork and 47 developers (3 of whom were recruited from professional networks, and the remaining 43 from Upwork). Out of these, 16 developers and 15 security experts agreed to participate in the final study. It's important to note that we excluded two Upwork developers from the initial 16, as they were part of the pilot candidates. Unfortunately, two security experts did not attend the scheduled interviews.

2) Participants' Demographics.: We conducted interviews with 14 developers (D_1 - D_{14}) and 13 security experts (S_1 - S_{13}). Our participants were from 15 countries, including Germany, Pakistan, India, the US, Lebanon, Serbia, Turkey, Australia, and the Netherlands. Of the interviewed participants, 24 were men, 2 were women, and one identified as non-binary/third gender (see Table I). The age range was 23 to 60 years, with an average age of 33.59 years. Participants had professional experience in the software industry, averaging 8.61 years, with a maximum of 24 years. All participants reported at least a college-level education, with 15 (55.6%) holding a Bachelor's degree and 12 (44.4%) holding a Master's degree. Further, participants held diverse roles and seniority levels, including Lead Security Research Engineers, Senior Developers, and Consultants. Five developers identified themselves as Development and Operations (DevOps) practitioners. DevOps has led to a significant shift aimed at removing boundaries between software development and software operations [24] and is currently widely adopted in the industry [27]. Participants were compensated with $60 (€60 in Europe) Amazon vouchers.

D. Data Analysis

For data analysis, we employed Thematic Analysis following Braun et al.'s six-phase approach [11]. Both researchers (R1 and R2) attended each interview and discussed notes to familiarize themselves with the data (Phase 1). They collaboratively and inductively developed an initial codebook after coding a random set of seven interviews (Phase 2). R1 coded all interviews and R2 recoded them after jointly refining the codebook (see Table 3, Supplementary Materials), after which R2 recoded all interviews. Conflicts were resolved immediately in a direct exchange between the two coders. Employing memos is considered a vital strategy in qualitative research for fostering thorough data exploration and ensuring the continuity of the research process [10]. Thus, throughout the research

TABLE I: Participants' demographics.

Gender		
Men	24	88.9%
Women	2	7.4%
Non-binary / Third Gender	1	3.7%
Countries		
Pakistan	5	18.6%
Germany	4	14.8%
United States	3	11.1%
India	3	11.1%
Netherlands	2	7.4%
*Other	10	37.0%
Age [years]		
Min.	23	
Max.	60	
Mean (Std.)	33.59	±7.59
Median	32	
Industry Experience [years]		
Min.	2	
Max.	24	
Mean (Std.)	8.61	± 5.56
Median	8	
Education		
Bachelor's degree	15	55.6%
Master's degree	12	44.4%

process, both researchers used memos to organize data into themes, revealing patterns and linking findings to the research questions (Phase 3). Both researchers identified core themes: working relationship, challenges, and potential improvements (Phase 4), achieving saturation after 24 interviews (Phase 5). We report our findings based on these themes in Section IV (Phase 6).

E. Ethics and Data Privacy

Ethical approval for our study was obtained from our Institutional Review Board (IRB). Participants were given a comprehensive consent form outlining the study's purpose, data utilization, duration, and associated risks. We ensured strict compliance with the General Data Protection Regulation (GDPR). Participants were informed of their right to terminate the interview at any point. We provided assurance that only de-identified data and quotations would be used in publication. Prior to the interview, participants were given a detailed overview of the study's main topic.

F. Limitations

Our study results need to be interpreted considering the following limitations. First, our sample does not include a representative cross-section of all developers or security experts within agile development. Thus, we cannot claim generalizability. Second, discussing interpersonal relationships within the work context might be challenging for participants. Despite our efforts to build a relationship of trust in the interviews, participants may have omitted or distorted content. Third, our findings are limited to Scrum in agile development. Other development models could produce different results.

IV. Results

In this section, we detail insights from interviews with developers and security experts, focusing on their collaboration across the various phases of a Scrum project (see Section IV-A2). Specifically, we asked developers and security experts questions to gain insights into the intricacies of their collaborative processes during Scrum meetings. While emphasizing qualitative analysis, this research includes the frequency and distribution of themes by reporting the number of participants who made relevant statements when necessary to lend weight to notable patterns within the sample.

If more than 5 participants mentioned a theme, we list a subset of participants. Additional tables and detailed analysis can be found in the Supplementary Materials.

A. Working Relationship

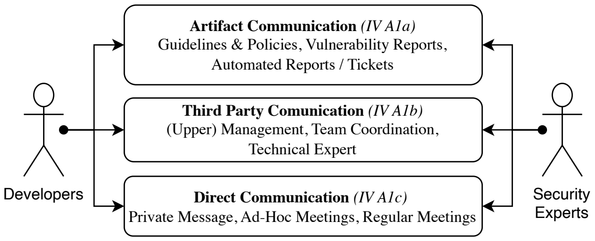
Artifact Communication (IV A1a)
Guidelines & Policies, Vulnerability Reports, Automated Reports / Tickets
Third Party Comunication (IV A1b)
(Upper) Management, Team Coordination, Technical Expert
Direct Communication (IV A1c)
Private Message, Ad-Hoc Meetings, Regular Meetings
Developers
Security Experts

Fig. 2: Observed communication patterns of developers and security experts.

*1) **Communication Patterns in Developer-Security Expert Collaboration**:* Based on participants' statements, we derived developer-security expert communication patterns from their perceived interaction with developers and security experts. Figure 2 illustrates the different communication patterns by listing them from top to down and left to right according to most to least prevalent.

a) Artifact Communication:

Guidelines and Policies. Starting a software project requires having clear security guidelines as they set the foundation, according to one participant stating that "*for every developer, those are all general rules, and every developer should be aware of that*" (D_2). Except D_{11} and D_{14} indicating a lack of security guidelines in their respective companies, most participants affirmed the presence of such guidelines. Eight participants (e.g., D_1, D_5, S_3, S_8, S_{10}) mentioned that security experts take the lead in creating these guidelines. However, other participants (e.g., D_4, D_9, D_{12}) highlighted that the creation process could involve collaboration with architects, CTOs, and CISOs. Two security experts (S_4, S_{13}) pointed out that they rely on the international institutions' guidelines. Developers actively influence these guidelines, contributing their insights throughout the process (e.g., D_6, D_1, S_3, S_6). However, D_{10} highlighted that, despite developer influence, "*the security team has the last word.*" Further, D_1 noted that security incidents might trigger updates to the guidelines as lessons learned.

Eight developers (e.g., D_4, D_6, D_9, D_{10}, D_{11}) highlighted the benefits of following security guidelines, such as preventing vulnerabilities and bringing them "*a level of comfort*" (D_{11}) (e.g., D_{13}, S_5) by helping to "*clear any confusion regarding* *security rules and set the record straight*" (D_{10}). However, some participants (e.g., D_4, D_{13}, S_3) stated that developers fall short of fully following these guidelines. This could be due to many reasons, including disagreement with the guidelines (S_4), company or project perspectives (e.g., D_9, S_2), developer awareness (e.g., S_3, S_{12}), non-enforcement of security guidelines (e.g., D_3, S_8), time pressure (e.g., S_3, D_1), perceived annoyance (e.g., S_4, D_4), or finding guidelines impractical or outdated (e.g., D_{13}, S_4). The absence of guidelines may result from a lack of security experts or a company's failure to prioritize security (e.g., D_{11}, D_{14}, S_7).

Vulnerability Reports. Alerts triggered by automated systems were often mentioned as the initial starting point of communication between developers and security experts as D_{10} described:

> *Some tools give you some security warnings or alerts. Or there is, for example, a big security issue in one typical library, like there was with log4j. So, an alert is triggered somewhere in a tool or from the security team, and everybody has to react.* (D_{10})

Automated Reports/Tickets. Fourteen participants (e.g., D_1, D_7, D_{10}, S_1, S_{13}) described that some aspects of their collaboration, such as scans and checks, are now automated and in the event of an issue a ticket is automatically generated to the development team. While the Scrum framework encourages the adoption of automation and tools to enhance development efficiency and quality [42], this focus on automation has minimized the direct involvement of security experts, with D_7 noting that "*security is getting like in the back of our minds.*" In line with Scrum, S_{13} highlighted the importance of automation by stating "*one day things will become like a crazy if you're not doing an automation, because you cannot do everything manual, the product is becoming a bigger and bigger*". Another security expert confirmed the need for automation by stating that they are simply "*trying to make your life easier. We don't want you to deploy any bad code*" (S_9), conveying a message to the developers. This communication pattern depended heavily on the companies' structures and processes and the tools used in the SDP. It appears to mainly occur during the release phase when security is integrated into the pipeline (e.g., D_2, D_7, D_{12}, S_{13}). A raised security issue on the management level might find its way through the product owner in the usual ticketing system. In contrast, automated action from a security tool within the pipeline might trigger alerts on the developers' and the security experts' side. It might be fixed immediately before continuing with the deployment.

b) Third Party Communication.: We observed that a third party was often involved, acting as a gateway between the developers and the security experts:

(Upper) Management. Communication often happens through CTO or project manager (e.g., S_4, S_{10}, D_7) who oversee the broader aspects of security integration and project progression.

Team coordinator. It strongly depended on the respective project and the company structure whether the exchange took place close to the team (e.g., via the product owner or a team lead) or even more distanced from the development team:

> *It's like product owners sitting in between [Security experts and developers]. Everything is routing through that person because that person needs to orchestrate what is being built, right?* (D_{13}).

Technical expert. Further, security-relevant information was also passed through technical experts, like architects, who are not always part of the same Scrum team. Interestingly, the participants did not mention the Scrum Master in this third-party communication, despite their central role in facilitating communication and removing impediments within Scrum teams.

c) Direct Communication.: Only eight participants (e.g., S_3, S_6, D_1, D_2, D_5) described that they had a direct communication with the other party. In most cases, a direct verbal interaction only happens if a problem arises that cannot be solved without the expertise of the other party (e.g., if an issue was raised from the security pipeline that is not understood by the development team). S_8 distanced himself from the term "*collaboration*" and named this pattern a "*need basis relationship.*"

Private Message. Some participants(e.g., D_6, S_3) stated that one way of communicating would be via internal company messaging programs such as Slack. D_6 noted that this way of communicating was not optimal and would prefer to replace it with meetings.

Add-Hoc Meetings. S_3 even told that he specifically reached out to development teams in advance before bringing issues to the developers in a later stage to first settle the basis for a relationship: "*For me, it's first try to build a good communication with them, I would say. And talk to them on a very lighter note*" (S_3).

Regular Meetings. Some participants (e.g., S_1, S_3, S_{10}) mentioning that they had weekly meetings. Note that their direct communication often occurs outside of the formal Scrum events. S_4 expressed a desire to attend daily Scrum, which is one of the events, but they "*just don't get invited to them*" and only observe it "*from afar.*" They highlighted the importance of joining this event early in the sprint helps "*to have factor in that risk of, hey, do we have the rights? Are we doing this from a security standpoint?*" (S_4). The Sprint Review and Sprint Retrospective, two other Scrum events that could foster direct communication between security experts and developers, appear to lack the participation of security experts. Out of the 27 participants, only 4 participants (D_{12}, S_5, S_7, S_8) mentioned security experts' involvement in the Sprint Review, while 6 (e.g., S_4, S_5, S_6, S_{11}, S_{12}) participants noted the participation of security experts in Sprint Retrospectives.

*2) **Security Expert Involvement in the Software Development Process (SDP).**:* We examined security experts' involvement in the Scrum-based software development process (SDP) phases—Initiation, Planning, Implementation, and Release—and the modes of communication used in each phase. Table II summarizes how many participants mentioned security experts were involved in the different processes.

TABLE II: # of participants mentioned security experts being involved in the SDP

Level of Involvement	Initiation	Planning	Implementation	Release
High Involvement	14	12	17	17
Variable Involvement	5	5	5	7
No Involvement	8	10	5	3

Initiation and Planning Phases. During the initiation phase, 14 (51.9 %) participants noted high security expert involvement, while 8 (29.6 %) reported none, largely due to company structure and culture (e.g., D_1, D_2, D_5, D_7, S_1), and project type (e.g., D_6, D_{13}, S_8, S_9, S_{12}). Those reporting no involvement often cited low prioritization of security at this stage (e.g., D_4, D_9, D_{10}, S_3,S_{10}). Communication is mostly occurring through *Third-Party Communication* (e.g., D_6, D_7, D_{14}, S_4, S_5) in stakeholder meetings. *Direct Communication* between security experts and developers is uncommon at this stage. In the planning phase, 12 (44.4 %) participants cited security expert engagement, while 10 (37 %) noted none, mainly because security wasn't a concern at this stage (e.g., D_{10}, D_{12}, S_4). Engagement occurs when guidance is needed for handling the project's security aspects (e.g., D_2, D_7, D_{14}, S_3, S_{11}). Communication involves *Direct Communication* (e.g., D_1, D_2, D_{14}, S_3, S_{12}), for creating user stories and requirements, and *Third-Party Communication* through intermediaries like architects or product owners (e.g., D_5, D_7. S_1, S_{10}, S_{11}).

Implementation and Release Phases. Security experts' involvement during both the implementation and release phases is primarily influenced by the integration of security into the development process. In the implementation phase, 17 (63 %) participants reported security experts' involvement, primarily through testing and automated security checks (e.g., D_2, D_7, D_{11}, S_3, S_{11}) and providing guidance it (e.g., D_6, D_8, D_{13}, S_5, S_{12}). However, some participants noted their absence due to a lack of outreach between teams (e.g., S_2, S_9, D_4, D_9, D_{14}). Communication predominantly occurs through *Artifacts Communication*(e.g., D_1, D_2, S_3, S_5, S_7), such as tickets and code reviews, with minimal *Direct Communication*, which only happens when a problem arises. Similarly, in the release phase, 17 (63 %) participants also noted security experts' involvement due to security being part of the development process (e.g., S_2, S_3, S_8, D_7, D_9). Communication remains mostly through *Artifacts Communication* (e.g., D_1, D_2, D_4, S_6, S_8), with *Direct Communication* being rare.

*3) **Responsibilities in Developer-Security Expert Collaboration**:* Scrum promotes a self-organizing team dynamic without rigid roles. We explored how security experts and developers collaborate, focusing on their perceptions of security responsibilities and role definitions.

Developers' Responsibilities. Seven developers discussed their responsibilities in security, with some (D_2, D_9, D_{12}) focusing on writing secure code and understanding basic security, while others (D_3, D_7, D_{11}) emphasized overall environment security. Two developers (D_8, D_{10}) acknowledged their limitations compared to security experts, and only three (D_3, D_5, D_{12}) mentioned fixing security issues. In contrast, six security experts (e.g., S_2, S_4, S_5, S_8, S_{10},) highlighted the development team's responsibility for fixing security issues, stressing the importance of following their recommendations and having basic security knowledge (S_9, S_{12}).

Security Experts' Responsibilities. One developer (D_{12}) highlighted the need for continuous availability of security experts. Four (D_2, D_9, D_{10}, D_{11}) saw them as consultants, while six (e.g., D_1, D_4, D_5, D_9, D_{10}) expected them to oversee development and prevent mistakes. Five (D_1, D_7, D_9, D_{13}, D_{14}) emphasized their role in communication and coordination. By contrast, security experts understood their role in coordination and support, with some mentioning monitoring akin to policing. However, some, like S_1 stressed supporting organizational growth rather than policing. Regarding decision-making, seven (e.g., D_6, S_1, S_2, S_4, S_{12}) noted non-technical individuals usually have the final say, while two (D_{13}, S_{11}) favored a collaborative approach involving management, architects, and security experts.

*4) **Developers and Security Experts Embrace Collaboration.**:* The working relationship between developers and security experts is underscored by mutual respect and collaboration, with both groups motivated to enhance their interaction and communication. Developers (e.g., D_1, D_2, D_{12}) expressed respect for security experts and their specialized work and viewed collaboration with security experts as an opportunity to improve their skills and gain valuable insights. Five developers (D_2, D_4, D_6, D_7, D_{13}) and three security experts (S_{S6}, S_{11}, S_{12}) shared a common goal of enhancing security through collaboration. Three developers (D_7, D_9, D_{11}) appreciated the clear requirements and straightforward communication from security experts, which facilitated smooth collaboration. Security experts valued developers' collaborative nature and openness to feedback, especially in challenging situations (e.g., S_2, S_5, S_7). Messages gathered from 22 participants highlighted the need for increased communication and collaboration. D_{11}, D_{12}, and D_{13} advocated for more engagement from security experts, while D_2 and D_4 emphasized security's importance. Security experts stressed promoting security awareness and better communication (e.g., S_8, S_3, S_5), S_{11}) viewing their role as enablers to help developers and prevent early mistakes.

B. Challenges in Developer-Security Expert Collaboration

Although one of Scrum's principles is *collaboration* [42], participants faced numerous challenges during their collaboration. Surprisingly, security experts and developers highlighted the same five challenges, each providing insights from their point of view. Notably, the top two challenges, "Interaction Difficulties" (ranked 1 for security experts and 2 for developers) and "Balancing Business Goals and Security Needs" (ranked 2 for security experts and 1 for developers), were unanimously acknowledged by both groups. An overview of the challenges' frequency in each group is presented in Figure 3.

1) Interaction Difficulties.: **Tensions and Misunderstandings.** Participants reported challenges in the interaction be-

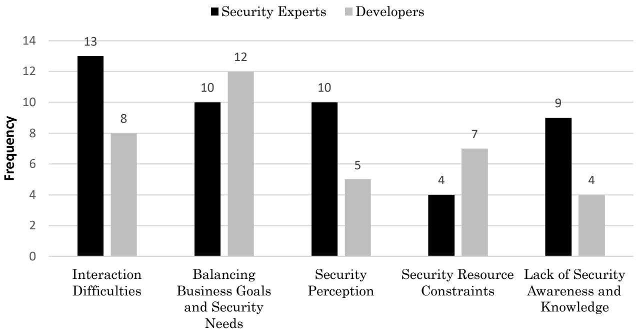

Fig. 3: Frequency overview of challenges faced by Developers and Security Experts.

tween security experts and developers. Ten security experts (e.g., S_1, S_2, S_8, S_9, S_{12}) faced difficulties reaching out to developers, who resisted sharing information and often overlooked the value of the security role. Recognition for their work requires extra effort, as one expert noted: "*they will laugh in your face because you don't know*" (S_1). On the other hand, four developers (D_8, D_9, D_{10}, D_{12}) perceived security professionals as aloof and isolated, with a noted lack of networking efforts: "*Security guys usually hide*" (D_8). D_9 also mentioned that developers feel "*afraid*" to contact security experts, citing a "*minimal relationship between them*" as a cause. The clash of perspectives becomes evident, as expressed by a participant: "*there is a security expert who thinks they know everything and a technology expert who thinks they know everything really kind of clash heads*" (S_4). D_{12} shared this sentiment, depicting security experts as authorities not always open to compromise. Cultural differences and resistance to change further complicate the collaboration. Developers often felt judged by security experts, a sentiment echoed by D_1 and S_1 in the collaboration. Gender challenges also emerged as a factor, with S_4 highlighting the resistance faced, such as the refusal to be directed by a woman. Ego preservation, particularly among men colleagues, proved to be a common challenge for S_4, reflected in the belief that "*99% of the time I'm working with a man who doesn't want to have his ego crushed*".

The Blame Game. D_1 and D_{11} expressed frustration over time-consuming security processes, feeling constantly monitored, as one participant puts it, "*breathing down my neck*" (D_1). On the other hand, S_{12} faces resistance in engaging with documentation, as developers "*wouldn't just make that extra effort of going through that document.*" S_5 struggled with integrating security measures as developers focused on functionality rather than security concerns. Simultaneously, security experts expressed challenges in overcoming resistance, particularly when developers were "*forced under pressure by their managers and these managers, under pressure by salespeople*" (S_{13}), prompting requests for security exceptions.

Technical Terminology Hurdles. Scrum highlights the importance of a clear communication and effective team work to deliver value. However, many participants struggled with effective communication and mutual understanding, due to their different technical backgrounds. Eleven participants (e.g., S_3, S_4, S_8, D_8, D_{12}) identified navigating technical terminology as a major hurdle, hindering the collaboration and preventing them from forming a solid partnership. S_3 frustration over the necessity to "*translate everything,*" highlighting the additional effort required to guide developers, as one participant noted they "*need so much hand holding*" (S_{12}). Developers echoed this need for translation, constantly feeling forced "*to explain everything, what's happening and what is not*" (D_8), which could be time-consuming because "*sometimes things get lost in translation*" (D_{13}). These challenges with technical terminology led to frustration and contributed to a "*blame game*" fueled by the challenges of communication when "*not speaking the same language*" (S_3).

2) Balancing Business Goals and Security Needs.: Scrum encourages prioritizing customer value and responding to market demands. However, these priorities may not always align with security needs, leading to challenges in finding a balance.

Functionality First, Security Second. Thirteen participants (e.g., D_9, D_{13}, S_4, S_{10}, S_{13}) mentioned that skipping security for business reasons is a common practice, as highlighted by D12: "*in most organizations, you develop first, security later.*" S_9 highlighted the priority given to functional products, stating that security is "*nowhere in the picture. It's working as long as it's working.*" This choice is often influenced by management's tendency to deprioritize security (e.g., S_{13}, S_5, D_{12}). Companies, driven by a need for speed and features, tend to relegate security to the bottom of their priorities, resorting to supplying temporary fixes or "*band-aids*" to address immediate concerns (e.g., S_8, D_5, D_9). The presence of strict timelines enforced by business needs can be compromised when faced with security requests, creating a conflict between meeting deadlines and ensuring proper security measures, as mentioned by two participants (D_{13}, D_6). Developers expressed pressure to invest time in learning about security, which inevitably impacts project timelines (e.g., D_5, D_6). Despite recognizing the urgency of making everyone aware of security concerns (e.g., S_5, S_1), security often takes a backseat due to budget constraints (e.g., S_{13}, S_8, D_9). Even when security concerns were escalating to team leads and managers, it did not guarantee prompt attention or resolution, as expressed by S_{10}: "*we escalated to the team lead first, and he told me that, yeah, I will get it sorted, but they didn't start working on it then*".

Project Security on a Budget. Twelve participants (e.g., S_2, S_{12}, D_1, D_6, S_{13}) emphasized the budget plays an essential role in the existence of security in the project as it is seen as expensive. S_2 stated that in a small project, the budget does not allow "*to have an IT security expert full-time.*" Even some clients refuse to "*pay ten bucks for something. They don't even have the budget for that*" (S_{12}).

Inefficiencies from Outdated Security Guidelines. D_1 encountered security task challenges, facing delays due to underestimated complexity by security experts perceiving solutions as straightforward. However, the reality proved more complex and time-consuming for developers, exemplified by D_7's struggle taking them "*almost two to three weeks*" to update. Outdated security solutions led developers to invest time in implementation, only to discover later that "*they didn't even release a patch for it*" (D_1), resulting in time lost. However, the challenge of security experts not being up to date indicated a link to outdated security guidelines (e.g., S_4, D_4, D_{10}) with S4 mentioning, "*It's been a publishing issue and a lot of like red tape around how quickly a business organization can push out a piece of guideline or a framework.*" D_1 expressed frustration, feeling envious of security experts who, as perceived by developers, only need to identify issues without handling implementation complexities.

3) Security Perception:

From Afterthought to Add-On. Perceptions of security are crucial in defining the collaboration between security experts and developers. Participants (e.g., D_{10}, D_{11}, S_3, S_9, S_{12}) noted that the project's nature determines the extent of security expert involvement. The project driver plays a significant role if security experts are involved with S_3 stating "*when the driver is business, the technical teams are mainly focused around the delivery.*" As security is not seen as the priority, D_{10} explained "*all the product has to be beneficial or has to have some value. So priorities are always that everybody should be aware, but at the beginning, is not the first priority.*" This viewpoint stems from the belief that security is an "*afterthought*" and an "*add-on*" that generally goes unnoticed or disregarded (e.g., S_7, S_9, D_{11}). This also resonates with the observation that only 4 participants (D_7, D_{10}, D_{11}, S_3) mentioned that security is included in their Definition of Ready (DoR), with one developer stating, "*We hardly take into consideration most security aspects during story development*" (D_{12}). Further, only 7 participants (e.g., D_2, D_4, D_7, D_9, D_{12}) claimed that security is part of the Definition of Done (DOD). S_4 noted that if security was part of the DoD, it was treated as a mere "*tick box sort of compliance. Yes or no?*" and emphasized that "*not every business and organization runs those tabletop exercises. They just immediately will say, yes, it's been run through a security expert.*"

Security as a Roadblock. Despite Scrum's value of courage in addressing challenges directly, developers often perceive security tasks as daunting. Security is often seen as an impediment slowing down project progress, as expressed by 6 participants (e.g., D_1, D_8, D_6, S_5, S_8). Developers (e.g., D_1, D_5) usually prioritized efficiency, aiming to focus on tasks and release their work to "*be done with it*" (D_1). Hence, developers tended to avoid working on security-related tasks (e.g., D_5, D_{10}, S_1). In addition, some participants (e.g., D_9, S_4, S_{11}, D_7) noted that developers find security measures challenging. As a result, security experts might be seen as obstacles rather than facilitators in the development process, affecting collaboration dynamics. (e.g., S_2, S_3, S_5).

4) Security Resource Constraints.: In Scrum, the concept of cross-functional teams emphasizes that all the skills necessary to deliver a product increment should be present within the team. However, it seems when it comes to security skills, there may be a gap or shortage within the team.

The Impact of Security Expert Shortage. Six participants (e.g., S_8, S_7, D_9, D_{12}, D_{13}) mentioned that the lack of security professionals makes hiring a challenging task. Additionally, six participants (e.g., S_7, S_8, D_8, D_{11}, D_{12}) highlighted the challenge posed by the shortage of security experts, noting that this challenge becomes apparent as they find themselves outnumbered by the many development teams and projects they have to manage. This imbalance makes it challenging for developers to quickly reach security experts, leading to extended waiting times for responses (e.g., D_8, D_{11}, D_{12}). Incorporating DevSecOps into the workforce is an even more significant challenge due to the limited availability of qualified experts (D_9). DevSecOps addresses the challenge of ensuring seamless security integration within the rapidly accelerated software release cycles of DevOps [30] by emphasizing the imperative to incorporate security at every stage of the SDP. Additionally, if a company prefers to educate a developer in security, it will require "*to complete around six certifications and pretty much around six months of other training and work together with the team*" (D_1). S_9 and S_{11} pointed out that even when companies employ qualified security experts, their services come at a high cost, demanding a considerable budget allocation. The competitive market for security experts worsens the problem, with some experts leaving for higher-paying opportunities, creating a "*pain point*" (D_1) for developers.

Heavy Workload of Security Experts. Security experts also face challenges due to limited resources, especially with the rise of third-party applications and the shift to remote work. These challenges make it increasingly difficult for them to monitor and keep track of everything effectively (e.g., S_8, S_9). The toll of these challenges on security experts is evident, with one participant expressing burnout and ultimately leaving their company due to the overwhelming demands (S_7). Accordingly, three participants (D_{11}, D_{12}, D_{13}) stated that security experts are stretched too thin on too many projects, which makes it difficult to include them in everything with S_9 complained that this is "*one of the reasons that I kind of got burnt out*" (S_9).

5) Lack of Security Awareness & Knowledge.: The lack of security awareness significantly challenged the collaboration between developers and security experts due to a lack of education and knowledge about security measures (e.g., D_4, D_5, D_{10}), creating a gap that holds back their ability to advocate for necessary resources for the project (D_{10}). Two participants (S_5, S_3) claimed that the knowledge gap could become a barrier for security experts trying to communicate effectively with developers. The "*skill gap*" in security knowledge is a persistent issue (S_3), and developers might not fully comprehend the impact of certain security measures. Security experts stressed the crucial adherence to compliance and other standards (S_3).

TABLE III: Categories of developer-security expert collaboration improvements.

Improvement Categories	#P	#C
Collaboration & Communication	21	65
More communication or collaboration; Have a clear or open communication; Actively build or improve the relationship of developers and security experts; Soft skills will help developers		
Involvement Frequency	18	43
Involve security experts early; Have early meetings with security experts and developers; Have regular meetings with the other party; Involve security people in the agile release train; Make part of the day-to-day work; Involve security people before launching; Be proactive for security		
Education or Awareness	18	41
Educate developers in security; Create awareness for security; Teach developers currents CVEs		
Attitudes & Perspective	11	15
Security experts and developers change roles for a periode; Prefer more senior security people; Hire security people with developer background; Team lead should have a security bakground; Developers should have a security background		
Guidelines, Best Practices and Documentation	10	18
Provide security best practices; Define hard rules; Have a good onboarding process; Have more documentation; Have clear documentation		
Developer Support	9	12
Follow advice from security expers; Security experts should support developers more; Security experts improve information sharing		
Organisational Structures	9	26
Build a security team early; Implement a DevSecOps; Have full time security experts; Have a security person within the team; Have a security stake in the software council; Have a centralized security team		
Tooling	6	8
Implement automated security testing; CI CD will help; Leverage more tools		
Threat modeling & Risk Communication	5	8
Communicate risks properly; Identify threats; Implement a RACI matrix		
Leadership Support	2	5
Get support from the management; Make security a priority in management		

#P: Number of participants contributing statements to the respective category.
#C: Total count of coded statements for each category.

C. Improving Developer-Security Expert Collaboration Bonds

Aligned with Scrum's philosophy of "inspect and adapt" [22], after discussing with our participants about their relationship and collaboration, reflecting on both the positive aspects and the challenges they face, we asked them about the best ways to improve it (see Table III).

Collaboration & Communication.

12 participants (e.g. D_6, D_8, S_2, S_4. S_{13}) highlighted the need for increased communication to enhance collaboration. Participant D_8 expressed a desire for security experts to be physically present in the office, stating, "*So, we both can sit on the same floor. So, we can just communicate. So, this will dramatically reduce the gap between security guy and a developer*" (D_8). S_2 suggested that increased communication would enable him "*to delve deeper into technical perspectives and aspects.*" Six participants (D_4, D_5, D_6, D_9, S_3), emphasized the importance of clear and open communication during meetings and in conveying requirements or concerns. D4 shared a positive experience of open communication, highlighting a meeting where everyone freely expressed their thoughts and constructive criticism without any negativity, believing that open communication can resolve any issue. S_{11} emphasized the importance of communicating sensitively, stating: "*being able to articulate, [...], the business case, the objective of why it is important to collaborate. [...]and being able to, you know, kind of put that message across.*" (S_{11}) Twelve participants suggested enhancing the relationship between security experts and developers to improve collaboration. S_1 emphasized the need "*to put energy into that relationship actively.*" S_3 mentioned "*company-wide initiatives within the team, hackathons, capture-the-flag activities*" are effective in strengthening the bond. D_4, D_9 and S_2 suggested "*having a coffee*" together, "*going for lunch*" or conduct "*team-building activities*" to foster better relationships between the groups. By contrast, S7 feared that security experts might hesitate to criticize developers' code due to personal sympathies and project pressures.

Involvement Frequency. Eighteen participants discussed their collaboration frequency. Many mentioned the importance of involving security experts early in development or "*invite them to the agile release train*" (D_2). Five other participants (D_3, D_{12}, S_3, S_4, S_9) also emphasized a holistic integration of security experts.

Education or Awareness. Eight developers (e.g., D_2, D_5, D_8, D_9 D_{14}) and ten security experts (e.g., S_1, S_3, S_5, S_6, S_{13}) highlighted the importance of security awareness and education in our interviews. Thirteen participants (e.g., D_1, D_9, S_5, S_8, S_{11}) noted that companies invest in developer education through various methods like yearly training, security champion programs, and workshops. These programs offer learning modules (D_1), video series (e.g., D_5, D_7, D_9), and yearly security sessions (e.g., D_7, D_{11}, S_{11}). One developer, D_6, preferred that their training is "*face-to-face because it's more interactive*". Security experts usually provide the materials for these training sessions (e.g., D_5,'D_7, D_{11}), with a developer noting that security experts "*are taking care about our brains*" (D_9). Some companies also collaborate with third-party education providers (D_5), enhancing knowledge exchange and benefiting both parties, as expressed by S_1.

Attitudes & Perspective. Participants suggested enhancing collaboration by ensuring security experts have a software development background. However, D_{11} proposed hiring developers with security expertise, and S_7 stressed the team lead's grasp of security. D_7 mentioned an internship program for team members to serve in various roles, while D_9 recommended seat swaps to foster better understanding among security and other roles. D_8 supported this approach, noting it helps developers think like security experts, strengthening their connection. Additionally, S_1 and S_6 emphasized extending security education to product owners and business managers.

Guidelines, Best Practices, and Documentation. D_2, D_4, and S_3 stressed providing best practices or enforcing a minimum set of security rules for developers. Additionally, six participants (e.g., D_5, D_{11}, S_5, S_{13}, S_7) emphasized the significance of documentation: "*Documentation is really crucial part of everything. [...] you also have to make sure that the people in the company are following the documentation.*" (S_{13}). Other participants underscored the importance of clear, accurate documentation and consistent communication of guidelines.

TABLE IV: Overview of challenges found in past research

Challenge		Past Research
IV-B1	Interaction Difficulties	[18], [52]
IV-B2	Balancing Business Goals and Security Needs	[5], [17], [18], [23], [26], [34], [35], [38], [51]
IV-B3	Security Perception	[5], [6], [18]
IV-B4	Security Resource Constraints	[5], [6], [12], [17], [18], [51],
IV-B5	Lack of Security Awareness and Knowledge	[5], [35], [51]

Developer Support. S$_6$ claimed, that "*they can follow [security experts'] guidelines. So that will make [their] life easier.*". Also, D$_4$, D$_6$, and S$_3$ emphasized developers should heed security experts' advice. D$_2$ suggested that security experts should go a step further and "*start working on the development as well. [. . .] which can make the developer's life easier.*" (D$_2$)

Organisational Structures. D$_1$, D$_{11}$, and S$_8$ emphasized the value of having a security person integrated into the team, similar to a security champion program. D$_{11}$ shared positive experiences with such programs. Additionally, five participants (D$_1$, D$_4$, D$_6$, D$_8$, S$_9$) proposed DevSecOps as a solution to bridge the gap between DevOps and security experts. Further, it was suggested to establish a centralized security team with full-time employees dedicated to security tasks. D$_6$ also recommended including security representation in the software council or a similar organizational unit.

Tooling. Six participants (e.g., D$_5$, S$_1$, S$_5$, S$_6$, S$_8$) were in favor of working with and embedding automated tools. This would make the work easier, take the burden off the shoulders of the developers, and make it easier for security experts.

Threat modeling & Risk Communication. Ds5, D$_{12}$, S$_3$3 suggested threat modeling to visualize risks from day one. Further, S$_3$ and S$_4$ emphasized that proper risk communication prevents the blame game.

Leadership Support. S$_1$ mentioned that support from the leadership is the most important factor, stating "*Number one [...] support from the management team*" (S$_1$) or else developers will claim they are " *busy. I'm programming here. Go away*" (S$_1$).

V. Discussion and Recommendations

A. Comparison with Related Work

Our participants identified five key collaboration challenges (see Section IV-B) that impede effective teamwork. These issues are consistent with findings from prior research, as illustrated in Table IV.

Interaction Difficulties. The combination of an unsupportive security environment and individual security perceptions leads to interaction problems between developers and security experts. Similarly, Van Der Linden et al. [52] outlined that developers faced difficulties in interaction with other stakeholders regarding security issues. Gutfleisch et al. [18] investigated a security champion program in a corporate context and found that team members in Scrum teams also encountered communication issues in the context of security. This misalignment of priorities increases tension, with developers viewing security measures as obstacles and security experts feeling undervalued or sidelined. Despite Scrum's values of *openness* and *respect*, our participants reported resistance to sharing information, cultural differences, fostering judgment, ego preservation, and general resistance to change (see Section IV-B1). Developers hesitate to accept feedback from security experts, and events like Sprint Reviews and Sprint Retrospectives are underutilized, resulting in missed opportunities to evaluate and adjust their relationship.

Balancing Business Goals and Security Needs. Balancing business goals with security needs was one of the most prominent challenges security experts and developers encountered (see Section IV-B2). In a Scrum setup, the emphasis on speed and flexibility often leads to security being deprioritized [38], [51]. Previous research indicated that security activities are more common during the requirement and implementation phases [51]. In contrast, our study revealed an *increased involvement during the implementation and release phases.* Scrum's principle of "value-based prioritization" might influence product owners to focus on functionality and sideline security [38], [50], [51], leading to security being absent in the early phases of the project.

Security Perception. Our findings indicated that a low level of security awareness and knowledge of individual developers, a shortage of security experts, as well as the negligence of management to prioritize security shape the perception of security (see Section IV-B3). Security tends to be seen as an afterthought or an add-on to focus more on functionality rather than integrating it into the core development process as also observed in [5], [6], [18], [38], [51]. The limited resources of security expertise might influence developers to view collaboration with security experts as a blocker that delays progress, especially since security experts, being fewer in number, take longer to respond or can not attend all meetings (see Section IV-B4). This perception is reinforced as security experts are more involved in the later stages of development than from the beginning. Their involvement in these later stages tends to be on a need-basis, with the most dominant communication pattern being *artifacts communication*, resulting in minimal direct interaction (see Section IV-A2).

Security Resource Constraints. All these factors further contribute to the challenge of security resource constraints, making it hard for developers to quickly access the necessary support, leading to delays and inefficiencies. Other studies have also shown that often software and security teams are restricted in budget and time for security (e.g., [5], [17], [51]).

Lack of Security Awareness & Knowledge. Our study further shows that, despite Scrum's promotion of cross-functional teams, gaps in security skills persist, resulting in a heavy reliance on scarce and overburdened external security experts. Scrum does not designate a specific role for security, leaving no one explicitly responsible for ensuring that security practices are consistently followed [51]. The lack of security awareness and knowledge confirms the findings in [47], citing that developers depend heavily on security experts, overloading them with work and creating bottlenecks. We believe this might stem from the absence of the Scrum value of *courage*, which is essential for open communication, respectful discussions, and confronting challenges head-on. Our participants expressed fear (Section IV-B1) and a lack of knowledge of security topics (Section IV-B5) and enthusiasm for collaborating with security experts (Section IV-B3).

B. Recommendations for Practitioners

Enhancing Scrum Master Engagement in Security. Scrum Masters play a crucial role in facilitating communication between developers and security experts, yet our findings suggest they are often underutilized in this capacity. By promoting the Scrum values of openness, respect, and courage, Scrum Masters can foster a supportive environment that enables open discussions of security issues. To integrate security effectively, Scrum Masters should involve security experts in Sprint Planning, where potential risks can be identified and discussed from the start. Additionally, when specific security issues arise, inviting security experts to Sprint Retrospectives ensures these issues are addressed and resolved promptly, preventing potential tension between developers and security experts.

Product Owners: Prioritizing Security in Agile Development. As evidenced by our identified challenge of balancing business goals with security needs, previous work showed that product owners (1) often lack awareness and knowledge of security importance [15], [48], (2) often have to prioritize functionality over security to meet business goals [50], [51]—resulting in quick fixes or ignoring security issues to meet deadlines [7], [38], [51]. Thus, product owners might integrate security into the Product Backlog, inviting security experts to Sprint Reviews to raise stakeholder awareness and improve product security.

Fostering Collaboration: Security Champions, DevSecOps, and Co-Creation. Organizations can help strengthen the relationship between security experts and developers, addressing the security expert shortage [5], [6], [12], easing the workload for security experts, and improving communication by following different approaches: (1) Foster a *security champions program* to enhance security communication [9], [16], [47], (2) Adopt *DevSecOps* to promote collaboration among development, operations, and security teams [3], [30], and (3) Adopt *Co-creation* (e.g., pair-programing) to work side by side, compensating for each other's weaknesses [43], [53].

Developing Personal Relationships through Secure Development Practices. Organizations should develop personal relationships between security experts and developers through methods like Secure Software Development Life Cycle (SSDLC) principles [33] or secure software development frameworks (e.g., NIST [40], OWASP SAMM [36], and BSIMM [29]). They could also draw inspiration from BSIMM's activity ”T2.5: Enhance satellite through training and events” to foster camaraderie and collaboration between security experts and developers [29].

C. Takeaways and Future Research

Open Scrum Challenges. The existing literature on Scrum did not discuss the direct challenges of its practices; instead, it focused on their impact on areas like security (see Table IV). Our research addresses this gap by directly examining the challenges Scrum practices pose in the security context. Our findings highlight that Scrum needs some changes to adhere to security implementation. While Scrum encourages collaboration, it seems to negatively impact collaboration between security experts and developers, specifically the principle of ”value-based prioritization.” Further research is needed on the three Scrum values (openness, respect, and courage) and how to overcome challenges created by Scrum's emphasis on speed, flexibility, and value-based prioritization.

Correlations. Our study revealed correlations within both developers and security experts regarding challenges in collaboration, particularly in communication and balancing security with business goals. Developers struggle with tension between deadlines and security, while security experts face resource constraints. Despite these distinct experiences, both groups acknowledge that these issues influence their collaborative efforts. Correlations between cohorts (developers and security experts) revealed a shared understanding of issues such as gaps in security knowledge and misaligned priorities, which may contribute to collaborative difficulties. No significant correlations were found with demographic data. However, individual observations raised potential areas for further investigation. For instance, one participant noted varied collaboration experiences based on developers' industry experience, while another highlighted challenges unique to women security experts.

Trade-offs. Following Robillard et al. approach [39], our study identified several trade-offs impacting its design and findings. One significant trade-off was the focus on specific challenges in Scrum and security integration, which limited the exploration of other agile methodologies or varied industry practices. Additionally, while our participant pool provided deep insights into personal experiences, a more diverse group might have offered broader perspectives. Future work could investigate these trade-offs further by exploring a wider range of agile practices, incorporating more diverse participant groups, and considering emerging security trends.

VI. Conclusion

In this paper, we conducted semi-structured interviews with 27 developers and security experts to explore their collaboration in a Scrum setting. We identified communication patterns and challenges affecting their relationship and collaboration, highlighting the need for proactive measures. Some Scrum values were found to be lacking, suggesting that embracing these values could improve the collaborative environment. However, Scrum alone is insufficient; additional approaches like Security Champions, DevSecOps and SSDLC are necessary. Future research should explore demographic impacts on collaboration open Scrum challenges and a wider range of agile practices.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. We are additionally grateful to Tobias Bruns for assistance with the literature review for this study. This work was co-funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and the joint project: digital fitness for citizens - realistic risk perception, secure routines - "DigiFit" - grant number 16KIS1646K.

REFERENCES

[1] Hege Aalvik, Anh Nguyen-Duc, Daniela Soares Cruzes, and Monica Iovan. Establishing a security champion in agile software teams: A systematic literature review. In *Future of Information and Communication Conference*, pages 796–810, Cham, 2023. Springer, Springer Nature Switzerland.

[2] Yasemin Acar, Michael Backes, Sascha Fahl, Doowon Kim, Michelle L Mazurek, and Christian Stransky. You Get Where You're Looking For: The Impact of Information Sources on Code Security. In *Proc. 37th IEEE Symposium on Security and Privacy (SP'16)*, pages 289–305, Los Alamitos, CA, USA, 2016. IEEE.

[3] Muhammad Azeem Akbar, Kari Smolander, Sajjad Mahmood, and Ahmed Alsanad. Toward successful devsecops in software development organizations: A decision-making framework. *Information and Software Technology*, 147:106894, 2022.

[4] Mohammed Alnatheer, T. Chan, and Karen Nelson. Understanding and measuring information security culture. *Proceedings - Pacific Asia Conference on Information Systems, PACIS 2012*, 0(0), 01 2012.

[5] Hala Assal and Sonia Chiasson. Security in the software development lifecycle. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 281–296, Baltimore, MD, August 2018. USENIX Association.

[6] Hala Assal and Sonia Chiasson. 'think secure from the beginning': A survey with software developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Zulkarnain Azham, Imran Ghani, and Norafida Ithnin. Security backlog in scrum security practices. In *2011 Malaysian Conference in Software Engineering*, pages 414–417, 2011.

[8] Ingolf Becker, Simon Parkin, and M Angela Sasse. Finding security champions in blends of organisational culture. *Proc. USEC*, 11:124, 2017.

[9] Odette Beris, Adam Beautement, and M. Angela Sasse. Employee rule breakers, excuse makers and security champions: Mapping the risk perceptions and emotions that drive security behaviors. In *Proceedings of the 2015 New Security Paradigms Workshop*, NSPW '15, page 73–84, New York, NY, USA, 2015. Association for Computing Machinery.

[10] Melanie Birks, Ysanne Chapman, and Karen Francis. Memoing in qualitative research: Probing data and processes. *Journal of research in nursing*, 13(1):68–75, 2008.

[11] Virginia Braun, Victoria Clarke, Nikki Hayfield, and Gareth Terry. *Thematic Analysis*, pages 843–860. Springer Singapore, Singapore, 2019.

[12] Sascha Fahl Charles Weir, Ben Hermann. From needs to actions to secure apps? the effect of requirements and developer practices on app security. In *Proceedings of the 29th USENIX Conference on security symposium*, page 289–305, USA, 2020. USENIX.

[13] Adéle da Veiga, Liudmila V. Astakhova, Adéle Botha, and Marlien Herselman. Defining organisational information security culture—perspectives from academia and industry. *Computers & Security*, 92:101713, 2020.

[14] digital.ai. Annual State of Agile Report. https://info.digital.ai/rs/981-LQX-968/images/RE-SA-17th-Annual-State-Of-Agile-Report.pdf?version=0, 2023.

[15] S. Dziwok, S. Merschjohann, and T. Koch. A software security study among german developers, product owners, and managers. In Tareq Ahram and Waldemar Karwowski, editors, *Human Factors in Cybersecurity*, volume 53 of *AHFE Open Access*, USA, 2022. AHFE International.

[16] Trevor Gabriel and Steven Furnell. Selecting security champions. *Computer Fraud & Security*, 2011(8):8–12, 2011.

[17] Marco Gutfleisch, Jan H. Klemmer, Niklas Busch, Yasemin Acar, M. Angela Sasse, and Sascha Fahl. How does usable security (not) end up in software products? results from a qualitative interview study. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 893–910, San Francisco, CA, USA, 2022. IEEE.

[18] Marco Gutfleisch, Markus Schöps, Stefan Albert Horstmann, Daniel Wichmann, and M. Angela Sasse. Security champions without support: Results from a case study with owasp samm in a large-scale e-commerce enterprise. In *Proceedings of the 2023 European Symposium on Usable Security*, EuroUSEC '23, page 260–276, New York, NY, USA, 2023. Association for Computing Machinery.

[19] Joseph Hallett, Nikhil Patnaik, Ben Shreeve, and Awais Rashid. "Do this! Do that!, And Nothing will happen": Do specifications lead to securely stored passwords? In *43rd International Conference on Software Engineering*, pages 486–498, United States, January 2021. Institute of Electrical and Electronics Engineers (IEEE). Edition: 43.

[20] Microsoft Threat Intelligence. Microsoft Threat Intelligence auf X: Microsoft has identified a Russian-based nation-state threat actor tracked as Forest Blizzard (STRONTIUM, APT28, FANCYBEAR) actively exploiting CVE-2023-23397 to provide secret, unauthorized access to email accounts within Exchange servers:. https://twitter.com/MsftSecIntel/status/1731626192300634585, December 2023.

[21] Martin Karlsson, Fredrik Karlsson, Joachim Åström, and Thomas Denk. The effect of perceived organizational culture on employees' information security compliance. *Information & Computer Security*, ahead-of-print, 12 2021.

[22] Ken Schwaber and Jeff Sutherland. Scrum guide. https://scrumguides.org/scrum-guide.html, 2020.

[23] Laura Kocksch, Matthias Korn, Andreas Poller, and Susann Wagenknecht. Caring for IT Security: Accountabilities, Moralities, and Oscillations in IT Security Practices. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW), November 2018. Place: New York, NY, USA Publisher: ACM.

[24] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. A survey of devops concepts and challenges. *ACM Computing Surveys*, 52(6):1–35, November 2019.

[25] J. Lim, S. Chang, S. Maynard, and A. Ahmad. Exploring the relationship between organizational culture and information security culture. *Proceedings of the 7th Australian Information Security Management Conference*, 107(3):88–97, December 2009.

[26] Tamara Lopez, Thein Tun, Arosha Bandara, Mark Levine, Bashar Nuseibeh, and Helen Sharp. Hopefully we are mostly secure: Views on secure code in professional practice. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, Montreal, QC, Canada, 05 2019. IEEE.

[27] Stahnke Mann, Brown and Kersten. State of devops report 2018, 2018.

[28] Michelle Mazurek. We are the experts, and we are the problem: The security advice fiasco. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 7, Los Angeles, CA, USA,, November 2022. ACM.

[29] Gary McGraw and Brian Chess. The building security in maturity model ({BSIMM}), 2009.

[30] Håvard Myrbakken and Ricardo Colomo-Palacios. Devsecops: A multi-vocal literature review. In *International Conference on Software Process Improvement and Capability Determination*, pages 17–29, Palma de Mallorca, Spain, 09 2017. Springer International Publishing.

[31] Alena Naiakshina, A. Danilova, Christian Tiefenau, Marco Herzog, Sergej Dechand, and Matthew Smith. Why Do Developers Get Password Storage Wrong?: A Qualitative Usability Study, 2017.

[32] NIST. National vulnerability database. https://nvd.nist.gov/vuln/search/results?form_$type{=}Basic\&results_type{=}overview\&search_type{=}all\&isCpeNameSearch{=}false$, 2024.

[33] OWASP. Secure software development life cycle. https://www.owasp.org/images/9/9d/OWASPLATAMTour-Patagonia-2016-rvfigueroa.pdf, 2016.

[34] Hernan Palombo, Armin Ziaie Tabari, Daniel Lende, Jay Ligatti, and Xinming Ou. An ethnographic understanding of software (In)Security and a Co-Creation model to improve secure software development. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 205–220, USA, August 2020. USENIX Association.

[35] Andreas Poller, Laura Kocksch, Sven Türpe, Felix Anand Epp, and Katharina Kinder-Kurlanda. Can Security Become a Routine? A Study of Organizational Change in an Agile Software Development Group. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 2489–2503, New York, NY, USA, 2017. ACM. Portland, Oregon, USA.

[36] Open Web Application Security Project. Open web application security project software assurance maturity model. https://owaspsamm.org/, 2022.

[37] Roshan N. Rajapakse, Mansooreh Zahedi, M. Ali Babar, and Haifeng Shen. Challenges and solutions when adopting devsecops: A systematic review. *Information and Software Technology*, 141:106700, January 2022.

[38] Kalle Rindell, Jukka Ruohonen, Johannes Holvitie, Sami Hyrynsalmi, and Ville Leppänen. Security in agile software development: A practitioner survey. *Information and Software Technology*, 131:106488, 2021.

[39] Martin P. Robillard, Deeksha M. Arya, Neil A. Ernst, Jin L. C. Guo, Maxime Lamothe, Mathieu Nassif, Nicole Novielli, Alexander Serebrenik, Igor Steinmacher, and Klaas-Jan Stol. Communicating study design trade-offs in software engineering. *ACM Trans. Softw. Eng. Methodol.*, 33(5), jun 2024.

[40] Karen Scarfone, Murugiah Souppaya, and Donna Dodson. Secure software development framework (ssdf) version 1.1: Recommendations for mitigating the risk of software vulnerabilities. https://tsapps.nist.gov/publication/get_*pdf.cfm?pub_id*=934124, 2022-02-03 05:02:00 2022.

[41] Ken Schwaber and Jeff Sutherland. The scrum guide. *Scrum Alliance*, 21(1):1–38, 2011.

[42] SCRUMstudy. Scrum body of knowledge (sbok® guide), 2016.

[43] Hamed Yaghoubi Shahir, Shervin Daneshpajouh, and Raman Ramsin. Improvement strategies for agile processes: A swot analysis approach. In *2008 Sixth International Conference on Software Engineering Research, Management and Applications*, pages 221–228, 2008.

[44] Ben Shreeve, Joseph Hallett, Matthew Edwards, Marvin Ramokapane, Richard Atkins, and Awais Rashid. The best laid plans or lack thereof: Security decision-making of different stakeholder groups. *IEEE Transactions on Software Engineering*, 48, Issue 5:1515 – 1528, September 2020. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[45] Benjamin Shreeve, Joseph Hallett, Matthew Edwards, Pauline Anthonysamy, Sylvain Frey, and Awais Rashid. "So If Mr Blue Head Here Clicks the Link..." Risk Thinking in Cyber Security Decision Making. *ACM Trans. Priv. Secur.*, 24(1):1–29, November 2020. Place: New York, NY, USA Publisher: ACM.

[46] Cisco Talos. Active exploitation of Cisco IOS XE Software Web Management User Interface vulnerabilities. https://blog.talosintelligence.com/active-exploitation-of-cisco-ios-xe-software/, October 2023.

[47] Tyler W. Thomas, Madiha Tabassum, Bill Chu, and Heather Lipford. Security during application development: an application security expert perspective. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[48] Inger Anne Tøndel, Daniela Soares Cruzes, Martin Gilje Jaatun, and Kalle Rindell. The security intention meeting series as a way to increase visibility of software security decisions in agile development projects. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19, New York, NY, USA, 2019. Association for Computing Machinery.

[49] Anwesh Tuladhar, Daniel Lende, Jay Ligatti, and Xinming Ou. An analysis of the role of situated learning in starting a security culture in a software company. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 617–632, USA, August 2021. USENIX Association.

[50] Sven Türpe and Andreas Poller. Managing security work in scrum: Tensions and challenges. In *SecSE@ESORICS*, 2017.

[51] Inger Anne Tøndel, Daniela Soares Cruzes, Martin Gilje Jaatun, and Guttorm Sindre. Influencing the security prioritisation of an agile software development project. *Computers & Security*, 118:102744, 2022.

[52] Dirk van der Linden, Pauline Anthonysamy, Bashar Nuseibeh, Thein Than Tun, Marian Petre, Mark Levine, John Towse, and Awais Rashid. Schrödinger's Security: Opening the Box on App Developers' Security Rationale. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, pages 149–160, New York, NY, USA, 2020. ACM. Seoul, South Korea.

[53] Laurie Williams and Robert Kessler. *Pair Programming Illuminated*. Addison-Wesley Longman Publishing Co., Inc., USA, 2002.

[54] Zoom Video Communications. Zoom. https://zoom.us, 2011. Zoom is a video conferencing and communication software widely used for online meetings, webinars, and virtual collaborations.