How to Integrate Security Compliance Requirements with Agile Software Engineering at Scale?

Fabiola Moyón[1(✉)], Daniel Méndez[2,3], Kristian Beckers[4], and Sebastian Klepper[5]

[1] Technical University of Munich and Siemens, Munich, Germany
fabiola.moyon@tum.com
[2] Blekinge Institute of Technology, Karlskrona, Sweden
daniel.mendez@bth.se
[3] fortiss GmbH, Munich, Germany
[4] Social Engineering Academy, Frankfurt am Main, Germany
kristian.beckers@social-engineering.academy
[5] Technical University of Munich, Munich, Germany
sebastian.klepper@tum.de

Abstract. Integrating security into agile software development is an open issue for research and practice. Especially in strongly regulated industries, complexity increases not only when scaling agile practices but also when aiming for compliance with security standards. To achieve security compliance in a large-scale agile context, we developed S^{2}C-SAFe: An extension of the Scaled Agile Framework that is compliant to the security standard IEC 62443-4-1 for secure product development.

In this paper, we present the framework and its evaluation by agile and security experts within Siemens' large-scale project ecosystem. We discuss benefits and limitations as well as challenges from a practitioners' perspective. Our results indicate that S^{2}C-SAFe contributes to successfully integrating security compliance with lean and agile development in regulated environments. We also hope to raise awareness for the importance and challenges of integrating security in the scope of Continuous Software Engineering.

Keywords: Secure software engineering · Scaled Agile Framework · Security standards

1 Introduction

Security compliance is a major concern for several industries [8,18]. Typically, security practitioners (and regulators) hold a holistic view on security affecting people, processes, and technology [8,19,20]. The perspective of practitioners, however, is rather dispersed and security is commonly treated as just another non-functional requirement [17]. Security engineering activities are further too

M. Morisio et al. (Eds.): PROFES 2020, LNCS 12562, pp. 69–87, 2020.
https://doi.org/10.1007/978-3-030-64148-1_5

often applied in an ad-hoc manner to a limited set of security problems, e.g., vulnerability testing or static code analysis [8]. Security concerns are often mixed with software functionality and limited to specific implementations like authentication or encryption [34].

Integrating security into lean and agile processes further intensifies these issues and constitutes a well-known research problem [1,17,35]. This is especially true for large software development projects. One challenge here is to fulfil requirements rigorously to comply with regulations while not limiting the speed and flexibility agile development methodologies promise. However security standards often require a series of processes to define, analyse, and mitigate security vulnerabilities [23] whereas lean and agile methodologies aim at avoiding rigid linear processes. While the agile manifesto states "to value individuals and interactions over processes", "collaboration over contract negotiation", and "responding to change over following a plan" [6], standards explicitly demand documented evidence of responsibilities, agreements, and established development procedures.

Our research shall provide a perspective for resolving this conflict through *Continuous Security Compliance*. In particular, we aim at implementing security standard requirements along with agile development methodologies. To this end, we analysed the issue in a large industrial setting and its currently applied norms: the Scaled Agile Framework (SAFe) as well as the IEC 62443-4-1 standard, later we propose a revised framework dubbed S^2C-SAFe . We chose the IEC 62443-4-1 standard for secure product development, released in 2018 based on previous secure product development standards such as BSIMM [25], ISO27034 [22], or Security by Design with CMMI [33]. Our framework shall maintain SAFe's perspective on development procedures and principles while capturing the essential requirements of security standards. In this paper, we contribute:

1. The proposal of our S^2C-SAFe framework, a security-standard compliant variant of the Scaled Agile Framework.
2. An evaluation of the S^2C-SAFe framework in large-scale software development environments. Given that the introduction of SAFe may take up to 8 years in the chosen organisational context, we conduct our evaluation in a preliminary manner focusing particularly on expert interviews.

We conclude our evaluation with the practitioners' perception of the challenges to achieve security compliance in a continuous manner. By sharing these insights, we particularly hope to raise awareness for the importance, but also challenges of integrating security in large-scale software development organisations following lean and agile principles.

2 Fundamentals and Related Work

Continuous Software Engineering (CSE) utilises lean and agile principles for a rapid and continuous "flow" of activities across business, development, and operations [16].

In their "Continuous *" model of CSE, Fitzgerald et al. [17] describe Continuous Security and Continuous Compliance as related but separate concerns and activities. *Continuous Compliance* (CC) seeks to satisfy regulatory compliance standards on a continuous basis rather than a "big-bang" approach to ensure compliance at release time [18,26]. *Continuous Security* (CS) elevates security from non-functional requirement to key concern by efficiently identifying and addressing security issues throughout all processes [16].

Related work discusses the suitability of agile methods for regulated environments [18] or the extensibility of their use [10]. With regard to security, authors focus on solving security aspects in agile environments, without considering regulations as focus [4,5,9,31]; or deriving security activities from a regulations perspective but lacking attention to lean and agile environments as well as corporate operating procedures, e.g., product life cycle [7,10]. Practical concerns of CS are: adapting the development process to security, better eliciting and tracking security requirements, and incorporating assurance into iterations [5].

Separating CS and CC is illustrated by Fitzgerald et al. [18], concluding that agile methods are suitable for security-critical environments, but not yet adopted in regulated environments.

We aim for *Continuous Security Compliance* (CSC): combining CC and CS through the holistic view of standardisation that spans across people, processes, and technology [20]. Regulatory requirements are utilised to derive security activities and therefore integrating security into a process while also making it standards-compliant [28]. Further work concentrates on security governance best practices [12]. This is complementary to prior work focused on the technology side, integrating security engineering into agile processes [1,3,8,11,13], or on the process side, integrating desirable but not standards-compliant security activities [1,2,32].

S^2C-SAFe is the result of applying this holistic principle to both a security-critical and a regulated domain: industrial and automation control systems. The result is an in-depth analysis of a security standard (IEC 62443-4-1) followed by the integration with lean and agile development practices represented by the Scaled Agile Framework (SAFe).

IEC 62443 constitutes a series of standards for network and system security published by the International Electrotechnical Commission (IEC). The standard focuses on requirements for component providers for industrial automation and control systems (IACS), part 4-1 describes process requirements for secure product development [21]. We reference this part of the standard as "4-1" or "4-1 standard". *SAFe* is a widely used process framework that scales lean and agile development to large organisations with multiple levels. It furthermore defines the corresponding roles, responsibilities, activities, and artefacts [24].

For such IACS environments our contribution aims to bridge the gap between lean and agile development, practical security, and compliance [34].

3 S²C-SAFe Framework in a Nutshell

The overall aim of our work is to improve product development life-cycle by integrating requirements of IEC 62443-4-1 into SAFe, resulting in the "Security Standard Compliant Scaled Agile Framework" (S²C-SAFe). Figure 1 shows how this is achieved by using visual modelling and by merging techniques as presented in our previous work [28]. Essential elements of SAFe and 4-1, such as roles, activities, and artefacts, were captured using Business Process Model and Notation (BPMN), a visual modelling language capable of expressing all of these aspects at once. After refining these models separately with expert practitioners, the process framework model is extended with elements from the security standard model, yielding the S²C-SAFe framework. Previously we found that a visual approach allows for more focused reviews than textual representation.

S²C-SAFe describes how requirements of 4-1 can be implemented within SAFe by showing when to involve roles, execute activities, or generate artefacts. It focuses on SAFe's Continuous Delivery Pipeline (CDP), where the actual product development occurs, and makes it compliant with security requirements (SR), secure implementation (SI), and security verification and validation testing (SVV). These scopes address concerns we captured from practitioners such as frequent vulnerability testing, security requirements traceability, or coding standards review. In addition to a CDP model integrated with SR, SI, and SVV, S²C-SAFe contains detailed models for each practice. Figure 2 shows an overview of the S²C-SAFe CDP. The full framework is available in the online material associated with this paper[1].

3.1 Security Requirements (SR)

SAFe does not specify where and how to elicit security requirements even though (security) requirements elicitation constitutes a major challenge both in practice and research [14], especially when developing a product threat model and deriving requirements to counter threats [5,15]. S²C-SAFe therefore explicitly considers security requirements at program and team level and makes them part of the Backlog, equal to all other requirements in prioritisation and traceability. Security Experts facilitate analysis but are not primarily responsible. Instead, Product Management, Business Owners, and Systems Architects are in charge so they become aware of threats. Similarly, the Product Owner requires adequate training to be able to prioritise and approve security requirements.

[1] https://dx.doi.org/10.6084/m9.figshare.7149179.

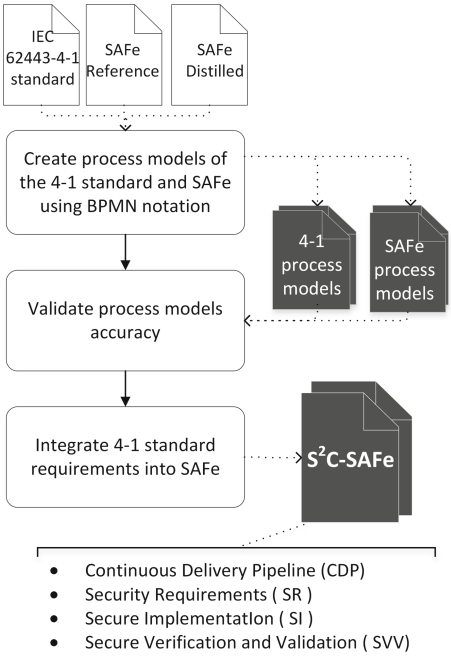

Fig. 1. Creation of S^{2}C-SAFe by generating and merging visual models of 4-1 and SAFe. Black document symbols designate our contribution. In previous work, we described the integration method [28]. The present contribution presents the S^{2}C-SAFe framework and its evaluation.

3.2 Secure Implementation (SI)

SI involves following secure coding standards to avoid vulnerabilities. S^{2}C-SAFe follows a process based on coding analysis as introduced in [2–4]. It defines coding standards early at program level during the PI Planning Event. Security Experts provide guidance so they suit domain and solution. To ensure that coding standards are followed, they are made part of the Definition of Done and agile teams as well as the product owner are trained accordingly.

3.3 Security Verification and Validation Testing (SVV)

SVV focuses on detecting and resolving vulnerabilities. One major concern is independence of testers which is enforced through independence rules during formation of agile teams. S^{2}C-SAFe also defines how further activities such as security functionality testing, vulnerability testing, or penetration testing apply to features, user stories, or both. It also defines criteria to keep resource allocation efficient and ensure continuous security testing, placing security functionality testing at team level and conducting all testing activities on program level before every System Demo. S^{2}C-SAFe contains models that shows a 4-1 compliant SAFe System Demo (see System Demo box in Fig. 2). Figure 3 is a more granular

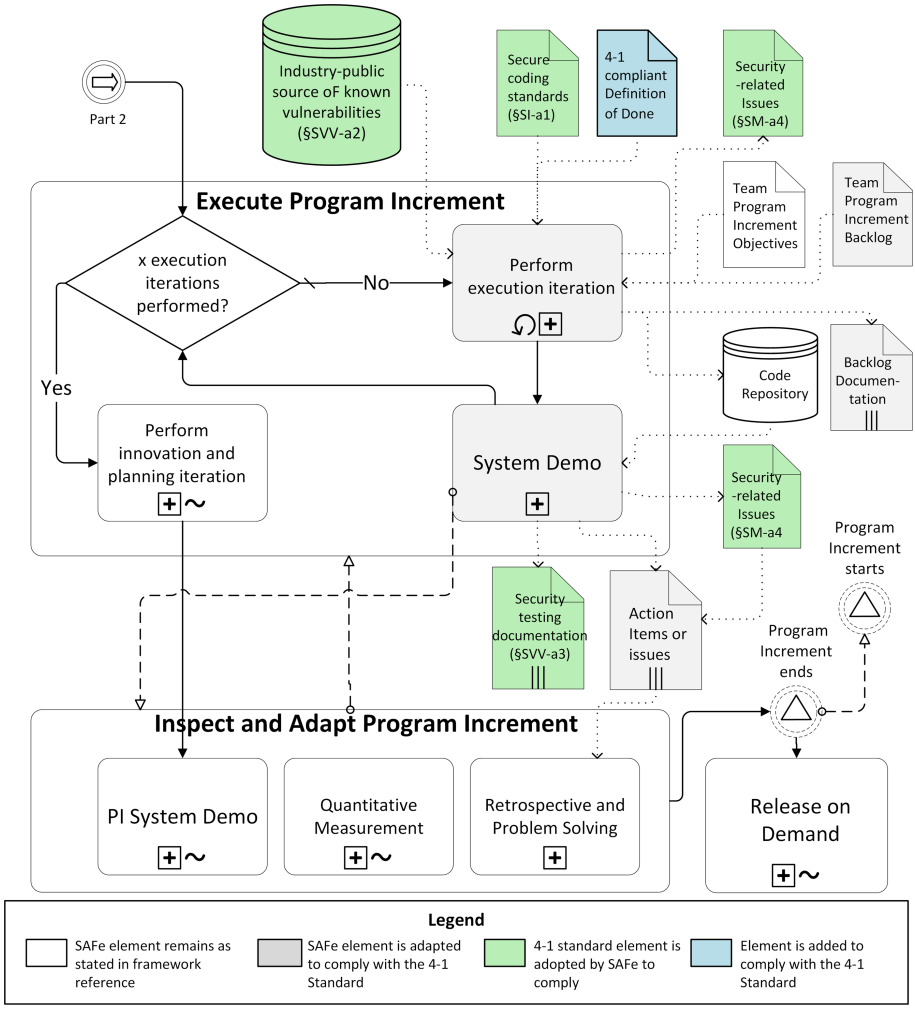

Fig. 2. Excerpt of S^{2}C-SAFe Continuous Delivery Pipeline (CDP). This overview model describes the processes involved to execute and inspect a program increment as described in SAFe plus the artefacts required by the 4-1 standard in the practices of SR, SI, and SVV.

refinement showing testing tasks and artefacts, as referred by the 4-1 practice SVV, and their mapping to SAFe roles. Further models are available in the online material.

4 Study Design

We evaluated S^{2}C-SAFe via expert interviews involving 16 practitioners working at Siemens in security compliance or (agile) software engineering. Among these experts are IEC committee members for 4-1 as well as SAFe core contributors.

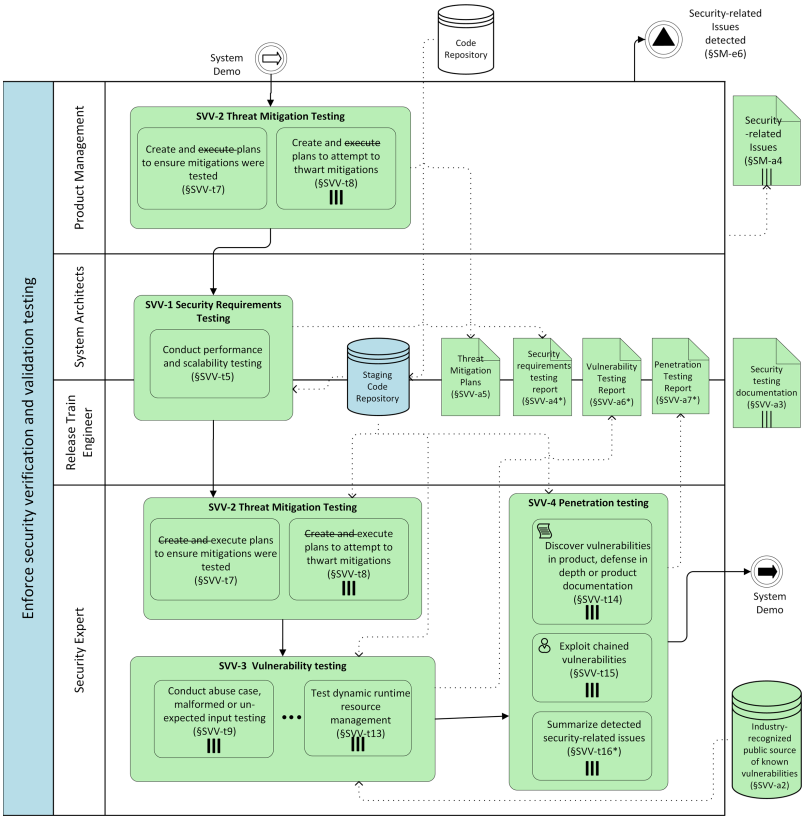

Fig. 3. S^{2}C-SAFe System Demo refinement model. Process diagram that depicts a new activity for SAFe System Demo to perform security verification and validation testing. A `Security expert` participates for certain types of testing while SAFe Program level actors are also responsible of security testing. Color coding is consistent with Fig. 2.

Our overall goal is to explore the meaningfulness of our approach to the needs in a practical context characterised by security-critical and large-scale agile development of software or software-intensive systems. Our evaluation is guided by the following two research questions:

RQ 1. From the perspective of practitioners, how applicable is S^{2}C-SAFe in this type of environment?

RQ 2. Which challenges do practitioners see when pursuing security compliance in this type of environment?

Our intention is to explore potential benefits and limitations of the here proposed framework. This shall lay the foundation for a roll-out that is minimally disruptive to the organisation and maximally intuitive for practitioners.

4.1 Subject Selection

In the industrial environment, where S^{2}C-SAFe is meant to be applied, projects are characterised by large-scale agile practices involving security experts on demand. Since industrial systems are part of critical infrastructure, such projects must comply with formal security standards, like the 4-1 standard when referring to product development. Such projects involve various agile teams with six people each. In those settings projects require direct cooperation between security experts and development teams.

We consciously selected from both groups: development teams working in these settings and security experts joining those projects on-demand, e.g., in conjunction with internal audits.

As these are all experienced professionals, we defined profiles to distinguish their level of expertise according to their key role. Table 1 shows each role's background and share of our 16 interviews. We distinguish top experienced subjects who contribute to the 4-1 standard (*Contributor IEC*) or to the SAFe framework and its dissemination within the company (*Contributor SAFe*). We further distinguish *Principle Experts*, having vast knowledge and leading teams, *Senior Experts*, having deep knowledge and guiding colleagues, and finally *Experts* who are responsible for setting up specific topics into practice.

Table 1. Mapping of interviews to subject profile and background.

Profile	Sample size	Interview numbers	Background
Contributor IEC	1	13	IACS software life cycle standardisation
Contributor SAFe	1	12	IACS agile development
Principal Expert	3	4, 5, 8	IACS security standards and processes, security life-cycle, security architecture
Senior Expert	4	1, 2, 6, 9	Cloud security, methods and tools for secure solutions, cyber security coaching, security processes improvement, IT security assessments
Expert	7	3, 7, 10, 11, 14, 15, 16	IACS agile development, quality compliance, development of access control systems, data privacy on smart cities, security design management, DevOps, security tools development, automated security testing, IT security in critical infrastructure

4.2 Survey Instrument

Since our goal is to explore practitioners' opinions about S^{2}C-SAFe , we identified semi-structured interviews as the most suitable technique [30]. Each interview lasted 1.5 to 2 h and took place in an isolated environment with one interviewee and two interviewers. One interviewer actively followed the questionnaire and the other one documented the answers and controlled attachment to interview protocol, available at our online material.

Each interview was dedicated to one S^{2}C-SAFe element according to the subject's background: SR, SI, or SVV (c.f. Sect. 3). Subjects were also introduced to the S^{2}C-SAFe CDP to have an overview of the processes involved the framework as shown in Fig. 4.

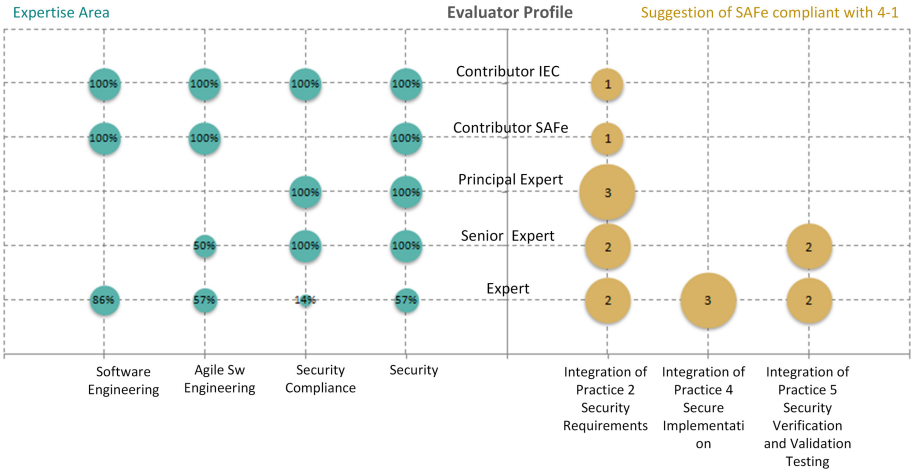

Fig. 4. S^{2}C-SAFe Suggestions distribution into profile groups. Right side: number of interviewees per suggestion. Left side: percentage of interviewees per expertise area.

Interviewers first briefed individual subjects about the interview flow and the purpose of S^{2}C-SAFe models as well as their hierarchy (overview model and individual practice models) but did not provide any instruction or training on the actual models. Then they showed a textual excerpt from 4-1 and SAFe, followed by the corresponding individual models and finally merged models from S^{2}C-SAFe . Subjects rated the perceived usefulness and practical applicability of each representation. Notes from throughout the interview were discussed before the interview's end to complete the picture.

5 Study Results

Evaluation is based on summarising the answers to closed questions and clustering comments and concerns according to commonalities. We further analysed

the emphasis of answers to differentiate acceptance vs. conviction, rejection vs. repulsion, and neutrality vs. doubt. Hence, we tabulated answers according to a 9-point Likert scale. In the following, we summarise and interpret our results according to our research questions.

5.1 Subject Knowledge

In total we selected 16 subjects with different levels of knowledge about 4-1 and SAFe. Figure 5 shows that now all of them know 4-1 but all except one are aware of other security and safety standards such as ISO/IEC 27001 or other standards of the IEC 62443 family. Similarly, not all know SAFe but all are familiar with other agile process frameworks such as Scrum.

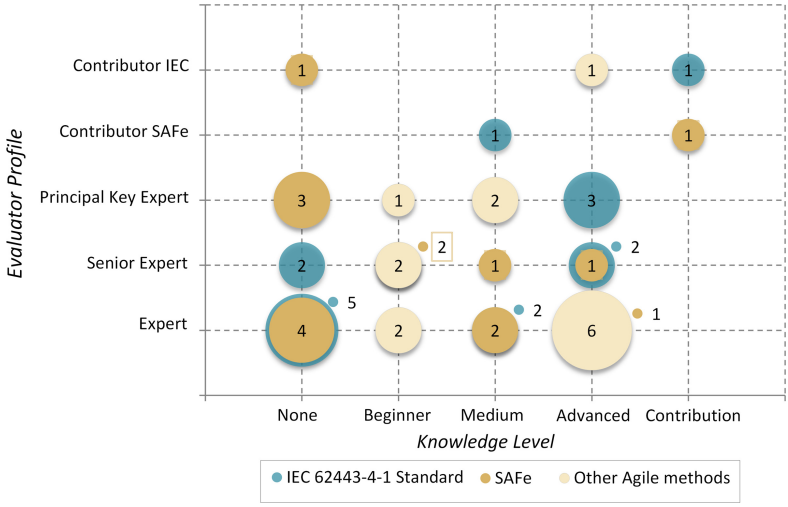

Fig. 5. Subject knowledge of IEC 62443-4-1 and SAFe or comparable process frameworks.

5.2 Applicability of S2C-SAFe (RQ 1)

We consider two aspects: applicability itself and potential implementation problems. Overall, while all interviewees strongly agree on the potential of using the integrated model as a means to foster discussions with their counterparts, they see potential problems in the integration of security aspects.

Applicability
S^2C-SAFe demonstrates that SAFe can be compliant with the 4-1 standard. All interviewees deem it usable in their environments and expressed their desire to use it for discussion with other practitioners (see Fig. 6). They particularly stated that it would provide a common language between security and development

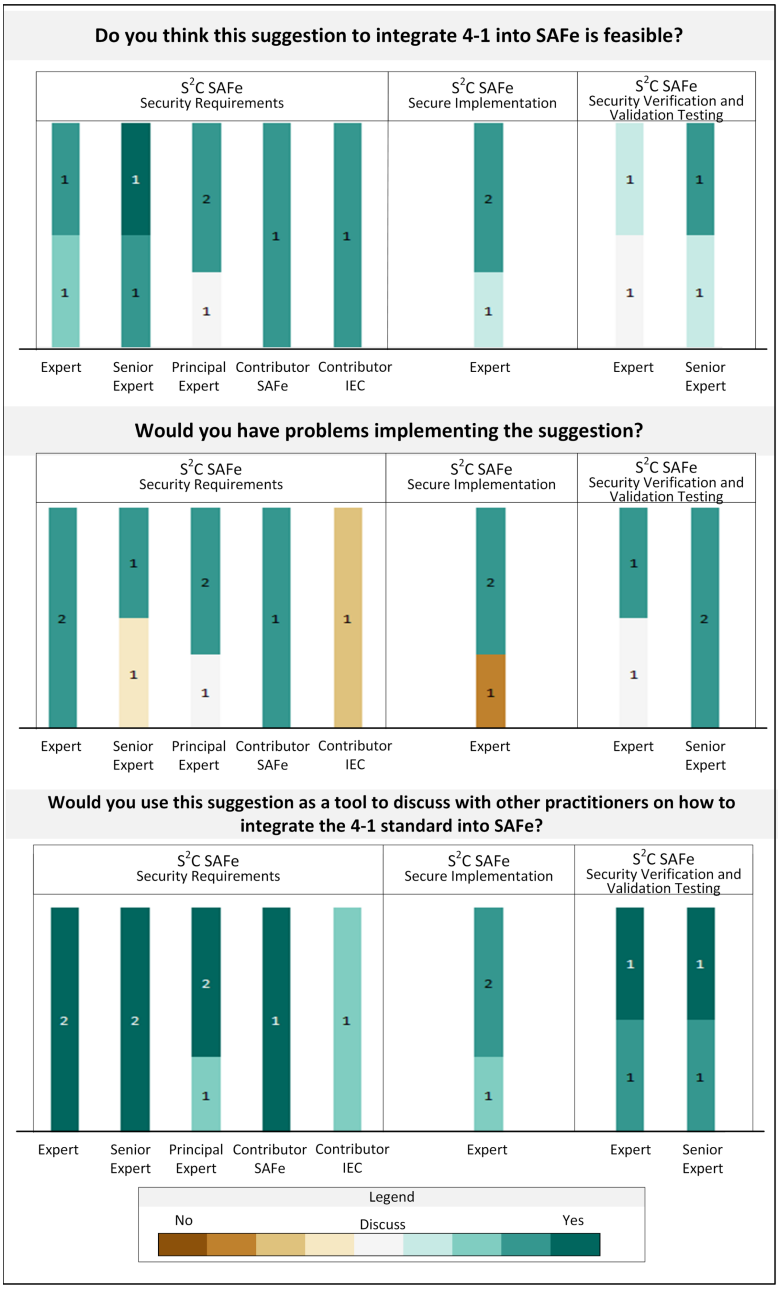

Fig. 6. Summary of opinions about S²C-SAFe applicability based on suggestions regarding 4-1 practices.

fields; some even saw it as the only such tool they are aware of (see Table 2). The following paragraphs give detailed results for each of the 4-1 practices introduced in Sect. 3.

SR: Subjects strongly agree that this suggestion is feasible. A *principal expert* (#8) did not give a positive answer, but instead argued about the complexity of having security experts within teams in general. Almost all envision problems during implementation, most relate to the lack of security practitioners, team security awareness, or split security requirements. *Contributor SAFe* thinks that proposed security activities overload PI planning while *contributor IEC* sees no problems if models are shown only to people that design processes and not to agile team. However, all subjects plan to use the suggestions as a discussion tool with their respective counterparts.

SI: Subjects strongly agree that this suggestion is feasible. One DevOps *expert* (#7) argues that educating the product owner on security is complex. Instead they propose a "security product owner" who would be capable of extending the definition of done (DoD) with security aspects. In contrast, an *expert* product owner (#14) remarks the adapted DoD as a key to apply. An *expert* security consultant (#11) is confident that problems would exist although they cannot refer to any specific one.

SVV: Although overall positive, opinions on feasibility of this suggestion are not as decided as previously. Two respondents (#11 *expert* scrum master and #1 solutions security *senior expert*) find the suggestion feasible and well integrated. Another security *senior expert* (#6) is concerned about automation support for testing non-functional security aspects and about effort for security practitioners. A security assurance *expert* (#3) argues about the role and interactions of security practitioners throughout the process. Hence, all of them envision problems related to the integration of automatic testing, workload, and expertise of security practitioners.

Additionally, as interviewers we experienced that S^{2}C-SAFe improves communication among practitioners with different profiles and backgrounds. We actively discussed interviewees' issues on security and agile development. All explanations were based on the models we provided. Subjects with the highest level of knowledge (*Contributor IEC* and *Contributor SAFe*) challenged us with management or operational questions, e.g., how to implement or even potentially bypass certain aspects. We succeeded in explaining our perspective purely by pointing out specific model aspects. Conversations were dynamic, indicating a common understanding between interviewer and interviewee. Table 2 summarises key opinions on S^{2}C-SAFe while Table 3 lists noteworthy remarks.

Potential Implementation Problems

Our interviewees raised concerns regarding implementation of S^{2}C-SAFe in their project settings. They are particularly interesting to us as they help steering future adaptations and because some concerns are rather general challenges on

Table 2. Summary of key opinions on S^2C-SAFe.

Opinion	Interviews
Facilitates common language to discuss between security experts and agile team	2, 5, 14, 16
Solution is a comprehensible, clear guide	4, 5, 7, 8
Increment effort and workload	5, 6, 9, 12
Concern about roles expertise to accomplish tasks: Product Owner, Product Management	3, 4, 6, 10
Need to increase security awareness	1, 3, 7, 8
Concerns on expertise and profile of security experts	1, 4, 10
To have security practitioners within agile teams is challenging	8, 10
Need to have a deep understanding of own process to implement suggestion	1, 16
It is the only tool available	7, 11
Concerns about fit activities into short cycles	8, 12
Find color-coding is useful	7, 13

Table 3. Interviewees' statements on S^2C-SAFe.

Quote	Interviews	Profile	Background
Big advantage, we could speak same language as SAFe experts. This would dramatically reduce problems to adapt SAFe. Yes, I would love to use it as a discussion tool	2	Senior Expert	Security compliance
It makes sense what you did. If it is not possible SAFe is broke	4	Principal Expert	Security research
Sure is feasible, how to measure success I wonder	5	Principal Expert	Head security group
It is a very nice way to reduce complexity to discuss	7	Expert	DevOps
Visibility of security into agile development environment. Transparency of what is being achieved	9	Senior Expert	Security assessments
Sure, there is nothing else. I don't think there is anything	11	Expert	Security consultant/Scrum practitioner
We need to involve a pilot implementation	12	Contributor SAFe	Head development group
I will use it as a basis to communication	14	Expert	Product Owner
I like it. It makes dedicated to think about security	15	Expert	Systems Architect

the integration of security, let alone continuous security engineering. These concerns can be summarised as follows:

Models Should Guide Instead of Comprehending Compulsory Processes
One *senior expert* argued that if a model is too strict, people will not adapt it and bypass compliance efforts (#1). The suggestions seem difficult to implement in iterations or in specific program increments. This seems particularly true for security testing (vulnerability/penetration) prior to or during a *System Demo*. This highlights the need for an incremental prototypical implementation of individual suggestions to shed light on potential adaptation barriers which might differ in dependency to the practices and the roles.

Achievement of Security Expertise and Awareness
During the design phase, we emphasised that S^2C-SAFe cannot compensate for a Product Owner with knowledge of 4-1. Our interviewees confirm that this holds not only for general SAFe roles but also for security practitioners in general. Both security and agile development experts agree that security expertise for each part of the solution requires specialisation. Such specificity on profiles would aggravate the deficit of security professionals. Exemplary statements are "During verification of compliance, people tend to deviate from the standard" (#7) or "Lack of experience on security compliance leads to failed projects" (#3, security *expert*).

Difference Between Agile and Express Development Delivery
Security is generally perceived to be something that slows down agile development processes. Some exploratory questions revealed that agile time constraints are not followed in our settings, e.g., daily meetings last more that 15 min. Our concept of agile therefore seems to relate more to iterative and incremental development than to express delivery and integrating security-related activities will surely expand this gap further. While we understand the need for a trade-off between effort and cost for adapting security (or any other quality facet) this aspect seems particularly hard to achieve and constitutes an open issue.

5.3 Continuous Security Compliance Challenges (RQ 2)

The interviewees were asked to mention priorities among the security activities described in the 4-1 development life cycle. *Security requirements* (SR) seems to be the most challenging practice for our interviewees. Other priorities differ per profile, as shown in the examples for *security management* and *security verification and validation testing*.

The top priority issue is raising awareness for security to achieve continuous security compliance. Second place is taken by an adequate prioritisation of security aspects and common perspectives among management and teams. Challenges for security integration into continuous software engineering seem similar to those with linear development models. Subsequently we summarise our key findings on the challenges raised.

Security Requirements Elicitation: Challenges go beyond elicitation, from prioritisation over allocating them to increments and tracking adequate testing. Respondents extend the concern to overall 4-1 activities into cycles, e.g., threat analysis, testing, or issue management. Some related quotes include "What does the standard says about iterations and when the required process should occur again?"(#15, software architect) or "Problem is to identify what is the most important and which things can be done in parallel" (#12).

Security More Than a Non-Functional Requirement: 4-1 contains an overview of security as described by compliance. Our interviewees state that security is normally addressed via functional requirements while other aspects, such as management-related ones, are too often left behind.

Software Architecture Impact: Software architectures are built incrementally in continuous development. One interviewee argued in particular: "How to have security design or requirements of something we don't know yet, something we create on the go" (#12, *Contributor SAFe*). We argue that security analysis can be done while thinking about the goal and later iteratively extend it to the solution-specific components. However, this needs a certain continuity just like other non-functional properties, which project participants seem to see as difficult to achieve.

Improvement Demand for Security Expertise and Awareness: In development teams the lack of expertise for security seems to be a common theme [5]. Particularly, our group of interviewees seems to have a sound level or security awareness: "I see the need of security" (#15, product owner). They comprehend that challenges also depend on the role and therefore some interviewees even suggest to define new (agile) security-related profiles such as a "secure product owner" or a "secure system architect". Furthermore, respondents argue that security expertise should generally be improved to achieve compliance. This is exemplified in the following quote: "A new secure product owner could do it" (#7). Interestingly, these observations corroborate the need to raise a common awareness for security in the overall agile team: "implementations deviate from standard [and often] lead to fake implementations" (#2, security compliance *senior expert*); "There are guidelines to bypass compliance rules" (#8, security *principal expert*).

Security Compliance as a Common Agreement: Related to our previous observation is that subjects perceive compliance as a complex endeavour. They noticed that management, teams, and even compliance practitioners have different perspectives on compliance. Some see security compliance as a burnout journey, others as a luxury and others again as a worthwhile goal. A common agreement on the need to achieve common security standards is therefore a prerequisite for the success of our undertaking.

Misunderstandings of Agile Engineering Terms: In our interviews we noticed that terms are used often in a cumbersome manner. For instance, subjects with agile development knowledge (e.g., #1, #2, #3) often referred to Scrum only implicitly by mentioning specific elements such sprint, iteration, and product owner; "definition of done" was often used when referring to acceptance criteria; other interviewees had difficulties in capturing the notion of artefacts in context of process models: "the word artefact is not easy" (#10, *expert*). As a matter of fact, such key concepts are still subject to current debates and need further attention in future work generally dealing with software processes [27].

6 Conclusion

In this paper we reported on our work towards integration of security requirements derived from IEC 62443-4-1 into large-scale agile development based on SAFe in order to facilitate CSC. We presented the S^2C-SAFe framework and evaluated it based on interviews with 16 industry experts. Evaluation results strengthen confidence that this approach and the resulting models provide a feasibly way for security compliance in large-scale organisations practising lean and agile development.

6.1 Impact and Implications

Results show S^2C-SAFe models have a clear impact for practitioners. They show precisely how software engineering and security practitioners have to interact to achieve the goal of security compliance. Furthermore, the models can be understood in a time-effective manner and challenge popular belief that agile processes are a gateway to chaos and therefore not reconcilable with security and compliance concerns. The unanimous response to our work was the exact opposite: Introducing large-scale agile processes demands a culture and mindset change. Even though not our intention, the models helped to convey to sceptical practitioners that both secure and agile development is feasible at scale with reasonable effort.

Our research strongly indicates that models are an excellent way to mediate between agile practitioners and security experts. Particularly visual models allowed them to engage the challenge of continuous security compliance together. Moreover, these models pave the way for analysing various further challenges of the research field: Do models increase the speed of adapting large organisations to secure agile processes at scale? Are models a better way of getting security norms accepted in daily software engineering activities? Can models provide guided and precise support for secure agile security governance? We are confident that our contribution supports researchers to further investigate these questions.

6.2 Relation to Existing Evidence

Our study is in tune with existing trends of empirical studies on secure software engineering [29], but extends the study population in number and profile. To the best of our knowledge, preceding studies involved up to 11 practitioners with mixed background or students as subjects and focused on valuated, yet isolated topics. An integrated view on a security standard compliant agile framework was not in their scope. Our contribution is aimed at this gap and involves 16 experienced professionals, partially with contributing roles to the standards or decision-making roles in the organisation. We focused on the highest ranking experts available. As explained, a SAFe integration may last up to 8 years and the interviewees are high-ranked professionals. Their opinion is the closest to certainty in a timely evaluation.

6.3 Limitations and Threats to Validity

Qualitative studies inherently carry limitations and interview research in particular has threats to validity that need discussion, the most important of which shall be discussed here.

The individual expertise of each participant might influence their attention and interpretation of security requirements as well as agile practices captured in the models. We tried to mitigate this with discussion-intensive preparation procedures, but also by letting subjects interpret the models as they are without any further instruction. We were interested in potential bias towards the subject of security compliance as that reflects on the projects where those models shall be applied.

Similarly, involving experts from each respective field carries the risk of self-selection and confirmation bias. To mitigate this we selected subjects according to typical roles in the target organisation environment instead of their particular interest in the topic. The same is true for which part of S^2C-SAFe they reviewed (requirements, implementation, or testing). We also designed interview plan and questionnaire accordingly and allocated interviewees to models based on previously defined profiles.

Overall, our study already strengthens our confidence in the capability of S^2C-SAFe to integrate security and compliance concerns with lean and agile development. We cordially invite researchers and practitioners to join our endeavour towards facilitating continuous security compliance in large organisations and regulated environments.

Acknowledgements. To the practitioners that evaluate this work and to M. Voggenreiter and F. Angermeir for their accurate review.

References

1. Ahola, J., et al.: Handbook of the Secure Agile Software Development Life Cycle. University of Oulu, Finland (2014)

2. Baca, D., Boldt, M., Carlsson, B., Jacobsson, A.: A novel security-enhanced agile software development process applied in an industrial setting. In: Proceedings of the ARES (2015)
3. Baca, D., Carlsson, B.: Agile development with security engineering activities. In: Proceedings of the ICSSP, pp. 149–158. ACM (2011)
4. Baca, D.: Developing Secure Software -in an Agile Process - Doctoral Dissertation. Blekinge Institute of Technology (2012)
5. Bartsch, S.: Practitioners' perspectives on security in agile development. In: ARES (2011)
6. Beck, K., et al.: Manifesto for agile software development (2001)
7. Beckers, K.: Pattern and Security Requirements - Engineering-Based Establishment of Security Standards. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16664-3
8. Bell, L., Brunton-Spall, M., Smith, R., Bird, J.: Agile Application Security. Enabling Security in a Continuous Delivery Pipeline. O'Reilly, Sebastopol (2017)
9. Beznosov, K., Kruchten, P.: Towards agile security assurance. In: Proceedings of the NSPW. ACM (2004)
10. Cawley, O., Wang, X., Richardson, I.: Lean/Agile software development methodologies in regulated environments – state of the art. In: Abrahamsson, P., Oza, N. (eds.) LESS 2010. LNBIP, vol. 65, pp. 31–36. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16416-3_4
11. Chóliz, J., Vilas, J., Moreira, J.: Independent security testing on agile software development: a case study in a software company. In: Proceedings of the ARES (2015)
12. Daennart, S., Moyon, F., Beckers, K.: An assessment model for continuous security compliance in large scale agile environments - exploratory paper. In: CAiSE (2019)
13. Felderer, M., Pekaric, I.: Research challenges in empowering agile teams with security knowledge based on public and private information sources. In: Proceedings of the SecSe (2017)
14. Fernández, D.M., Wagner, S.: Naming the pain in requirements engineering: design of a global family of surveys and first results from Germany. In: Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering. ACM (2013)
15. Fernandez, E.B.: Threat modeling in cyber-physical systems. In: Proceedings of the 14th International Conference on Dependable, Autonomic and Secure Computing (2016)
16. Fitzgerald, B., Stol, K.J.: Continuous software engineering: a roadmap and agenda. J. Syst. Softw. 1–14 (2015)
17. Fitzgerald, B., Stol, K.J.: Continuous software engineering: a roadmap and agenda. J. Syst. Softw. **123**, 176–189 (2017)
18. Fitzgerald, B., Stol, K.J., O'Sullivan, R., O'Brien, D.: Scaling agile methods to regulated environments: an industry case study. In: Proceedings of the ICSE. IEEE (2013)
19. Humphreys, E.: How to measure the effectiveness of information security (2017). https://www.iso.org/news/2016/12/Ref2151.html
20. IEC: 62443-1-1 Security for Industrial and Automation Control Systems Part 1–1 Models and Concepts. International Electrotechnical Commission, USA, 2014 (2014)
21. IEC: 62443-4-1 security for industrial automation and control systems Part 4–1 product security development life-cycle requirements (2017)

22. ISO/IEC: 27034 Information technology - Security techniques - Application security (2011)
23. ISO/IEC: 27001 IT - Security techniques - Information security management systems (2013)
24. Leffingwell, D., Yakyma, A., Knaster, R., Jemilo, D., Oren, I.: SAFe Reference Guide. Pearson, London (2017)
25. McGraw, G., Migues, S., Chess, B.: Building security in maturity model. https://www.bsimm.com/about.html
26. McHugh, M., McCaffery, F., Fitzgerald, B., Stol, K.-J., Casey, V., Coady, G.: Balancing agility and discipline in a medical device software organisation. In: Woronowicz, T., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2013. CCIS, vol. 349, pp. 199–210. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38833-0_18
27. Méndez Fernández, D., et al.: Artefacts in software engineering: what are they after all? ArXiv e-prints (2018)
28. Moyón, F., Beckers, K., Klepper, S., Lachberger, P., Bruegge, B.: Towards continuous security compliance in agile software development at scale. In: Proceedings of the RCoSE. ACM (2018)
29. Othmane, L., Jaatun, M., Weippl, E.: Empirical Research for Software Security: Foundations and Experience. CRC (2017)
30. Shull, F., Singer, J., Sjøberg, D.I.: Guide to Advanced Empirical Software Engineering. Springer, London (2007). https://doi.org/10.1007/978-1-84800-044-5
31. Siponen, M., Baskerville, R., Kuivalainen, T.: Integrating security into agile development methods. In: Proceedings of the HICSS (2005)
32. Stephanow, P., Khajehmoogahi, K.: Towards continuous security certification of software-as-a-service applications using web application testing techniques. In: Proceedings of the CAINA (2017)
33. Technology, S.A.C.: Security by Design with CMMI for Development Version 1.3. CMMI Institute (2013)
34. Tøndel, I.A., Jaatun, M.G., Cruzes, D.S., Moe, N.B.: Risk centric activities in secure software development in public organisations. IJSSE **8**(4), 1–30 (2017)
35. Turpe, S., Poller, A.: Managing security work in scrum: tensions and challenges. In: Proceedings of the SecSE (2017)