Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

In practice

Continuous software security through security prioritisation meetings[☆]

Inger Anne Tøndel [*], Daniela Soares Cruzes

Norwegian University of Science and Technology (NTNU), Department of Computer Science, Sem Sælandsvei 9, Gløshaugen, 7034 Trondheim, Norway

ARTICLE INFO

Article history:
Received 22 March 2022
Received in revised form 18 June 2022
Accepted 2 August 2022
Available online 18 August 2022

Keywords:
Software security
Security meeting
Security prioritisation
Security requirements
Agile software development

ABSTRACT

Software security needs to be a continuous endeavour in current software development practices. Frequent software updates, paired with an ongoing flow of security breaches, requires software companies to address software security throughout development and post deployment. Prescriptive software security approaches do not match well with agile software development and its emphasis on self-management. Agile approaches are however in favour of meetings as a coordination and problem-solving strategy. This article investigates the role of regular security meetings centred on making security priorities and decisions for achieving continuous software security. Through technical action research and an observational case study, we studied variations of such meetings in three companies. We found that such meetings can reach key stakeholders, make security more visible, and contribute to ongoing security prioritisation. Thus, security meetings are a promising approach, especially for small and medium sized development companies with basic yet immature security competence. Future research should investigate further the role of such meetings and how best to organise them for different contexts and needs. For this we outline implications for research and practice, e.g., related to participants and how to organise the discussions and prioritisations in the meeting.

1. Introduction

Contemporary software development happens as a continuous flow of software development rather than as larger planned increments. Further, software products are updated throughout their whole lifetime, to meet customer demands for new and improved features and to ensure continuous quality. Pairing this with daily news of security breaches, it becomes clear that software security needs to be a continuous endeavour as well.

As Fitzgerald and Stol (2014), we use *continuous* to represent a "holistic endeavor" and the "entire software life-cycle". They define continuous security as, "Transforming security from being treated as just another non-functional requirement to a key concern throughout all phases of the development lifecycle and even post deployment" (Fitzgerald and Stol, 2014).

Continuous security does not come without effort. For security experts it would thus be tempting to prescribe security activities and tools to use during development and beyond, to ensure security is properly addressed. Research, however, shows that such prescriptive approaches are challenging to pair with the self-management of agile software development (ASD) (T̀urpe and Poller 2017; (Weir et al., 2020a), and rather suggest "sensitizing the developers to their security needs, allowing them to choose for themselves which tools and techniques to use" (Weir et al., 2020a).

There are several frameworks and maturity models available for software companies wanting to continuously address software security, here exemplified by the OWASP Software Assurance Maturity Model (SAMM) (Crawley et al., 2020) and the Building Security In Maturity Model (BSIMM) (Migues et al., 2021). Both are agnostic to the development approach, and thus are relevant also for ASD (van der Veer, 2019). However, the comprehensiveness of these models (e.g., BSIMM12 now consists of 122 activities within the domains of governance, intelligence, SSDL touchpoints, and deployment) can make them hard to approach, especially for smaller companies that do not necessarily have the resources to build a large security program (Tøndel et al., 2020). And research points to small and medium-sized enterprises (SMEs) as having the largest potential for improvement of software security (Weir et al., 2020a). Further, knowing which activities to apply is not straight-forward. Being too ambitious may lead to an overspending on security (Tøndel et al., 2020), which can have negative business implications. For companies, what is considered adequate and cost-effective when it comes to security may vary between projects and change over time (McGraw et al., 2013; Tøndel et al., 2020), as development progresses and requirements

[☆] Editor: Dr. Daniel Mendez.

[*] Corresponding author.

E-mail addresses: inger.anne.tondel@ntnu.no (I.A. Tøndel), daniela.s.cruzes@ntnu.no (D.S. Cruzes).

https://doi.org/10.1016/j.jss.2022.111477

are negotiated. Thus, there is a need for strategic decisions on security on a regular basis.

ASD is oriented towards people, interactions, and self-management (Beck et al., 2001), with meetings as a major mean of coordination (Strode et al., 2012). To exemplify, Scrum (Schwaber, 2004) is largely centred on meetings and include five meeting types: sprint planning meeting, daily Scrum meeting, sprint review meeting, sprint retrospective meeting, and product backlog refinement. None of these meetings are focused on security. Some researchers have proposed regular meetings on security in ASD, in form of adding security review meetings to Scrum when necessary (Kongsli, 2006), using Protection Poker to collectively estimate the security risk in every iteration (Williams et al., 2010), or organising a Security Intention Meeting Series to regularly involve decision makers in security prioritisation (Tøndel et al., 2019a). Further, less regular meetings are proposed in form of Threat Modelling sessions (Bernsmed et al., 2022) or security workshops aimed toward security incentivisation (Weir et al., 2020a). Many of the activities included in BSIMM or OWASP SAMM could be performed through or supported by regular meetings, examples being Threat Assessment, Security Architecture, and Architecture Assessment in OWASP SAMM. However, existing research points to uncertainties in how to best organise such meetings to ensure effect on the software security (Cruzes et al., 2018; Tøndel et al., 2019b). A better understanding is needed on what are the effects of security meetings related to development, and how practitioners can be guided in ensuring effect from meetings. Furthermore, previous studies have identified challenges regarding longer-term adoption of security meetings (Tøndel et al., 2019b; Weir et al., 2020a, 2021; Bernsmed et al., 2022), leading to the need for more knowledge of how to support ongoing adoption.

This article proposes and studies regular security meetings that: (1) are not confined to security experts but rather include key decision makers as participants, (2) identifies and assesses security needs, and makes prioritisations and decisions on the next steps, and (3) are flexible and can be adjusted to the needs and priorities of the company when it comes to meeting scheduling and organisation. As such, we build on the previous suggestions for a Security Intention Meeting Series (Tøndel et al., 2019a). The research we report on is part of a design science study aimed at improving software security prioritisation by developing a meeting approach that satisfies the needs of ASD projects. We study such security meetings in three SMEs, one of which were already running this type of meeting, and two where we brought this meeting type to the company. Our main research question is the following: *How can regular security meetings centred on making security prioritisations and decisions be organised to maximise their positive effect on the priority given to security? (RQ1).*

We have previously described the concept *security prioritisation* as "prioritisation among security requirements and activities, prioritisation of security vs. other aspects such as functionality, as well as the priority and attention given to security in the day-to-day work". Thus, we take a broad view of security prioritisation. To relate to this rather intangible concept of security prioritisation, it is necessary to concretise what security prioritisation may look like in a project, and what can be done to influence the priority given to security. As part of this design science study, we have previously performed a case study to investigate what influences the security prioritisation throughout an ASD project. We found that the priority given to security was influenced by the presence of a driving force for security, the visibility of security, the motivation, the room to manoeuvre, and the process match (Tøndel et al., 2022). In the study reported in this paper, we use these previously identified influence categories to support us in understanding how the studied meetings can have a positive effect on the priority given to security. Additionally, we study *what effects are seen by adopting this type of meeting (RQ2)* and *what facilitates or hinders the adoption of such meetings (RQ3).*

The studied meeting instances varied in their structure, the support offered, and in who participated. The companies varied in size, development approach, and in customer relations. This variety allowed us to identify similarities and variations across the cases, and use this to understand: (1) what are common experiences that a broader set of companies might expect from applying this meeting concept, and (2) how key variations can be explained based on the cases. The article contributes both to practice and research: (1) we use the lessons learned to provide development companies with better support in deciding whether to take up regular security meetings in development, and how to organise such meetings, and (2) we provide researchers with insight into the practical experiences in performing such meetings and point to research needs.

The article is organised as follows. Section 2 uses literature to motivate regular software security meetings, as well as presents current knowledge on security meetings in ASD. Section 3 describes the research approach, including the cases studied. Section 4 presents the findings according to the three research questions. Section 5 discusses the implications of these findings, Section 6 discusses the threats to validity, and Section 7 concludes the article.

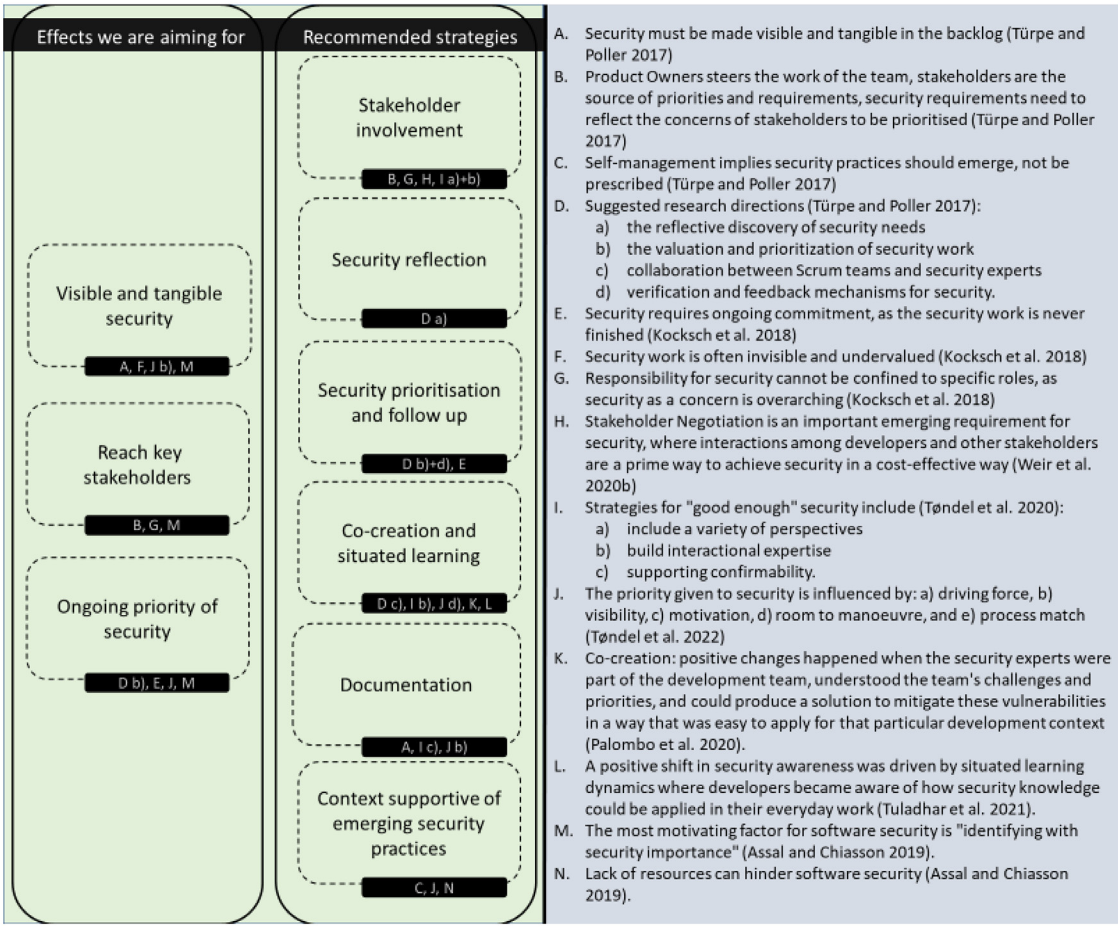
2. Background and related work

This section uses current literature on software security and on meetings in ASD to explain the theoretical background for investigating regular security meetings. Furthermore, it describes the known evidence from studies of similar types of meetings, and introduces in more detail the Security Intention Meeting Series that we build on in this work.

2.1. Background for investigating regular security meetings

There is a growing body of literature on how to integrate security and other software qualities with ASD. This includes literature on working with security requirements in ASD (Villamizar et al., 2018; Tøndel and Jaatun, 2020), managing quality requirement sin ASD (Behutiye et al., 2020); (Jarzębowicz and Weichbroth, 2021), and on bringing ASD to safety-critical systems (Heeager and Nielsen, 2018). Literature reviews within this area point to some recurring challenges of working with security and other quality requirements in ASD. One challenge identified by many studies is that of neglect of quality requirements (Behutiye et al., 2020); (Jarzębowicz and Weichbroth, 2021). Related challenges are that of late consideration of quality requirements (Behutiye et al., 2020) and a lack of recognition by stakeholders (Jarzębowicz and Weichbroth, 2021). Proposed solutions include initiatives to start focus on quality requirements earlier in the project, and to involve multiple roles and viewpoints in eliciting and reviewing quality requirements (Jarzębowicz and Weichbroth, 2021). Still, a main criticism towards much of the existing research in this field is the limited empirical evaluations of proposed techniques and approaches to integrate software security into ASD (Villamizar et al., 2018; Bishop and Rowland, 2019; Behutiye et al., 2020). There is a call for more guidelines, not only practices and methods (Behutiye et al., 2020). Furthermore, there is still a need for more lightweight strategies, as the challenge of time constraints due to short iterations have not received adequate attention in the proposed strategies, despite this being a commonly reported challenge (Behutiye et al., 2020).

Literature provides some knowledge on what can increase the priority given to security in an agile development context. Newton et al. (2019) studied literature and performed interviews to



A. Security must be made visible and tangible in the backlog (Türpe and Poller 2017)
B. Product Owners steers the work of the team, stakeholders are the source of priorities and requirements, security requirements need to reflect the concerns of stakeholders to be prioritised (Türpe and Poller 2017)
C. Self-management implies security practices should emerge, not be prescribed (Türpe and Poller 2017)
D. Suggested research directions (Türpe and Poller 2017):
   a) the reflective discovery of security needs
   b) the valuation and prioritization of security work
   c) collaboration between Scrum teams and security experts
   d) verification and feedback mechanisms for security.
E. Security requires ongoing commitment, as the security work is never finished (Kocksch et al. 2018)
F. Security work is often invisible and undervalued (Kocksch et al. 2018)
G. Responsibility for security cannot be confined to specific roles, as security as a concern is overarching (Kocksch et al. 2018)
H. Stakeholder Negotiation is an important emerging requirement for security, where interactions among developers and other stakeholders are a prime way to achieve security in a cost-effective way (Weir et al. 2020b)
I. Strategies for "good enough" security include (Tøndel et al. 2020):
   a) include a variety of perspectives
   b) build interactional expertise
   c) supporting confirmability.
J. The priority given to security is influenced by: a) driving force, b) visibility, c) motivation, d) room to manoeuvre, and e) process match (Tøndel et al. 2022)
K. Co-creation: positive changes happened when the security experts were part of the development team, understood the team's challenges and priorities, and could produce a solution to mitigate these vulnerabilities in a way that was easy to apply for that particular development context (Palombo et al. 2020).
L. A positive shift in security awareness was driven by situated learning dynamics where developers became aware of how security knowledge could be applied in their everyday work (Tuladhar et al. 2021).
M. The most motivating factor for software security is "identifying with security importance" (Assal and Chiasson 2019).
N. Lack of resources can hinder software security (Assal and Chiasson 2019).

Fig. 1. Literature supporting the suitability of regular security meetings to achieve prioritisations and decisions regarding security.

identify success factors for software security in ASD. They pointed out the importance of practices centred around individuals and culture. This is in line findings in the previously mentioned case study (Tøndel et al., 2022), where we identified five areas that influence software security prioritisation in an ASD project. Many of these influence areas (driving force, visibility, motivation, room to manoeuvre, and process match) are related to individuals and culture.

Türpe and Poller (2017) explored tensions between Scrum and security requirements through a case study. They identified challenges related to key stakeholders such as product owners. They pointed out that it is important to make security visible as a concern, and they emphasised the need for more research on prioritisation of security and on improving collaboration between agile teams and security experts. Drawing on data from ethnographic studies, Kocksch et al. (2018) explained how security's similarities with care work (invisible, undervalued, never finished) can make recognition of security work challenging. Furthermore, they pointed out how security needs to be handled in collaboration among a broad set of stakeholders. Building on theory on objectivity, Tøndel et al. (2020) identified the inclusion of a variety of perspectives and the building of interactional expertise as key strategies to move towards "good enough" security.

Assal and Chiasson (2019) performed a survey aimed at understanding how to support developers in their work on software security. They found that developers need to identify with the importance of security. In this respect, the ethnographic studies of Palombo et al. (2020) and Tuladhar et al. (2021) pointed to the role of co-creation and situational learning (respectively) in changing software security practices. It is not enough to point developers to the presence of security problems. Rather, changes are seen when developers and security experts work together to solve problems. This is in line with Weir et al. (2020b), who identified the need to move towards a "dialectic" approach to security. Building on data from interviews they identified interactions among developers and other stakeholders as a prime way to achieve security in a cost-effective way.

Fig. 1 provides our summary and synthesis of the findings from the above-mentioned studies. On the right side of this figure, we list key findings from the studies. Then, on the left side of the figure, we synthesise these findings into a set of effects called for in literature, and a set of recommended strategies. Letters are used to link the findings from the studies (on the right) with the effects/strategies (on the left). As shown in the figure, there is a call for more visible and tangible security (Türpe and Poller, 2017; Kocksch et al., 2018; Assal and Chiasson, 2019; Tøndel et al., 2022), for reaching key stakeholders with security (Türpe and Poller, 2017; Kocksch et al., 2018; Assal and Chiasson, 2019), and for making security an ongoing priority (Türpe and Poller, 2017; Kocksch et al., 2018; Assal and Chiasson, 2019; Tøndel et al., 2022). Recommended strategies for security include stakeholder involvement (Türpe and Poller, 2017; Kocksch et al., 2018; Tøndel et al., 2020; Weir et al., 2020b), security reflection (Türpe and Poller, 2017), security prioritisation and follow-up (Türpe and Poller, 2017; Kocksch et al., 2018), co-creation and situated learning (Türpe and Poller, 2017; Palombo et al., 2020; Tøndel et al., 2020; Tuladhar et al., 2021; Tøndel et al., 2022), documentation for security (Türpe and Poller, 2017; Tøndel et al., 2020, 2022), and having a context supportive of emerging security practices

(Türpe and Poller, 2017; Assal and Chiasson, 2019; Tøndel et al., 2022)

Security meetings are not a goal, but a possible mean to achieve ongoing and strategic prioritisation of security. Regular security meetings are likely to be able to support the effects and strategies identified in Fig. 1, e.g., by involving stakeholders on security and engaging them in regular security reflection, prioritisation, and follow-up. Furthermore, addressing security through meetings is highly compatible with an agile approach. According to the Agile manifesto, "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation" (Beck et al., 2001). As already mentioned, Scrum relies on several meetings (Schwaber, 2004). Daily stand-up meetings are also recommended within Kanban (Ahmad et al., 2013). In ASD, meetings support teamwork quality through balancing member contributions and facilitating mutual support (Lindsjørn et al., 2016), and offers a major mean of coordination (Strode et al., 2012; Moe et al., 2018).

2.2. Related work on regular security meetings in ASD

Despite meetings playing a central role in ASD, there is limited research on the effect and organisation of meetings that involve activities related to ongoing prioritisation, planning, and follow-up. Of such meetings specific to ASD, the daily stand-up meeting has been most extensively studied by Stray et al. (2016).

When it comes to security meetings directed towards agile or continuous software development, we have identified the following approaches:

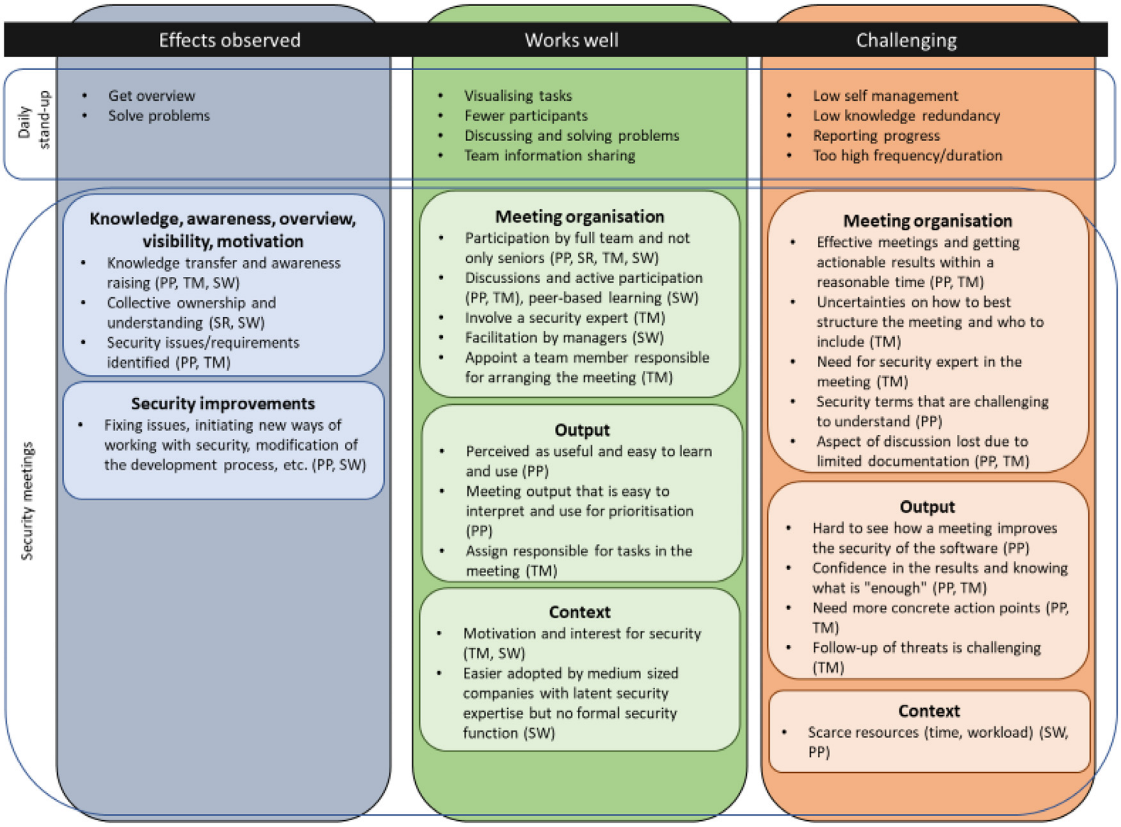
- Security Review (SR) meeting (Kongsli, 2006): Arranged after the Scrum iteration planning meetings in cases where there were many or complex security concerns related to the user stories that were picked. The full development team participated in this meeting to ensure collective ownership also of security issues.
- Protection Poker (PP) (Williams et al., 2009, 2010; Tøndel et al., 2019b): A collaborative risk estimation game that gathers the whole development team to discuss, identify and rank the software security risks related to the features to be implemented in the upcoming iteration.
- Threat Modelling (TM) meetings, as studied by Cruzes et al. (2018) and Bernsmed et al. (2022): Meetings centred on performing threat modelling, e.g., using Data Flow Diagrams and the STRIDE mnemonic.
- Facilitated Security Workshops (SW) (Weir et al., 2020a, 2021): Workshops centred on incentivisation, threat assessment, and on-the-job training. Used the Agile Security Game, a simplified threat assessment, and follow-up sessions, to discuss security issues and questions. Workshops were led by researchers who were not security experts, to study the effect of this workshop package also when there were no security experts available.
- The Security Intention Meeting Series (Tøndel et al., 2019a): A meeting series to gather key decision makers in a project, to regularly assess the state of the software security of the project and identify concrete actions moving forward.

Fig. 2 gives an overview of identified effects from these meeting types, as well as what has been found to work well or be challenging. The figure is organised so that findings from studies of the security meetings are presented together, while findings on the daily stand-up meetings are presented separately. In the following, we first present aspects related to meeting organisation, before moving on to output and effect, and, finally, point to facets of the context.

When it comes to meeting organisation, many of the aspects of the meetings that worked well were related to strategies called for in existing literature (see Fig. 1). Examples are stakeholder involvement (participation by the full team Kongsli, 2006; Williams et al., 2010; Weir et al., 2020a; Bernsmed et al., 2022, facilitation by managers Weir et al., 2021), security reflection (discussions and active participation Tøndel et al., 2019b; Bernsmed et al., 2022), and co-creation and situated learning (peer-based learning Weir et al., 2021). This points to meetings as a powerful intervention, which should be properly addressed in research. Broad participation and discussions were pointed out as important across the different security meetings. Broad participation however came with the risk of less effective meetings (Stray et al., 2016). Both Protection Poker and Threat Modelling found clear needs for some security expertise, e.g., to be able to explain terms and ensure quality (Cruzes et al., 2018; Tøndel et al., 2019b; Bernsmed et al., 2022). This contrasts with the Security Workshops, which had as a requirement that they should work also without security experts (Weir et al., 2020a). The challenges and uncertainties identified related to meeting organisation (e.g., uncertainties on how to structure the meeting and who to include in order to make the meetings effective Cruzes et al., 2018; Tøndel et al., 2019b) points to a need to further explore different meeting types and collect more experiences to guide both researchers and practitioners. This article meets this need.

When it comes to effect and output, both the daily stand-up meeting and the security meetings could help get overview of issues and solve problems/make improvements. Related to the desired effects outlined in Fig. 1, the meetings generally contributed to making security more visible and tangible. Despite the meetings leading to security improvements in processes and code, studies point to challenges in following up the risks identified in the meetings and seeing how the meeting output is linked to improved security of the products (Cruzes et al., 2018; Tøndel et al., 2019b). There is a need to understand better what makes this transition challenging, so that better guidance can be offered. Existing literature (see Fig. 1) points to following up of prioritisations, something that can be done in meetings. However, there are likely more complex reasons that make this transition challenging. To exemplify, both Palombo et al. (2020) and Weir et al. (2020a) emphasise the systemic aspect of software security. The study reported on in this article examines how effects from meetings can be supported or hindered (RQ1), considering aspects of the meeting as well as the context.

When it comes to the context, the studies of these meetings pointed to the importance of motivation and time for security (Tøndel et al., 2019b; Weir et al., 2020a; Bernsmed et al., 2022). Further, the company size and the security maturity level might be important for longer-term adoption. Weir et al. (2021), who studied eight organisations of varying size, found that adoption was strongest in the medium-sized organisations, followed by the smaller organisations, while adoption was low in the larger organisations. However, we have reason to believe that a broader set of contextual factors have implications for adoption of meetings and their effect. We base this expectation on the large number of documented challenges to software security and other quality aspects in ASD (Oueslati et al., 2015; Behutiye et al., 2020; Jarzębowicz and Weichbroth, 2021), as well as the substantial amount of organisational blockers and motivators (Weir et al., 2020a) and influences (Tøndel et al., 2022) identified for adopting and prioritising software security practices. In this article we contribute with more knowledge on contextual factors important for getting effect and adoption of regular security meetings in development companies or teams. Further, as time has been identified as one key obstacle, we study meeting approaches where the schedule can be adapted to the time pressure experienced in the company.



	Effects observed	Works well	Challenging
Daily stand-up	• Get overview • Solve problems	• Visualising tasks • Fewer participants • Discussing and solving problems • Team information sharing	• Low self management • Low knowledge redundancy • Reporting progress • Too high frequency/duration
Security meetings	**Knowledge, awareness, overview, visibility, motivation** • Knowledge transfer and awareness raising (PP, TM, SW) • Collective ownership and understanding (SR, SW) • Security issues/requirements identified (PP, TM) **Security improvements** • Fixing issues, initiating new ways of working with security, modification of the development process, etc. (PP, SW)	**Meeting organisation** • Participation by full team and not only seniors (PP, SR, TM, SW) • Discussions and active participation (PP, TM), peer-based learning (SW) • Involve a security expert (TM) • Facilitation by managers (SW) • Appoint a team member responsible for arranging the meeting (TM) **Output** • Perceived as useful and easy to learn and use (PP) • Meeting output that is easy to interpret and use for prioritisation (PP) • Assign responsible for tasks in the meeting (TM) **Context** • Motivation and interest for security (TM, SW) • Easier adopted by medium sized companies with latent security expertise but no formal security function (SW)	**Meeting organisation** • Effective meetings and getting actionable results within a reasonable time (PP, TM) • Uncertainties on how to best structure the meeting and who to include (TM) • Need for security expert in the meeting (TM) • Security terms that are challenging to understand (PP) • Aspect of discussion lost due to limited documentation (PP, TM) **Output** • Hard to see how a meeting improves the security of the software (PP) • Confidence in the results and knowing what is "enough" (PP, TM) • Need more concrete action points (PP, TM) • Follow-up of threats is challenging (TM) **Context** • Scarce resources (time, workload) (SW, PP)

Fig. 2. Key findings from studies of the daily stand-up meeting, as well as the Security Review Meeting (SR), Protection Poker (PP), Threat Modelling (TM), and facilitated security workshops (SW).

2.3. The security intention meeting series

In this study, we decided to examine meetings that followed the spirit set out for the Security Intention Meeting Series (Tøndel et al., 2019a) but with freedom for companies to adapt the approach to match their needs. This is based on recommendations that developers should not be prescribed a particular way of addressing security, but rather be empowered to make their own decisions (Türpe and Poller, 2017; Weir et al., 2020a). The Security Intention Meeting Series approach was a response to challenges with getting companies to consider security in every iteration, e.g., as is done in Protection Poker, while needing a more lightweight and recurring approach than what is common for threat modelling and security risk analysis (Tøndel et al., 2019a). Further, this meeting approach had not yet been studied empirically.

The Security Intention Meeting Series approach (Tøndel et al., 2019a) can be summarised as follows. Early in the project, key decision makers are gathered, together with people knowledgeable about security and development, to discuss the security intentions of the project. This implies agreeing on the security goals and needs of the project, and what aspects need to be given particular attention (*intention setting*). Then, regular follow-up meetings are arranged throughout. These consists of a *status assessment* ('Are we moving towards our goals regarding software security?') and an identification of *action points* for the next period ('What will be our concrete priorities moving forward?'). Companies and projects are, however, free to adapt the meeting approach to their needs, e.g., regarding how often to arrange such meetings, who should facilitate the meetings, and who should be invited as participants. Still, some guidance is given. Shorter meetings are preferred to longer ones, there should be some regularity to the meetings (e.g., by always agreeing on a time for the next meeting as part of the meeting), one person should be responsible, and roles such as product owner or project manager should be present.

3. Research approach

First, this section describes the overall research approach of this work as that of design science, and explains how technical action research and case studies were used in combination to support the design goal. Then, it introduces the research design choices that were made for both the technical action research and case study research parts of the study, before it describes each of these parts in more detail. Finally, the analysis approach is presented.

3.1. Combining technical action research and observational case studies into a design science research approach

The main research question of this work represents a design goal, aimed at improving software security prioritisation by developing a meeting approach that satisfies the needs of agile development projects. Consequently, the overall research approach of this work is design science (Wieringa, 2014). Design science "is the design and investigation of artifacts in context", where the artefacts "interact with a problem context in order to improve something in that context" (Wieringa, 2014). Thus, design science iterates between two problem-solving activities: (1) designing artifacts to bring about improvements, and (2) answering knowledge questions about the context and the artifact in the context. To answer the knowledge questions, the researcher can bring in other research methods (Wieringa, 2014). Two of the possibilities are observational case studies ("a study of a real-world case without performing an intervention" (Wieringa, 2014) and technical

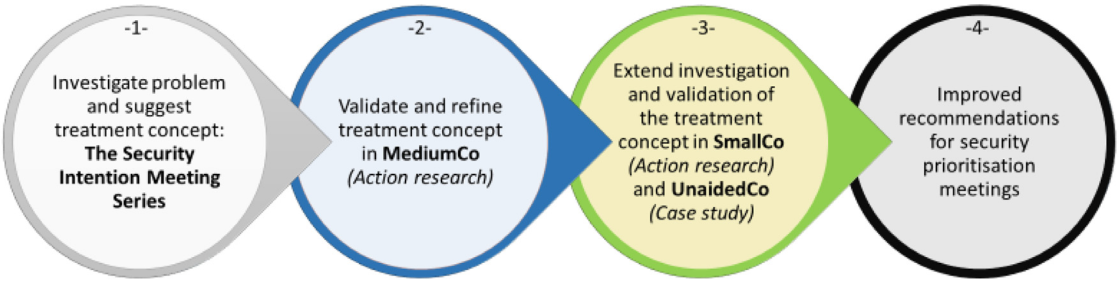

-1-
Investigate problem and suggest treatment concept: **The Security Intention Meeting Series**
-2-
Validate and refine treatment concept in **MediumCo** *(Action research)*
-3-
Extend investigation and validation of the treatment concept in **SmallCo** *(Action research)* and **UnaidedCo** *(Case study)*
-4-
Improved recommendations for security prioritisation meetings

Fig. 3. The overall research approach, using design science with technical action research and observational case studies to validate and refine the recommendations for security prioritisation meetings.

action research ("the use of an experimental artifact to help a client and to learn about its effects in practice" (Wieringa, 2014). Both observational case studies and technical action research can provide understanding of the underlying mechanisms that produce real-world phenomena (Wieringa, 2014). Thus, both these research approaches were suited to understand the adoption and the effects of these meetings (the artefact) and what brought about this adoption and effect.

The overall design science approach is depicted in Fig. 3. The concern of this article is step 2–4 in this figure. The decision to combine action research and case study research was pragmatic; we used technical action research with the companies we studied that did not already perform regular security prioritisation meetings and used observational case study with a company that was already performing such meetings. When action research was applied, our goal in the study was both to help the company with their security and to validate and refine the meeting concept, and we iterated upon and adapted the meeting approach throughout the study to better meet the needs of the company. When case study research was applied, we studied a meeting approach already in use by the company without aiming to improve upon the approach or help the company in their approach to software security.

3.2. Overarching research design choices

The Security Intention Meeting Series approach (Tøndel et al., 2019a) that we wanted to validate and refine through this study had already been designed. However, as our goal was to improve this approach, we were free to adapt this meeting concept to the needs of the companies we interacted with. We see this as a strength of our study. There is no one single way to achieve security prioritisation through regular security meetings. The size of the company and the type of project and customer might impact both how to run the meeting effectively, what kind of support would be needed, e.g., in form of a template, and who should participate. Thus, the meetings we studied all shared key characteristics with the original idea of the Security Intention Meeting. Still, we allowed for variation from these characteristics on some aspects depending on the needs of the company. Thus, there are some discrepancies in who participated in the meetings and in the setting of intentions, compared to the original design of the approach. An overview of the characteristics of the studied meeting models is given in Table 1.

To perform this study, we needed to recruit companies with: (1) an intention to perform ongoing security prioritisation, and (2) an ongoing security prioritisation meeting initiative or a wiliness to initiate such a meeting initiative. Through a research project with several company participants, we had access to several cases that matched these needs, and we opted to involve three companies in the study. The companies have been given the pseudonyms MediumCo, SmallCo, and UnaidedCo for the purpose of this article. An overview of characteristics of these companies is given in Table 2. This variety of companies allowed us to evaluate this meeting type in companies ranging from very small to medium size, and with different customer relations that in varying ways constrained their ability to incorporate regular security meetings as part of development. Further, it allowed us to study how to start applying such a meeting, as well as study a meeting that had already been successfully adopted by a development company. The choice to focus on SMEs was guided by literature showing smaller companies have a larger potential for improvements in their software security than larger companies, and showing more success with security meetings in smaller companies (Weir et al., 2020a).

For all cases, we used multiple data collection methods, as is recommended for case studies (Yin, 2018). An overview is given in Table 3. The action research study was centred on meetings that we facilitated and observed. Meeting observations were supplemented with other data sources, including interviews. For the observational case study, we used a similar approach where observation of meetings was a central part of data collection. All data collection was done by the first author, and in the cases where we used action research, the second author had the role of facilitator of the meetings. In MediumCo and UnaidedCo the observer was largely passive in meetings, while the observer participated more to the discussion in SmallCo. The first author took detailed notes from all security meetings, including notes on the structure of the meetings, the topics that were discussed, the types of security decisions and priorities made, the participants' level of engagement, what worked well, what was challenging, and if anything was surprising. We also reflected on the potential influence of the observer in the meeting. Interviews were semi-structured, and covered topics such as the goal of the security meetings, their effect, how the meetings could be improved, and the intention to continue with the meetings. With UnaidedCo, the retrospective was led by the first author and covered similar topics as the interviews. More information on the data collection instruments is given in Appendix B.

The main ethical aspects of this study are the privacy of the individuals participating in the study, the sensitive information on the security of the solutions as shared with us in meetings, and ensuring volunteer participation in the study. Privacy related to data collection and analysis was specified in a report sent to the Norwegian Centre for Research Data, an organisation that provides data protection services to Norwegian research institutions. This ensured that data handling plans were in accordance with current privacy legislation. Observation notes were made in such a way that individuals were not directly identified. Interviews were only recorded upon interviewees informed consent. When it comes to security of sensitive company data, this was ensured through non-disclosure agreements (NDAs) with the companies. In observations, we took care not to write down what the participants pointed out as highly sensitive. Access to the raw data and

Table 1
Meeting model characteristics.

Meeting model characteristics	MediumCo	SmallCo	UnaidedCo
Meeting maturity	Initiated in one project, then continued in another project, and eventually brought to another team.	Not used before.	Had run this type of meeting for 10 months when we started observation.
Scope	Project/team	Project	Department
Participants from the company	*Initial use:* 3–5 people from the following roles: security resources (security officer, security champion), product owners (mainly those with more technical background), developer representatives. *Brought to new team:* the product owner and the full team.	Developers (1–2)	Department lead and system architects (5)
Physical/online facilities	*Initial use:* mixture of physical and online participants, shared screen *Brought to new team:* online meeting with shared screen	Online meeting with shared screen	Physical meeting, one location, screen shown in meeting
Facilitation	*Initial use:* security officer *Brought to new team:* product owner	External security expert	Department lead
Frequency	*Initial use:* Monthly — time for next meeting decided upon in the meeting *Brought to new team:* NA	Biweekly — time for next meeting decided upon in the meeting	Monthly
Duration	45 min–1 h	Initially 1 h. Later 30 min.	2 h scheduled, usually spent 1 h and 30–45 min
Support material	Confluence page with security areas and supporting questions.	Excel sheet with security areas and supporting questions.	NA
Meeting documentation	Confluence page: concerns and action points added within the structure of the support material	Excel sheet: concerns and action points added within the structure of the support material	Meeting memos with action points + excel sheet with overview of all identified security concerns that were not yet fully addressed
Typical agenda	(1) Status of tasks and open issues from previous meetings; (2) Discuss security areas not previously addressed, or where there are open issues still; (3) Time for next meeting.		(1) Status of action points from last meeting; (2) Open discussion on security issues; (3) Excel sheet with security concerns; (4) New action points.

Table 2
Company characteristics.

Company characteristics	MediumCo	SmallCo	UnaidedCo
# developers	About 80 developers	2–3 developers	About 20 developers
# locations with developers	4	1	1
Criticality of security	*(Medium)* Clear security risks, mainly related to offering a public service to many users, and in ensuring validity of tickets.	*(Low)* Limited security risks but with plans to develop new solutions that brings in both privacy and security concerns.	*(High)* Develops solutions that handle security critical information.
Customer relation	Targets mainly one sector. Bid process in competition with other actors. Varying security concerns among customers.	Several smaller customers without much security competence and concerns. New bigger customer upcoming.	One main customer (the mother company) that is concerned about security.
Agile principles adherence	Hybrid. Scrum-based development process but with rather fixed contracts.	Few developers, thus few clear processes.	Hybrid. Kanban-based processes within the development department.
Presence of central security support	Initially: security officer as part of the development department and security champions in development teams. Later: key resources left without being replaced.	NA	Chief Security Officer in mother company (the customer). One person in the development department with an informal security role.

Table 3
Overview of data collection.

Data collection	MediumCo	SmallCo	UnaidedCo
Main research approach	Action research	Action research	Case study
Observations of security meetings *(documented in an observation template)*	7 (one of which were shortened/largely skipped) (11/2019–08/2020)	11 meetings (10/2020–03/2021)	4 (09/2019–03/2020)
Other observations *(documented in an observation template)*	2 meetings to bring this technique to new team	3 introductory meetings before starting with the meeting template; 1 visit to the company	NA
Interviews *(interviews at MediumCo and SmallCo were recorded and transcribed upon interviewee consent; interviews and retrospective at UnaidedCo were performed by two researchers, where one was responsible for taking notes)*	Interviews with two product owners after participating in meetings (online) (06/2020)	Interview with the main developer after participating in meetings (online) (05/2021)	Interview with 2 meeting participants and 3 outsiders before observations started (in-person) (04/2019); retrospective after observations (2 sessions, 4 participants from the meeting + one outsider, in-person but with one online participant) (09/2020)
Status updates *(documented in notes or in emails)*	6 informal talks with the security officer (08/2019–06/2020); email exchange with one product owner (08/2020)	Email exchange with main developer on adoption of the technique 10 months after the other data collection (01/2022)	NA
Documentation	Example confluence page; description of security areas and security questions	Excel sheet used as template for meetings; description of the meeting approach written by the main developer; some of the security material developed as a result of the meeting	NA

the analysed data was limited to a few individuals. Participation in the study was voluntary for the companies and the individuals. However, we recognise that for MediumCo and SmallCo, participation in the study led to them getting support in their software security work. This may have made it more difficult for them to refuse participation. For all companies, it might also be challenging for individuals to refuse participation if the company was part of the study. However, we got the impression that participants were positive towards contributing to this research.

3.3. Technical action research: MediumCo and SmallCo

Technical action research relies on mutual trust, and such trust can take a long time to establish (Wieringa, 2014). When we recruited the first company, MediumCo, the meeting concept had not yet been tried out in a company and we thus needed to work with a company where we had predefined trust to try out new ways of working. MediumCo matched this need. Furthermore, it was a good case as it had characteristics that we suspected to be common among SME; it had few dedicated security resources, an ongoing yet immature software security initiative, cross-border working arrangements, and operated in a strongly competitive business. Thus, it was a relevant case to study, also if we would end up with a single-case design (Yin, 2018). We had worked with MediumCo before and had good knowledge of the company and its context, something that reduced one of the common challenges in canonical action research, i.e., to deal with the organisational complexity when diagnosing the current situation (Davison et al., 2012). This also made it easier for us to make an initial instantiation of the meeting concept that matched this company

MediumCo and its development was organised in several teams, and projects could be performed by one or more teams. Each of these teams had one or two product owners associated with the team. A security officer role was part of the development department, overseeing and supporting the security work in the teams. In addition, each team had one assigned security champion – a developer with extra attention to security issues. We started working with a team (in the following referred to as Team A) where the product owners were technically skilled and interested in security, thus making them open to try this technique. The security officer was active in adapting the security intention meeting concept to the needs of MediumCo.

In the instantiation of the security intention meeting for MediumCo, the status assessment was supported by a checklist consisting of some general questions and a long list of security areas to consider (See Appendix A). This way the meeting participants were supported in identifying all the important security issues to be considered. However, this made the intention setting be more technical and thorough than originally envisioned (Tøndel et al., 2019a). In the meetings, participants assessed the status of already identified security tasks, discussed security needs and progress related to the security areas in the checklist, and identified priorities moving forward, including a time for the next meeting. Participants were product owners, the security officer, and key representatives from development.

As illustrated in Fig. 4, we facilitated and observed five meetings in Team A. Then we performed interviews with the two product owners that were considered the key participants in these meetings. Based on the experiences from Team A, the company wanted to bring the meeting to another team (Team B). An introduction to the meeting approach was given by one of the product owners involved in Team A. In this case, no researcher was involved as facilitator. Instead, the Product Owner of Team B did the facilitation. The meeting concept was the same, however, in this case participants included the full team.

Experiences from MediumCo made us interested to see how this meeting approach would work in a company with less resources dedicated to security. This led to the recruitment of SmallCo, a very small development company with close to no previous experience in software security. They were motivated to participate, as envisioned changes to their software product portfolio made it necessary to incorporate more security into their development activities. Thus, working with SmallCo represented

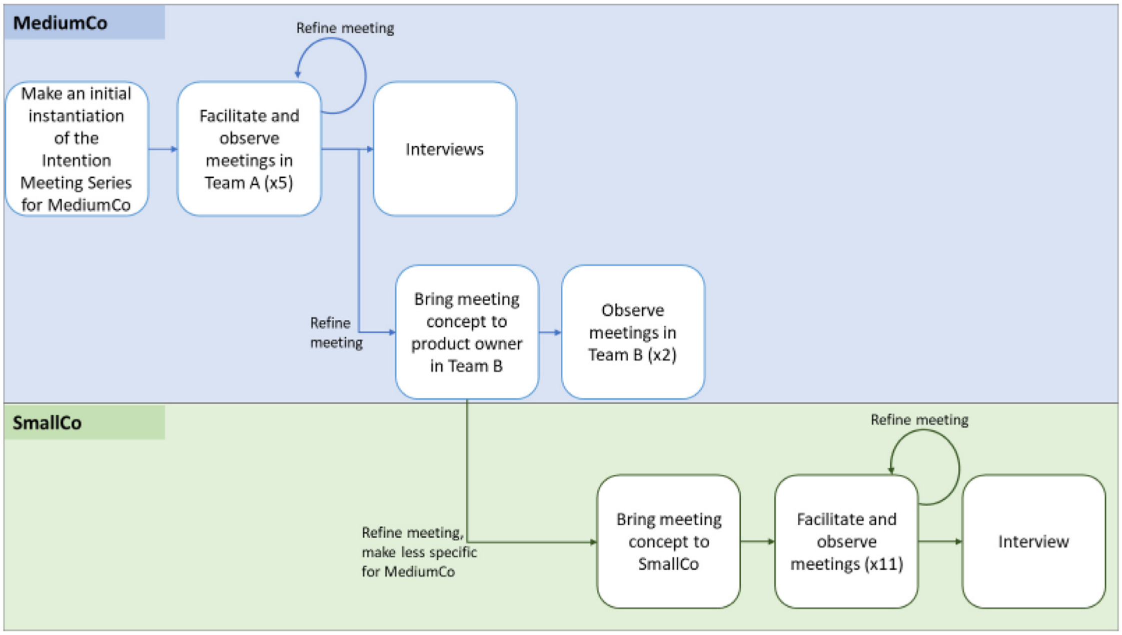

MediumCo
Make an initial instantiation of the Intention Meeting Series for MediumCo
Refine meeting
Facilitate and observe meetings in Team A (x5)
Interviews
Refine meeting
Bring meeting concept to product owner in Team B
Observe meetings in Team B (x2)
SmallCo
Refine meeting, make less specific for MediumCo
Bring meeting concept to SmallCo
Refine meeting
Facilitate and observe meetings (x11)
Interview

Fig. 4. Overview of the technical action research with MediumCo and SmallCo.

an opportunity to see how regular security prioritisation meetings could support companies with limited security resources and experiences, a situation we envision to be common in many SMEs.

After having iterated over the meeting approach with MediumCo, the meeting support material used was refined to make it less company specific. Then it was brought to SmallCo. Initially, we spent some time getting to know the company, as we had not worked with this company before. This took the form of meetings and a company visit. Then, we facilitated eleven meetings, followed by an interview with the main developer of SmallCo. The meetings had a similar structure as in MediumCo, but with only developers as participants. However, due to the limited size of this company these developers had roles also related to security, and they were involved in strategic discussions with managers in the company.

The number of meetings facilitated and observed was based on practical considerations as well as principles related to saturation. In MediumCo, we facilitated enough meetings in Team A to make the company confident that they could continue with meetings if they chose to. When meetings were brought to Team B, we observed all meetings that were performed. In SmallCo, we ended up achieving saturation in our observations, with no main new issues emerging in the last few meetings. More details on the meeting model and the support material used in MediumCo and SmallCo can be found in Appendix A and in Table 1.

Technical action research is different from other action research as it is artifact-driven, not problem driven (Wieringa, 2014). Still, it satisfies the principles of canonical action research (Davison et al., 2004; Wieringa and Moralı, 2012). Table 4 provides an overview of how the technical action research, as applied in this study, relates to these principles.

3.4. Observational case study: UnaidedCo

Case studies study phenomena in their real-world context (Runeson and Höst, 2009; Yin, 2018), and in this case required a company already performing some sort of regular security prioritisation meeting. Through our interaction with companies, we identified a company that had such a practice. This happened as part of work we were doing with this company to identify and evaluate their software security practices. At that point we were already doing technical action research with MediumCo, and we saw the opportunity to complement the knowledge gained from technical action research with a case study of a security meeting approach that were ongoing and led by the company itself.

The meeting approach had been developed by UnaidedCo independent of the ideas related to the security intention meeting series (Tøndel et al., 2019a). However, the meeting had many similarities with the original security intention meeting concept. Due to the organisation of the meeting at the department level, no product owners or similar were present (these were in a different department). Meeting participants consisted of the department manager as well as senior employees. Together this group had security competence, decision making authority, and knowledge of the development. There was less attention to the setting of intentions and following them up, as compared with the original idea. A typical meeting started with going through the status of previous action points. Then followed an open discussion on security concerns, with the aim to identify and note down any such concerns to inform security prioritisations. Then the participants decided on action points for the next period. The meeting happened monthly. More details on the meeting model and the support material used in MediumCo and SmallCo can be found in Appendix A and in Table 1.

Fig. 5 provides an overview of the case study with UnaidedCo. As already stated, it started with interviews aimed to identify and evaluate current software security practices. Then, we moved on to observing four security group meetings. These were facilitated by UnaidedCo, and the first author acted as an observer. Eventually, we arranged a retrospective that served as an opportunity for the company to discuss and improve upon their own meeting practice, as well as an opportunity to get feedback on initial findings from the observations. This way, we established a basic overview of the context in which the meetings took place, got deep knowledge on the meetings through observations, and got to know the participants' thoughts on the meetings and their effect.

The number of meetings observed (4) was agreed with the company beforehand. We however experienced that few new issues came up in the meeting observed last, indicating that we were moving towards saturation in the observations.

Table 4
Adherence to the principles of canonical action research (Davison et al., 2004).

Canonical action research principle	Technical action research as performed in this study
1 – the principle of the researcher–client agreement	The action research was part of a bigger research project where we had established NDAs with the companies. The companies agreed that this meeting could be a good approach for them given their situation, and they agreed to participate in data collection.
2 – the principle of the cyclical process model *(diagnosis – action planning – intervention – evaluation – reflection)*	For both MediumCo and SmallCo, there were initial activities to understand their needs and assess the relevance of the security meeting approach as an intervention *(diagnosis)*. Then, the meeting approach was instantiated for the company *(action planning)* before the meeting series started *(intervention)*. In relation to the meeting there were reflections with the participants on the meeting itself, and the researchers also made their assessment of how the meetings could be improved *(evaluation, reflection)*. Based on this, adjustments were made before the next meeting. Evaluation also happened in semi-structured interviews.
3 – the principle of theory	The action research was guided by the hypothesis that the artifact would be able to improve the security prioritisation in the companies. Previous knowledge on influences on security prioritisation were used to understand the effects of applying the artifact.
4 – the principle of change through action	Each cycle aimed to improve the security prioritisation of the company/project, as well as improve the security meeting approach.
5 – the principle of learning through reflection	The researchers and the company representatives reflected on the meetings, both as part of the meetings themselves and in interviews.

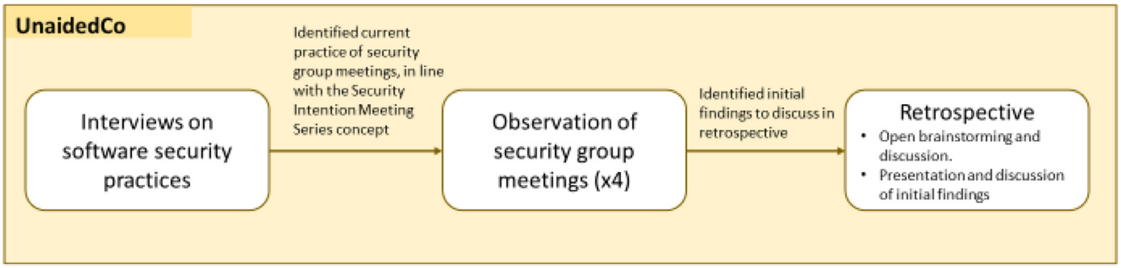

UnaidedCo
Interviews on software security practices
Identified current practice of security group meetings, in line with the Security Intention Meeting Series concept
Observation of security group meetings (x4)
Identified initial findings to discuss in retrospective
Retrospective
• Open brainstorming and discussion.
• Presentation and discussion of initial findings

Fig. 5. Overview of the observational case study with UnaidedCo.

3.5. Analysis

The analysis approach we applied allowed us to dig deep into each case individually, while allowing for the necessary overview to identify findings and learning points across cases. The analysis process consisted of two main stages, as depicted in Fig. 6. The first stage was concerned with analysing each case individually. This was important, as a main strength of case studies – and we would claim, also of action research – is to be able to dig deep into cases and take the wholeness of the case into account (Yin, 2018). Thus, cross-case synthesis, the second analysis stage, should be performed with the goal "to retain the integrity of the entire case and then to compare or synthesise any within-case patterns across the cases" (Yin, 2018). These recommendations informed the analysis process of this study. We used the same analysis strategy for each case, including the same coding structure. However, synthesis was done on the aggregated findings from each case. These aggregated findings were documented in longer memos. This approach was chosen to ensure we did not perform a simplified comparison on the variable level but rather compared and synthesised findings on the case level (Yin, 2018). To support cross-case comparison, the findings from each case were summarised in a table in an excel sheet, as visualised in Fig. 6. This is in line with recommendations from Miles et al. (2018) of using matrix displays to support cross-case analysis. This table provided an overview of the findings related to the effect identified from the meetings (RQ2), what was found to increase or reduce this effect (RQ1), and what contributed to or hindered adoption (RQ3). For each entry in the table, it was stated which case (marked M (MediumCo), S (SmallCo), or U (UnaidedCo) in Fig. 6) it was related to. The entries were sorted according to the relevant influence category. In Section 4 that presents the findings, Tables 7, 8, and 10 use the same format as was used for cross-case synthesis.

The process used for analysing individual cases is depicted in Fig. 7. All the collected data was imported into the qualitative data analysis software MAXQDA Pro 2020, and deductively coded within the coding structure shown in Table 5. According to (Maxwell, 2013), there are three main types of codes. *Organisational categories* represent areas you want to investigate. *Substantive categories* describe what happened or what was said. *Theoretical categories* place the data into a more general abstract framework. In this study, organisational categories were selected according to the research questions, as shown in Table 5. These categories were used to structure the data material, and initial coding used only these categories to organise the data. This could be considered indexing, in line with recommendations from Deterding and Waters (2021), and implied coding larger chunks of text into the organising categories. Then, for each of the organising categories, we performed analytical coding into substantive and theoretical categories. This is in line with recommendations from Deterding and Waters (2021) to focus on one research question at a time and to apply only a few analytic codes at the time to increase the reliability and validity of the coding.

As we were interested in identifying effects on the priority given to security (RQ1), we used previous knowledge on influences on security priority as theoretical codes (Tøndel et al., 2022). Thus, our coding approach was deductive. Our deductive approach was however not motivated by theory-testing (Wohlin

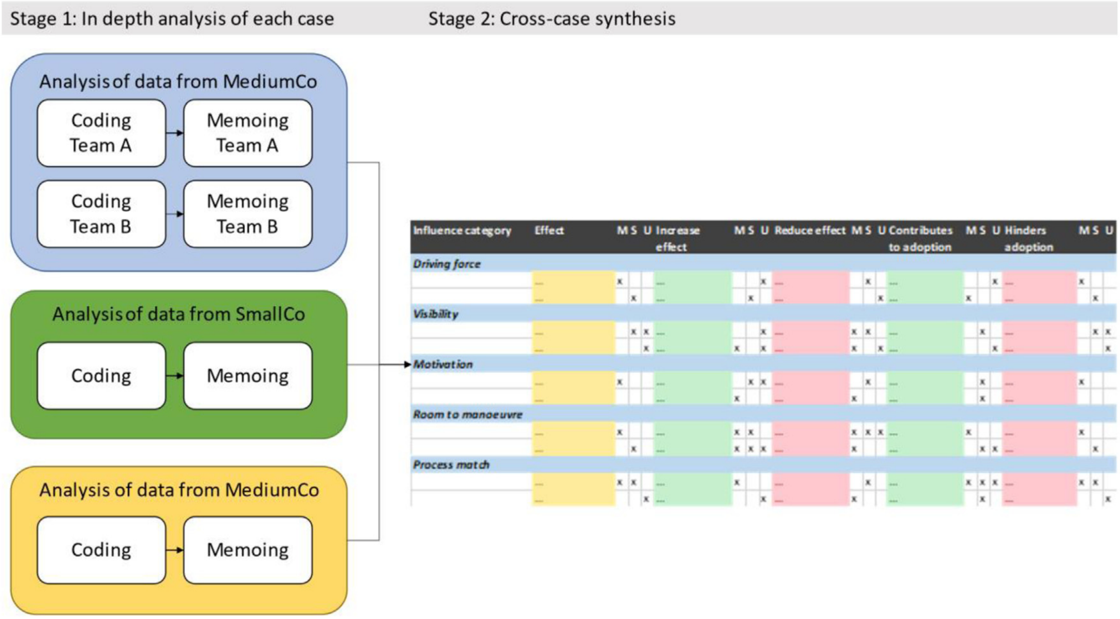


Fig. 6. Overview of the analysis process.

Table 5
Overview of coding structure.

Organisational category – area	Organisational category – subarea	Theoretical codes	Relation to RQs
Adoption	Examples of adoption	NA	RQ3
	Examples of non-adoption	NA	
	Reasons for adoption	Driving force; Visibility; Motivation; Room to manouvre; Process match	
	Challenges to adoption	Driving force; Visibility; Motivation; Room to manouvre; Process match	
Effects	Positive effects	Driving force; Visibility; Motivation; Room to manouvre; Process match	RQ2
	Contributes to effect – Context-related – Meeting-related – Task-related	Driving force; Visibility; Motivation; Room to manouvre; Process match	RQ1
	Hinders effect – Context-related – Meeting-related – Task-related	Driving force; Visibility; Motivation; Room to manouvre; Process match	RQ1
Challenges and improvement suggestions	Challenges in the meeting	NA	RQ1
	Challenges in the context	NA	
Worked well	Worked well in the meeting	NA	RQ1
	Worked well in the context	NA	

and Aurum, 2015). Rather, it was motivated by a need to approach and understand this rather broad and abstract concept (the priority given to security) in a systematic way. For definitions of the influence categories, see Table 6. As can be seen from the overview of the coding structure in Table 5, we used these theoretical categories in the coding related to adoption and effect, based on the following considerations:

- Effect (RQ2): We had a special interest in identifying and understanding effects related to security prioritisation, and these influence categories could help identify effects likely to have an impact on the prioritisation given to security
- Aspects that contributed to or hindered effects (RQ1): The five categories structure influences on the priority given to security, and thus could also help identify and structure influences on the effect of these meetings.
- Reasons and challenges for adoption (RQ3): We hypothesised that these influence categories could also help identify and structure conditions that influence adoption of the meetings, as adoption of these meetings can be considered part of giving security priority.

As was presented in Section 2 and summarised in Fig. 1, there are many potential effects and recommended strategies that support the suitability of regular security meetings to achieve prioritisations and decisions regarding security. The influence areas we decided to use as theoretical categories in the coding are part of this foundation. Still, it was important to ensure that we, by deciding to build on these influence areas, did not exclude

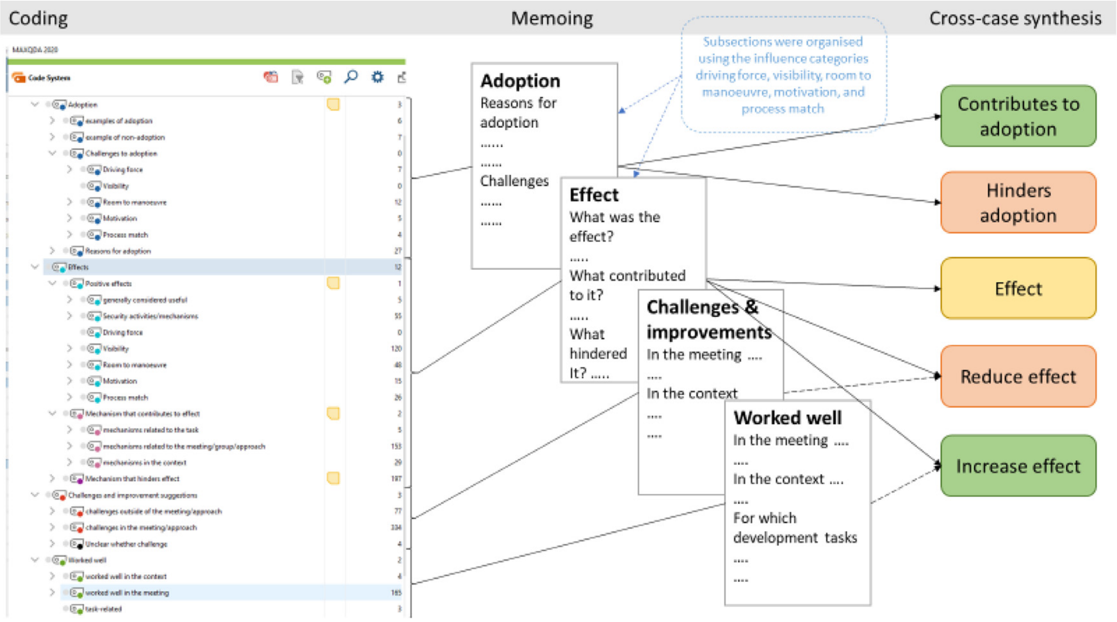


Fig. 7. Strategy for coding, memoing, and contributing to cross-case synthesis.

Table 6
Influence areas from Tøndel et al. (2022).

Influence area	Definition from Tøndel et al. (2022)
Driving force	Someone who takes initiative and responsibility for making software security happen. A negative driving force would actively hinder software security.
Visibility	The degree to which security is visible (seen, known about) to stakeholders related to the project. This includes the visibility of security to developers in their daily coding activities, to project management and top management, to the customer, and in the product.
Motivation	The willingness to focus on software security, as well as the aspects that cause such willingness. Reasons for doing or not doing software security, and activities that provide such reason would be part of this category.
Room to manoeuvre	Resources and opportunities to prioritise software security, and to act accordingly. This might include time, budget, competence, etc.
Process match	The ability to fit the security approach into the existing software development process, so that they align well.

other aspects that could be important as well. We thus performed a mapping of the effects and recommended strategies identified from the broader set of literature (and as depicted in Fig. 1) with the influence areas from Tøndel et al. (2022). This mapping is shown in Fig. 8. Some of the relations are quite clear, like the relation between the effect "Visible and tangible security" and the influence area "Visibility". Other relations were more subtle. Examples are the effect "Ongoing priority of security" and the recommended strategy "Context supportive of emerging security practices". These we consider covered by all the influence areas in combination. Considering each influence area, *driving force* includes effects and strategies related to stakeholders as these can be important driving forces (or the opposite) for security prioritisation. *Visibility* includes making security more tangible, e.g., through prioritisation and documentation. *Motivation* includes getting towards an ongoing priority of security. *Room to manoeuvre* includes aspects related to reflection and learning, as this supports security knowledge and awareness. *Process match* concerns how the process for security prioritisation and follow up is organised, including who is involved.

After coding, we wrote four longer memos per case (as shown in Fig. 7), identifying and describing the findings related to the topic of the memo: adoption, effect, challenges and improvements, and worked well. Fig. 9 provides an example from this process. The memos offered an opportunity to summarise the key findings and reflect on them. The key findings were then again used as input to the cross-case table. As is shown in Fig. 7, the memos on adoption and effects were organised using the influence categories, and were used as the main input for the cross-case table. The memos on challenges and improvements and on what worked well were used to complement the findings.

4. Findings

In the following we present the findings, organised according to the research questions. We start by describing the effects of the meetings (RQ2). Then we move on to presenting lessons learned on what contributed to or hindered the effect (RQ1). Finally, we describe findings related to adoption (RQ3).

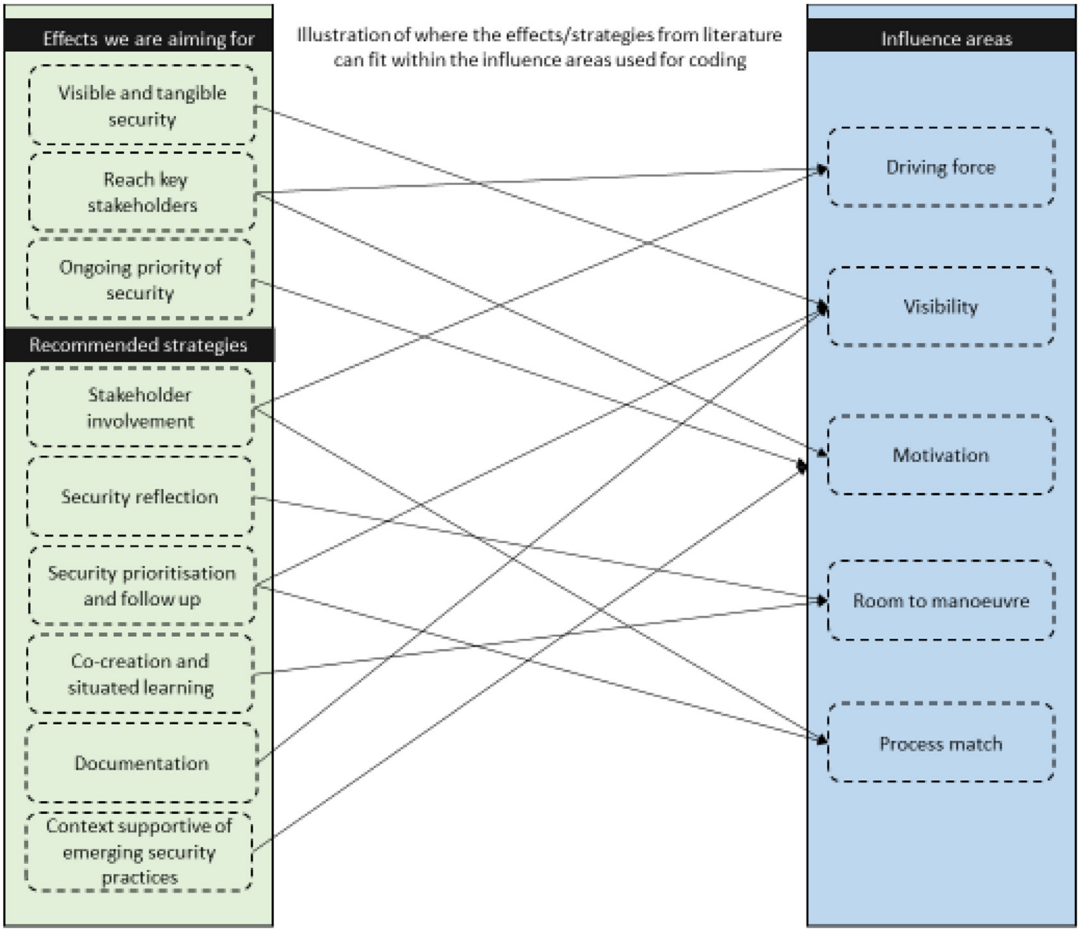
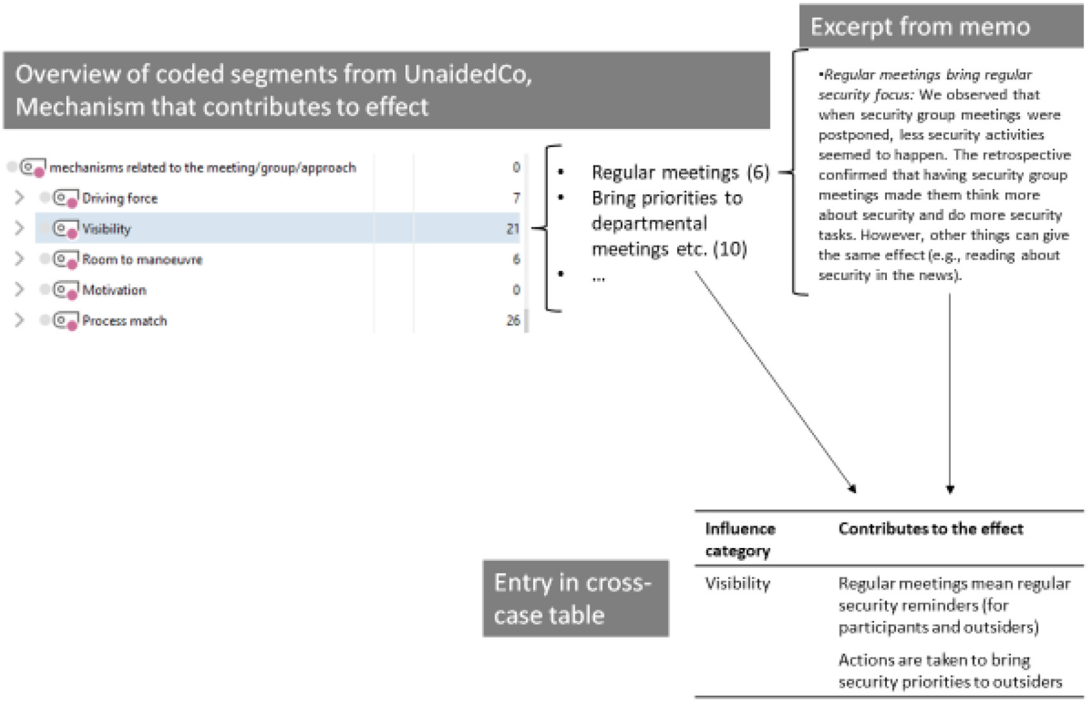

Effects we are aiming for
Visible and tangible security
Reach key stakeholders
Ongoing priority of security
Recommended strategies
Stakeholder involvement
Security reflection
Security prioritisation and follow up
Co-creation and situated learning
Documentation
Context supportive of emerging security practices
Illustration of where the effects/strategies from literature can fit within the influence areas used for coding
Influence areas
Driving force
Visibility
Motivation
Room to manoeuvre
Process match

Fig. 8. The effects and strategies identified in literature (ref. the overview given in Fig. 1) can be covered by deductive coding based on the influence categories from Tøndel et al. (2022). Note that the effect 'ongoing priority of security' and the strategy 'context supportive of emerging security practices' are covered by all influence categories in combination.


Overview of coded segments from UnaidedCo, Mechanism that contributes to effect
Excerpt from memo
Entry in cross-case table

mechanisms related to the meeting/group/approach	0
Driving force	7
Visibility	21
Room to manoeuvre	6
Motivation	0
Process match	26

- Regular meetings (6)
- Bring priorities to departmental meetings etc. (10)
- ...

•*Regular meetings bring regular security focus:* We observed that when security group meetings were postponed, less security activities seemed to happen. The retrospective confirmed that having security group meetings made them think more about security and do more security tasks. However, other things can give the same effect (e.g., reading about security in the news).

Influence category	Contributes to the effect
Visibility	Regular meetings mean regular security reminders (for participants and outsiders) Actions are taken to bring security priorities to outsiders

Fig. 9. Example illustrating the link between the codes, the memos, and the cross-case table.

Table 7
Effects on the influence categories on security priority (M = MediumCo, S = SmallCo, U = UnaidedCo).

Influence category	Effect	M	S	U
Driving force	Enabling developers to take on responsibility and initiative for security in their work (including further security meetings)		X	X
Visibility	Identify, clarify, and document security needs, issues, and tasks	X	X	X
	Visibility of the security tasks they are doing, and the security they currently have in place		X	X
	Uncover non-functioning security roles/tasks	X		
Motivation	Positive view of the security work and meetings in the department/company	X	X	X
Room to manoeuvre	Awareness and knowledge building on security	X	X	X
	Reuse of knowledge and security work across projects, opening for reduced cost of security	X	X	X
	Increased confidence on security and on the decisions made, and opening for getting support for decisions from colleagues		X	X
Process match	A way to start and continue working with security, including ideas for how to modify and adapt the approach to better match their needs	X	X	

4.1. Effects from the meetings

The meetings brought many positive effects. All meetings led to the creation of security documentation. SmallCo and UnaidedCo reported on direct effects in their development; examples being a merge request template, security workshops, developing an incident response management process, starting to use a password manager, implementation of solutions for signatures and authentication, improved solution for remote support, the establishment of additional security meetings, and fixing of identified weaknesses. As shown in Table 7, the meetings also brought effects related to all the influence categories previously found to affect the priority given to security.

Looking at the influence categories, the main effects came within 'room to manoeuvre' and 'visibility'. In all three companies, the meetings helped build security competence and awareness among participants. The discussions brought both general and specific security competence relevant for the software being developed, and note-taking made the identified security needs, issues, and tasks visible also longer term. There is even evidence that this awareness, competence, and visibility spread to individuals who did not participate in the meetings (in the following termed 'outsiders'). As meeting participants gained more competence and confidence on security, they improved their ability to act as a driving force for security. Note, however, that the meetings did not necessarily give more time for security, although there is some evidence that they made it easier to ask for time to do security tasks (SmallCo).

The meetings helped get an overview of current security work and uncover potentials to improve this work, thus contributing to more cost-effective security. Improvements identified concerned reuse across projects, addressing non-functional security roles/tasks, and improving the security meetings. The meetings offered one way to get started with and continue working with security.

4.2. Lessons learned on meeting organisation

Table 8 gives an overview of identified influences on the effect of the meeting. In the following we describe lessons learned from all the cases when it comes to effectively organising these meetings. We start with describing the similarities identified across cases. Then, we bring up the main variations among the cases. Finally, we delve into one overarching issue emerging from the analysis: confidence in the software security prioritisations and their effect.

4.2.1. Similarities across the cases

In all three cases, the following aspects of the meeting were important *contributors* to the effect:

- A view in the company of security as important and worthwhile (motivation)
- Regular security meetings as regular security reminders towards both participants and outsiders (visibility)
- Participants positioned to take action and bring a security mindset to outsiders (driving force)
- Participants being positive and engaged towards the meeting and security (motivation), having security competence and experience (room to manoeuvre)
- Good discussions in the meetings to build awareness and competence (room to manoeuvre)
- Concrete action points from the meeting (process match)

These contributors could, e.g., play out as follows. UnaidedCo had for several years experienced an ongoing push for security and this ensured an opening to spend time and money on security (including gathering senior people for a security meeting). In UnaidedCo, participants explained that the meetings helped them think more about security and made them do more security tasks, and we observed that when the security meetings were postponed less security activities seemed to happen. In MediumCo, personal engagement motivated a product owner to take on responsibility for security tasks despite strong time pressure. Participants in UnaidedCo stated that the discussions were the most important part of the meeting. In MediumCo, one of the product owners held clear action points as the most useful result of the meetings.

All the studied cases experienced the following *challenges* in getting effect from the meeting:

- Important discussion points risked being lost as not all points seemed to be noted down (visibility)
- A focus on functionality in the company and among customers pushed security to the background (motivation)
- Time pressure made it hard to take on responsibility for and/or perform the action points, or made it hard to set aside time for necessary security training (room to manoeuvre)

In all meetings, one person took responsibility for taking notes, and usually these notes were made visible to all participants. Still, we observed that there was a risk of forgetting to write

Table 8
Overview of influences on the effect of the meeting (M = MediumCo, S = SmallCo, U = UnaidedCo). Note that the term "outsiders" is here used to refer to company employees not participating in the meeting.

Influence category	Contributes to the effect	M	S	U	Hinders the effect	M	S	U
Driving force	Participants that are positioned to take action and bring a security mindset to outsiders	X	X	X				
	Facilitator that pushes for documentation and follow up of action points	X		X				
	Skilled security expert as facilitator and contributor to the meeting	X	X					
Visibility	Regular meetings mean regular security reminders (for participants and outsiders)	X	X	X	Important discussion points can be lost as not all points seem to be noted down	X	X	X
	Template gives visibility to topics for discussion and support in identifying and documenting issues	X	X		Outsiders may not read security documentation although it is made available to them		X	X
	Template brings attention to similarities and differences among projects	X			Outsiders do not feel they get enough information from the meeting			X
	Actions are taken to bring security priorities to outsiders		X	X	Visible costs of security while effects of security work are less clear			X
	Security issues visible to outsiders are easier prioritised			X				
Motivation	Participants that are engaged and positive towards the meeting and security	X	X	X	Focus on functionality in the company and among customers	X	X	X
	A view in the company of security as important and worthwhile	X	X	X	Too many prioritised tasks, long list of issues to address and consider			X
	External pushes for security, e.g., pentest, customers			X	Security tasks that are boring or unpleasant			X
Room to manoeuvre	Meeting participants with security competence and experience	X	X	X	Lack of trust in own judgement — analysis paralysis		X	
	Good security discussions in the meeting build awareness and competence	X	X	X	Challenges in understanding security terms in the template		X	
	Template (security areas, questions) that support discussion and knowledge building		X		Knowledge needs related to practical security solutions make it hard to translate decisions into code, etc.		X	
	Initiatives to bring security competence to the company Room to spend time on security		X	X X	Time pressure makes it hard to take on responsibility for and/or perform the action points	X		X
	Ability to identify a wide variety of issues quickly in the meeting	X			Tasks that are difficult, large, or concern old systems			X
			X		Hard to set aside time for training		X	
	Having concrete things to discuss helps justify the time spent in the meeting				Lack of roles such as sysadmin to establish security infrastructure, etc.		X	
Process match	Concrete action points from the meeting	X	X	X	Not well placed to deal with non-project related issues	X	X	
	Ability to include results from meeting in external planning process			X	Participants that are not the right ones to be responsible for an action point	X		X
	Small company, low security maturity, easy to get effects		X		Need perspectives from outsiders	X		
					Lack process for following up meeting results in development	X		
					Meeting too early or too late during development	X	X	

down discussion points, and that it could be challenging to know what to document and how. Unsurprisingly, all studied companies experienced a push for functionality that impacted the priority of security. UnaidedCo explained that the product owners were mainly concerned with functionality. Similarly, MediumCo explained that customers pay for features, not security. Consequently, it was hard to push for security when the security tasks could delay development of features. This push for functionality was somewhat related to time pressure, which was particularly strong in MediumCo. Their most stated reason for skipping security tasks was time pressure, and all roles experienced such pressure. One product owner thus explained that the security meetings mainly served to give him bad conscience, as it made him aware of all the things he did not manage to do:

> *"Product owner: It kind of works that you are part of discussions and contribute with what you know, one hour and every month. However, it does not happen that much in-between.*

Interviewee: But it does not have an effect that you are reminded of it every month?

Product owner: Yes, reminds me of it and gets a bad conscience for everything that should be in Jira, and tasks related to that".

4.2.2. Variation: participants

The characteristics of the meeting participants varied across the cases, as shown in Table 9. The cases had different needs to be met by the meetings. This can explain the variations, as all have their benefits and challenges. In the following we point to lessons learned when it comes to meeting participants.

It was important to have security competence in the meeting (room to manoeuvre), but it was not necessary to *only* have participants with security competence. To illustrate, in SmallCo the security competence was held by the facilitator who could explain terms, point to potential challenges, question assumptions, and point towards solutions. Further, the need for security competence did not mean there had to be a security expert in the meeting; many of the participants with developer, architect, or product owner roles had sufficient competence to identify and discuss security issues and mitigations. We, however, observed variations in depth and speed of discussion that may be related to security expertise and driving force. The clearest example of valuing efficiency over depth was found when bringing the meeting to a new team at MediumCo. Here the product owner leading the meeting managed to go through the full template, including all the 13 security areas, in 75 min. The previous meetings in MediumCo that were facilitated by a security expert, had spent considerably more time on each security area, digging deeper and asking hard questions on assumptions.

The meeting needed participants positioned to take responsibility for the tasks prioritised in the meeting (process match, room to manoeuvre). In UnaidedCo, meeting participants were given responsibility for security action points, while other tasks were normally the responsibility of team leads. Thus, the security tasks were not fully integrated in their process. However, assigning responsibility for security tasks to the team leads was challenging as they were not participants in the security meetings, and thus not present to report on the status of the action points. For MediumCo, meeting participants lacked the capacity to take on responsibility for more tasks due to time pressure. Thus, there were discussions on whether to add participants who were better positioned to do the necessary work on the action points (e.g., a security champion or developer). Further, many of the meeting discussions in MediumCo covered topics that involved operations or other development teams. Thus, participants lacked knowledge to make realistic assumptions about the risk, and many of the issues and action points identified were concerned with gathering more information.

4.2.3. Variation: meeting scope

The meetings we studied either had a department scope or a project and team scope, and both scopes had their benefits and challenges (primarily related to process match). Regarding benefits, the department-scope of UnaidedCo made their meetings well placed to make decisions that affected the whole department and not only one project or team. Several department level initiatives stemmed from these meetings, including a merge request template and hacker workshops. The project-scoped meetings of MediumCo and SmallCo were able to dig deeper into the individual projects, but also cross-cutting security concerns were discussed. Although the department-level meetings were well positioned to support learning across projects and technology, such effects were also seen in MediumCo where cross-team learning happened through participants being involved in security work in several projects. And as stated by the developer in SmallCo: *"We have talked about one project, but I have always kept in mind all the other projects".*

Challenges were more prominent in the project-scoped meetings. Both scopes experienced issues falling between two stools; security issues could concern another development team (MediumCo), operations (MediumCo), or be too big to address within current plans (UnaidedCo). However, project-scoped meetings had more challenges in acting on cross-cutting concerns. Both scopes experienced challenges related to keeping a lifecycle perspective; also UnaidedCo found it harder to make security happen in existing vs. new systems. However, a product owner at MediumCo advocated for a product scope rather than a project scope in the meetings, as software changes were made also for products not in active development in a project; *"A customer comes and wants to pay 50 000 NOK to add a button, and then we add that button. This does not become a project, and then there is no security decisions meeting (…). We make a change; we spend two weeks and make a change and that's it".*

Challenges related to meeting scheduling were specific for project-level meetings. When scheduling meetings too early in the project, there was not enough information to make security prioritisations and decisions. Changes to projects were normal, especially in the beginning, thus there were considerations on how long to wait for things to settle. Further, there were concerns about when to revisit previous assumptions. When starting the meetings too late, the option to influence the project plans and estimations were largely lost. We observed a risk that the meeting could be experienced as a security kick-off for a project, without this leading to regular security meetings as was the intention.

4.2.4. Variation: support material

Both SmallCo and MediumCo used support material in form of security areas and security questions in their meetings (see Appendix A). Observations pointed to a potential effect in both companies, related to triggering ideas for discussion, aiding in documentation, supporting awareness and knowledge building, and building confidence in the assessments made (visibility, room to manoeuvre). The material supported the experienced facilitator, and even allowed the product owner in the new team at MediumCo to run the meeting after being introduced to the material. In SmallCo, it offered a way to get started with the big topic of security, by breaking security into more manageable pieces, and it brought visibility to topics that had not been much considered in their solutions before; *"The biggest advantage is that you know a bit more, it is more structured what to talk about. It is easier to remember what we have talked about, and what we have not talked about"* (developer SmallCo). These effects came despite both companies identifying many potential improvements to the material. Note, however, that UnaidedCo did not use such support and still covered a broad set of security aspects in the discussions, produced organised security documentation, and built competence and awareness on security.

4.2.5. Variation: company's size and security maturity

Our study included a very small company just starting to work with software security (SmallCo), as well as medium-sized companies that already had experience with software security (MediumCo and UnaidedCo). SmallCo experienced challenges related to room to manoeuvre, while MediumCo and UnaidedCo experienced challenges related to process match and visibility.

Competence was important in all cases, but as a small company with few developers and technical resources, SmallCo experienced that a lack of practical security skills was a hindrance for implementing security measures. Further, as they lacked roles such as sysadmin, developers had to take broader responsibility for practical tasks. In, e.g., MediumCo, such roles were filled,

Table 9
Observed variations that can be related to the selection of participants.

Case	Participants	Benefits	Challenges	Main effect
MediumCo — initial project	Small set of participants with decision making power and competence. Facilitation by security expert.	Good discussions, confidence.	Individuals highly pressured for time, lacked involvement in all parts of the project.	Addressed the need to identify security issues not explicit in customer requirements and get towards solving the security issues.
MediumCo — new team	Full team. Facilitation by product owner.	Broad awareness of security issues.	Efficiency over depth in discussions. Did not redo the meeting.	Identification and documentation of issues.
SmallCo	Two developers. Facilitation by security expert.	Developers that could bring improved security focus.	Need additional meetings to bring broader changes.	Got started, built competence, established new practices.
UnaidedCo	Seniors. Facilitation by manager.	Reach key actors in the department, positioned to make changes.	Get broader effects, reach beyond the participants, get tasks prioritised by team leaders.	Identified, documented, and addressed issues. Support for participants in their ongoing attention to security.

and more competence was available, but the challenges were related to company silos and security concerns being viewed as part of someone else's responsibility. To exemplify, the new team applying the meeting found that for many issues they were dependent on third parties, other teams, or operations. However, these issues were generally skipped in the discussions, thus no action was made to ensure that they were in fact addressed.

Integration into the larger development process was challenging for the medium-sized companies. In MediumCo, the product owners were used to doing refinement in Jira. There was, however, no surrounding security process that ensured action points from the meetings were followed up on, e.g., by adding them to Jira. In UnaidedCo, security tasks were generally not included as user stories and added to Target Process and their Kanban. Thus, in both companies there was a need to remember to look at a separate list/page for security tasks. Adding the security tasks into Jira or TargetProcess was, however, not without challenges. In MediumCo, Jira was explained as already being filled with too much information, making it hard to navigate. It was thus easier to get an overview of all security decisions in a Confluence page. In UnaidedCo, adding all security concerns to Target Process was not an option, as the information was considered too sensitive to store in a Cloud solution. Note that, on a related point, UnaidedCo successfully included larger security tasks in yearly plans for the department, showing that integration with the planning process already in place could be effective.

Both SmallCo and UnaidedCo took action to spread information from the meetings to outsiders. In SmallCo it was easy for the participants to discuss the meetings informally with other employees, including management. Still, they started a new security meeting series with management. UnaidedCo spread information from the meetings in weekly department-level status meetings and in emails — with meetings being most effective. Still, outsiders expressed that they wanted more information from the security meetings, and roles outside the department (such as product owners) most likely did not know about these meetings and the prioritisations made there. Broad sharing of information was however challenging, as much of the security documentation created in the meetings was considered highly sensitive. Further, making security documentation available to a broader set of individuals did not imply that this documentation was read and understood by others.

4.2.6. Variation: maturity of meeting series

Challenges to the meetings varied depending on whether the meeting series was just starting, or whether it had been going on for some time. Initial challenges included understanding how much time to set aside, and clearly communicating the goal and structure of the meeting (MediumCo). Further on, challenges included how to proceed when all security areas had been discussed, and when to revisit assumptions (SmallCo). After meetings had been going on for a long time, the number of issues identified but not yet addressed could be challenging to manage (UnaidedCo) — as stated in the retrospective of UnaidedCo: *if you should go through the to-do list, this is the whole meeting.*

4.2.7. Overarching issue: confidence in the software security prioritisation and follow-up

Experiences from UnaidedCo showed that prioritisation and concretisation of tasks were important prerequisites for action. Security discussions took place also before they started with this meeting series, but those discussions usually did not lead to actions, as there was no clear process for following up on the issues. Due to the meeting series, all these issues were documented, and they ended up with a long list of security issues. To start addressing the issues, prioritisation became important. But prioritisation was also challenging. The retrospective showed that often more action points were prioritised than what was realistic to address before the next meeting, leading to erosion of responsibility.

Challenges related to prioritisation were found within all influence categories. In all the studied cases it was difficult for the observer to understand why some issues were prioritised over other issues, indicating unclear prioritisation criteria. The developer from SmallCo talked about the risk of analysis paralysis, especially if participants lacked confidence in own ability to make good security decisions. For UnaidedCo, it was challenging to manage the long list of concerns identified over the course of all meetings, and thus it seemed easier to prioritise newly identified concerns. Furthermore, tasks that were boring (e.g., fixing an existing system) or unpleasant (e.g., could cause down-time or required work outside of normal working hours) were less likely to be prioritised.

Prioritisation also happened outside of meetings, and in relation to the totality of the tasks that the participants were expected to address. This prioritisation (security vs. features) was described as more challenging than the prioritisation among security tasks happening in the meetings. In the meetings, we observed that when action points had not been addressed, there was often little discussion as to why this was the case. Thus one missed the opportunity to learn about barriers to security work and improve how action points were addressed in the future. All observed meetings benefited from an open and non-judgemental tone where participants were willing to share knowledge needs and insecurities. However, the need to hold participants account-able for following up on their responsibilities for action points was slightly neglected.

4.3. Conditions leading to adoption

An overview of what contributed to or hindered the adoption of the meeting, both shorter and longer term, is given in Table 10. In the following we describe mainly what we found to be important for longer-term adoption. The meetings that were adopted longer term were the monthly security group meetings at UnaidedCo, that had been going on for some time before our study, and the monthly management meetings on security, that were started by SmallCo while we did our study with them. Further, SmallCo included security in daily meetings among developers. The meetings we initiated in SmallCo and MediumCo were not continued.

The meetings that were adopted longer-term shared some characteristics. Though both SmallCo and UnaidedCo were triggered by external security experts in initiating their meetings, the meeting approach they applied had been created by the company itself, and was driven by key individuals from development. Both companies applied cross-project meetings. Moreover, management in both companies acknowledged the importance of spending time on security, and the time needed for the meetings was perceived as acceptable.

Offering a good process match was supportive of adoption, although not enough to ensure or hinder adoption (process match could be achieved over time). It was considered beneficial to have an easy approach that could be done efficiently, and that could be adapted to the needs of the company. On the other hand, it was challenging with security meetings that were somewhat "on the side" of their development process, and thus had to be remembered and prioritised over "real development tasks", with each development project needing to consider when to start with security meetings.

In both cases where we brought in the support material and helped facilitate the meetings, these ended up not being adopted. The reasons put out were limited need for such a thorough approach in the new projects they had currently initiated (SmallCo), and time pressure of product owners in combination with no central security officer role that continued to push for the meetings (MediumCo). These are all aspects of the context. Thus, we cannot conclude that aspects of the support material or the agenda of these meetings prevented adoption. On the contrary, SmallCo expressed an intention to continue using this support material; *that excel sheet was really good, so we need to remember to use that!* (statement from observation notes).

5. Discussion

We started this article with introducing the concept of continuous software security. This concept implies that software security is treated as a key concern throughout the software's lifecycle (Fitzgerald and Stol, 2014). For this to happen, literature points to the need to reach key stakeholders with software security, to make security more visible and tangible, and to prioritise security in an ongoing manner (Fig. 1). In this section we relate our findings to the literature, and identify implications for research and practice. We organise the discussion according to the topics we identified from literature in Fig. 1, and we use **bold** whenever we refer to topics in that figure.

The meetings contributed towards **ongoing priority of security** directly through the activities that happened in the meeting, and through positive effects within all the influence categories related to security priority (Table 7). This included contributions towards **visible and tangible security**. The effects observed resemble those found for related techniques (Fig. 2) in that the meetings led to concrete security improvements, and the strongest effects were related to visibility, competence, and awareness of security. Thus, we claim that such effects can be expected from security meetings in general. However, literature points to stronger effects of security workshops in smaller companies, compared to larger and more mature ones (Weir et al., 2021). In the study by Weir et al. (2021), this finding may, however, be due to organisational turmoil in one of the large companies they studied, rather than their approach (Weir et al., 2020a). We found that the meetings were effective in all cases, but that it was easier to get effects in the smaller and less mature company (SmallCo).

Implication for practice:

(P1) Regular security meetings are recommended for small and medium sized development companies that need to strengthen their software security maturity.

Implication for research

(R1) As we found that regular security meetings can be effective in smaller companies, further research should study what role (if any) regular security meetings can play in larger and more mature organisations, and how they should be organised in these contexts to support adoption and be effective.

The meetings' ability to **reach key stakeholders** was related to who was participating in the meeting, although we experienced that the effects of the meeting could reach beyond participants. Relevant strategies when deciding on participants are **stakeholder involvement** and **co-creation and situated learning**, but these need to be balanced towards the concern that larger meetings tend to be less effective (Stray et al., 2016). The variations among the cases concerning meeting participants, all had their pros and cons (Table 9), and it appears that there is no one-size-fits-all in this respect, as different participants may serve diverse needs (Weir et al., 2020a). Recommendations on whether to include the full team, a security expert, managers, and product owners come up in literature (see Fig. 2). Experiences from our study relate to these recommendations in the following way:

- Literature highlights the importance of involving the full team and not only seniors (see Fig. 2). Both MediumCo and UnaidedCo violated this recommendation, and still found the meetings effective. However, we found that the meeting needed some participants with room to take on responsibility for security tasks, and seniors may experience more time-pressure hindering them to take on this responsibility.
- Literature conflicts on whether meetings need a security expert (Weir et al., 2020a; Bernsmed et al., 2022). We found involvement of a security expert to be beneficial, and probably necessary when initial security competence was low (as in SmallCo). However, a security expert was unnecessary when at least some participants were aware and knowledgeable about security.
- Previous findings that facilitation by management is beneficial (Weir et al., 2021) is somewhat supported (UnaidedCo).

- The importance of product owner participation is unclear in literature and in our study. Weir et al. (2021) found surprisingly few effects of involving the product owner. In our study, product owners were involved in MediumCo but not in UnaidedCo. Both report on challenges that functionality is prioritised over security, and one of the product owners of MediumCo expressed that the meeting mainly led to bad conscience, not improved security.

Implication for practice:
(P2) When selecting participants, consider what are the main needs for your team/project/company. If you are in dire need of security competence, consider involving a security expert with the full team, alternatively a security expert with individuals interested in security, who can spread security competence and awareness to their team (e.g., security champions). If you need decision making power and competence, consider involving senior people. If communication and overview is a main need, consider involving participants across silos (e.g., both from dev and ops).
(P3) Include participants who can take responsibility for security tasks in development.
(P4) If possible, have a manager as facilitator of the meetings.

Implication for research:
(R2) As our results suggest that there is no one-size-fits-all regarding meeting participants, companies need guidance on which participants to include for varying meeting aims and contexts.

Weir et al. (2020a) recommended “the promotion of software development security as a systemic, rather than purely a development team, matter”. Literature shows that awareness of security weaknesses is not enough to induce change (Palombo et al., 2020). Clearly, a meeting alone cannot create a **context supportive of emerging security practices.** It does not replace the need to address the more structural and systemic blockers that hinder developers and product owners in prioritising security in practice, but it can support identification of these blockers. For this, product owner participation appears important but not necessary; discussions of systemic blockers and conflicting demands took place in all cases. However, these were often hard to address in practice and they could easily be viewed out of scope for the meeting, especially if the meeting had a project scope.

Implication for practice
(P5) Product owners can have an important role to play in the meetings, but beware that meeting participation will not necessarily change the priority they give to security if systemic blockers are not addressed.
(P6) Discussing reasons for not doing security work (without assigning blame), is one potential way to get more insight into the blockers that are present.
(P7) If there is a strong need for overarching changes, a department-level meeting may be called for.

Implication for research
(R3) Systemic blockers for software security came up in meeting discussions, indicating that security meetings can address such blockers. Still, more knowledge is needed on how to position and structure security meetings to best address systemic concerns, including how to document and follow up on such concerns within and beyond the meetings.

Literature advocates for emerging security practices (in contrast to prescribed practices) (Türpe and Poller, 2017; Weir et al., 2020a). Our study support this; the practices that are adopted longer term are those that emerged in the companies. Further, we hypothesise that emerging practices can be better positioned to deal with challenges in the context that may hinder security work, such as time-pressure — a prominent challenge in literature (Fig. 2) and in our study. The studied companies were able to select a meeting schedule that suited their need, and thus time for the meeting seemed not to be a main issue, although meetings were sometimes postponed. To ensure continuous adoption, we would highlight the importance of having a strong driving force for the meeting, someone with the authority to invite the right participants, facilitate the meeting, and ensure meetings are arranged regularly. We identified a potential need to change the meeting as a project progresses (SmallCo) and as the number of identified concerns increases (UnaidedCo). Although findings suggest that a good process match is supportive of adoption, this can be achieved over time. An engaged facilitator can take responsibility for making strategic decisions on how to improve the meeting, and adjust to changes in the needs of a project and/or company.

Implication for practice:
(P8) Adopt your own meeting approach that suits your needs (including your level of time-pressure), rather than copying an approach from others. This does not exclude learning from others’ experiences.
(P9) Ensure that someone is championing the meeting, and that this person has the necessary authority and motivation to make the meeting happen regularly and make improvements.

Implication for research:
(R4) As results indicate that meeting effectiveness can change with time, companies can benefit from guidance on how to ensure meeting adoption and efficiency in varying stages (e.g., as a project progresses or as the company matures).

The scope of the studied meetings varied, with some taking a project-scope and others a department-scope. We cannot claim that one is always better than the other, although some benefits were identified with a department scope (stronger ability to address more overarching concerns and cover products not in active development). We also saw a potential challenge with project-scope meetings in that each project must remember to initiate its meeting series.

Implication for practice:
(P10) If you go for a project-specific security meeting, ensure that the meeting becomes part of routines so that the meeting is remembered for all projects. Further, consider whether there is a need for meetings also for key products not under active development.

Implication for research:
(R5) Our results slightly favour department-scoped meetings, and we speculate that this is related to us studying smaller companies where such broader-scoped meetings may be more feasible than in larger companies. More knowledge is needed to determine what meeting scope is most effective in varying contexts.

In the meetings, **t**ime for **security reflection** and discussion was highly important for the effect of the meetings. This is in line with experiences from similar techniques (Fig. 2). The effect was however linked to **documentation;** there was a need to ensure key discussion points were documented (and not lost Cruzes et al., 2018; Tøndel et al., 2019b), and visible documentation supported the discussions (Stray et al., 2016). This study is not conclusive on how to structure the discussions and the documentation. Meetings were successful both with and without support material. Support material in form of a checklist has been suggested as an improvement to the security workshops studied by Weir et al. (2020a). The main benefits of the support material as used in this study was that it made security more concrete and manageable, structured the discussions, gave confidence, and supported non-experts as meeting facilitators. However, we are not aware that the companies continued using it after the study.

Implication for practice:
(P11) Meetings should allow ample time for discussions; thus, the agenda should not be too rigid.
(P12) One participant should be responsible for taking notes.
(P13) Notes on a shared screen, including notes from previous meetings, can support discussions.

Implication for research:
(R6) Though this study sheds light on the potential benefits of using support material in security meetings, the necessity of such material is not clear. The potential role of support material should be investigated further, including when such support is most needed, what support is effective for which types of meetings and contexts (e.g., with a department-scope vs. a project-scope), and how the needs for support material can vary with time as the company's security maturity changes.

Positive effects of the meetings were observed in all the studied cases, despite all cases struggling with how to get the most effect from the meetings. We even discovered a potential for the meeting to contribute with more cost-effective security, e.g., through supporting reuse across project and address non-functional security roles and tasks. **Security prioritisation and follow up** was essential for getting these positive effects. Still, confidence in the decisions was challenging, and it was not always clear how the security priorities were made. Similar challenges have been identified also for other security meeting types (Fig. 2).

Studies show that not all security vulnerabilities are exploited (Nayak et al., 2014). Thus, to arrive at cost-effective security, it is important to address those issues most likely to cause problems. Yet, we observed that questions like "what is most important?" often remained unanswered in the meetings, whereas other criteria (e.g., ease, concreteness) tended to be more used in the prioritisation.

A learning point from the meetings was that it was important to spread information to outsiders. Prioritisation of security did not only happen in the meetings — the prioritisation of security vs. features and other tasks that happened outside of the meetings was even more challenging than the prioritisation that happened in the meeting.

Implication for practice:
(P14) Beware of the tendency to prioritise easy and concrete tasks and newly identified tasks, without considering whether they are the most important tasks.
(P15) Beware of prioritising too many action points in a meeting. It is better to identify a few action points that end up being addressed, than identifying many action points where the understanding is that all will not be addressed.
(P16) For those action points that are prioritised, there is a need to ensure that they are concrete, and that responsibility and deadline are properly defined.
(P17) As the number of identified concerns grow, consider pruning the list outside of the meeting, to avoid overload and make the meeting more engaging.
(P18) Actions should be taken to bring information from the security meetings to a broader set of individuals. Sharing of such information in person (e.g., in meetings rather than on email) is preferred.

Implication for research:
(R7) As results indicate that security meetings can contribute to more cost-effective security, future research could investigate how meetings can be organised to support cost-effectiveness and how this cost-effectiveness can be assessed and made visible.
(R8) Research can contribute with prioritisation and decision-making support and strategies to be used in security meetings, to address the overarching challenge of confidence in security prioritisation.
(R9) Companies can benefit from improved strategies for communicating priorities from the meeting, and integrating them into their way of working.

6. Threats to validity

In the following we discuss the threats to the validity of our study results, using the classification scheme suggested for case studies (Runeson and Höst, 2009; Yin, 2018).

Construct validity can be defined as the "accuracy with which a case study's measures reflect the concepts being studied" (Yin, 2018) (Runeson and Höst, 2009). The concept of security priority is important in this study; we were interested in understanding the effects these meetings had on security prioritisation. 'Security priority' was operationalised through five influence categories: driving force, visibility, motivation, room to manoeuvre, and process match (Tøndel et al., 2022). These influence categories stem from one case study, and thus cannot be said to be an established theory on what brings priority to security in ASD. We are however not aware of the existence of such a theory. Though stemming from one case study, these influences are prominent also in the broader literature (Tøndel et al., 2022), strengthening our assumption that they would be relevant also in the contexts studied in this paper. We did find these influence categories useful in structuring and reasoning about our findings, something that supports their relevance. Still, there is a need for more studies that can form a stronger theoretical basis for understanding what brings priority to security in ASD. Measuring the effect on security prioritisation was however challenging, also with the use of these influence categories. Effects could be quite invisible (e.g., knowledge, motivation) and they could manifest outside of the meetings. Thus, we relied on participants reporting this effect in meetings or in interviews/retrospective.

Internal validity is concerned with causal relations and the risk that effects observed may have been caused by factors not considered in the study (Runeson and Höst, 2009). In qualitative studies, internal validity can be supported by strategies such as triangulation, thick descriptions, linking results to theory, seeking negative evidence, considering rival explanations, and having participants find the conclusions accurate (Miles et al., 2018). These strategies were applied in this study. Still, there are potential biases. The influence categories used in the analysis helped relate to the broad concept of security prioritisation. However, by building this strongly on these influence categories, that we ourselves had developed in a previous case study, there is a risk of confirmation bias. This study did not use these influence categories in order to confirm them. Still, the categories informed our understanding of security prioritisation and may have led us to see prioritisation that was not there, or miss aspects of prioritisation that we had not previously identified. However, these risks would be there also with a more inductive analysis approach, although less visible.

We acknowledge that in the cases where we used action research, our influence as researchers was considerable (Petersen and Gencel, 2013). We helped develop the meeting used in SmallCo and MediumCo, and thus there is a risk of bias related to us wanting this meeting to succeed. We have been aware of this risk throughout. Also note that the findings tip in favour of department level meetings (which we did not initially suggest). It is likely that our presence as researchers influenced the meetings we studied. Participants may have wanted to let the meetings we facilitated (MediumCo, SmallCo) look good to please us as researchers. They may also have wanted the meetings they organised (UnaidedCo) to come out as successful. We took care to regularly reflect on our influence as researchers, as part of the observation template. In interviews we made sure to express a need to not only hear about the good aspects of the meetings, but that we wanted to know about the challenging parts as well, so that we could improve. Our impression is that participants trusted us enough to give us their honest feedback. For us, it was particularly important that participants were honest about what they saw as effects of the meeting, as we did not have direct ways to measure this. In addition to encouraging participants to give honest feedback, we considered rival explanations in our analysis. To exemplify, in the study we saw that it was easier to get and see the effects of the meeting when you started from "nothing" (as in SmallCo). But then, what we saw as effects might not be effects of the meeting approach but rather effects of starting to work on security and to interact with external security experts. Thus, we were restrictive in claiming something to be effects of the meeting and made efforts to point to contextual factors that could be important for the effects.

External validity "is concerned with to what extent it is possible to generalise the findings, and to what extent the findings are of interest to other people outside the investigated case" (Runeson and Höst, 2009). As is common for case studies and action research, there are many contexts and many meeting approaches that we have not considered in this study. However, studying three companies with varying meeting operationalisations, allowed for identifying similarities. Thus, it pointed us towards findings that are likely to be shared by more than one context/meeting type. To support readers in judging whether the findings might be relevant for their context, we provided details on the company contexts and the meetings we studied (within the limitation that the anonymity of the companies should not be compromised). Note that the companies we have studied are of small or medium size, and we expect that results for larger companies may vary considerably from the findings in this study.

Reliability "is concerned with to what extent the data and the analysis are dependent on the specific researchers" (Runeson and Höst, 2009). In our study, reliability is supported through the use of an observation template, the recording and transcription of interviews, and the clear protocol for analysing the data. Further, data triangulation and member checking (as part of interviews, retrospectives, and the sharing of the draft research report) help support reliability. To exemplify, in interviews and retrospectives we took care to let the participants present their view, without first bringing in our understanding. Still, we also used the opportunity to provide our understanding when relevant, and get the interviewee to respond to that. We also shared a previous draft of this article with key study participants, and the feedback received support the validity of the findings. In the action research part of our study, two researchers were involved, where one took the main role as facilitator and the other did the data collection and analysis, thus being able to take a more external view of the meeting. Having only one researcher doing the analysis represents a threat to reliability. However, findings were discussed among the two researchers at several points throughout analysis. Replication is a general challenge for action research studies as it relies heavily on a trust relationship between the researcher and the company (Wieringa, 2014). However, to support replication in theory, we have provided details about the data collection instruments and the meeting concept used, as well as our approach to analysis.

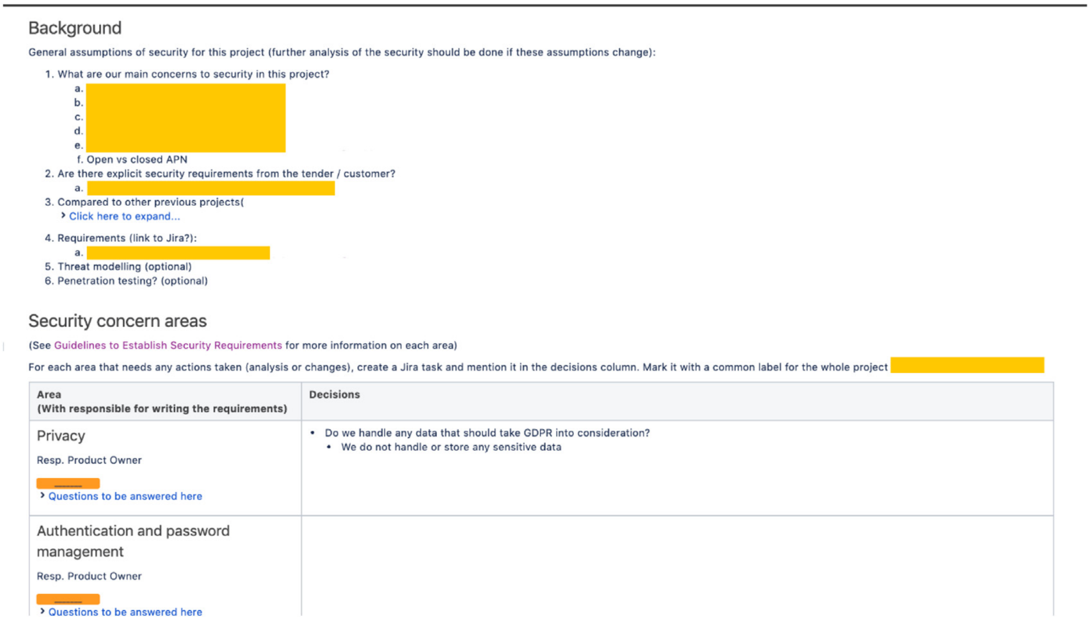
7. Conclusion

This article proposes regular security meetings as a strategy for continuous software security, and reports on a study of such meetings in three companies. The studied meetings varied in scope (project/team or department), participants (full team, seniors), support material, and context (small to medium size, low to medium security maturity).

Results from this study show that regular security meetings can contribute to ongoing priority of security, more visible and tangible security, and can reach key stakeholders — all these effects are called for in literature. Based on the lessons learned from the cases, we identify implications for practice and for research.

For practitioners, we find evidence that regular security meetings are useful for small and medium sized development companies that need to strengthen their software security maturity. Companies should seek to adopt a meeting approach aligned with their own needs, this includes selecting a meeting scope, participants, a meeting schedule, and a meeting structure. However, the lessons learned from this study can give some directions, e.g., pointing out the need for participants able to take responsibility for tasks, the importance of having time for discussions, and benefits and pitfalls of a department scope vs. a project scope.

For future research, we point to the need for more competence to improve the ability to support companies in selecting a meeting approach that suits their needs. This includes knowledge on how meetings should be adapted to different contexts and needs, knowledge on how to position the meeting to address systemic blockers for software security, and knowledge on the role of support material in different meeting models.



Background

General assumptions of security for this project (further analysis of the security should be done if these assumptions change):

1. What are our main concerns to security in this project?
   a. [redacted]
   b.
   c.
   d.
   e.
   f. Open vs closed APN
2. Are there explicit security requirements from the tender / customer?
   a. [redacted]
3. Compared to other previous projects(
   › Click here to expand...
4. Requirements (link to Jira?):
   a. [redacted]
5. Threat modelling (optional)
6. Penetration testing? (optional)

Security concern areas

(See Guidelines to Establish Security Requirements for more information on each area)

For each area that needs any actions taken (analysis or changes), create a Jira task and mention it in the decisions column. Mark it with a common label for the whole project [redacted]

Area (With responsible for writing the requirements)	Decisions
Privacy Resp. Product Owner [redacted] › Questions to be answered here	• Do we handle any data that should take GDPR into consideration? • We do not handle or store any sensitive data
Authentication and password management Resp. Product Owner [redacted] › Questions to be answered here	

Fig. 10. Excerpt from Confluence page of MediumCo.

Table 10
Influences on adoption (M = MediumCo, S = SmallCo, U = UnaidedCo).

Influence category	Contributes to the adoption	M	S	U	Hinders the adoption	M	S	U
Driving force	Someone with authority initiating and inviting to meetings	X	X	X				
	Participants are senior people			X				
Visibility								
Motivation	Participants that are motivated to meet and discuss security	X	X	X	Not all projects have a clear security need, thus meeting not always necessary		X	
	A general push for security in the company		X	X				
Room to manoeuvre	The cost of security meetings is accepted			X	Challenging to set aside time	X	X	
	The meeting is considered easy to do, and can be done quickly	X	X					
Process match	Ability to adapt the meeting to own needs		X		Disconnected from the "real work" they are doing that is more urgent	X		
	The practice of deciding on a time for the next meeting	X	X		Initial project work is creative and exploring, not considering security, etc.		X	
					If it turns out that all projects have similar security needs, a checklist may be a better option		X	

CRediT authorship contribution statement

Inger Anne Tøndel: Conceptualization, Methodology, Validation, Investigation, Writing – original draft, Visualization. **Daniela Soares Cruzes:** Conceptualization, Methodology, Validation, Resources, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors have previous relationships with two of the companies, either in form of previous part time deployment or being involved in consultancy projects towards companies.

Acknowledgements

This research was funded by the Research Council of Norway, grant nr. 247678. The authors would like to thank the companies where this study was performed, especially the interviewees and the participants in the meetings we observed. Thanks to our colleague Martin Gilje Jaatun for participating in the data collection at UnaidedCo. Thanks to Prof. Guttorm Sindre and our colleagues in the "Fabrikk" for many useful comments on this article.

Appendix A. Meeting approach used by the companies

The meetings in UnaidedCo had the following structure. In the beginning of the meeting, they went through the list of activities that was prioritised in the previous meeting, to assess status. In

Table 11
Overview of areas covered by the support material, as used in SmallCo.

	Area	Supporting questions
Overall concerns	Level of security we aim for	Why is security important in this project? What are our main assets that we need to secure? What are our main security concerns in this project? Are there explicit security requirements from the tender/customer?
	Analysis and training needs	Compared to previous projects, are there new types of technology, new types of security requirements, new types of threats, new types of assets? Do we plan considerable design changes in existing solutions? Do we have the necessary overview to understand the security risks in the solution? Are we aware of areas where we lack the necessary security competence?
	Security coding and testing practices	Do we have the necessary coding practices to ensure the security of our code is according to our needs? Is the current practice of code review sufficient? Do we need specific types of testing for security? What about pentesting?
Security areas	Privacy	What sensitive data is handled by the project?
	Authentication and password management	Which authentication and password mechanisms will be used here and where?
	Authorisation and role management	What is the RBAC for the front-end and back-end? What are the database user accounts and privileges? Which privileges will operations have on the databases? Which privileges others will have? Does some actions need extra authentication?
	Session management	Is there any type of session definition for a "Device?" Do we need to document this?
	Cryptography and key management	Where are we using Tokens and Keys for authentication? Which are they? Do we have a procedure that is good for managing these? Should we have changes in the way we do things today?
	Network security	Are there any requirements to be added based on the list of equipment? Do we need a better description of how the connections will be done and who will be part of this? What are the requirements for the network setup on the Server side? Who will be responsible for intrusion detection? What are the trust boundaries here? Do we need to describe back-end access and how it is secured? How is the access to the network? What will be the privileges?
	Audit logging and analysis	Which logging is needed for different reasons, such as repudiation and for detecting attacks or problems in the system? Is there any logging of actions we should do for cases of auditing in our systems?
	Attack detection	Who is responsible for doing performing attack detection? Do we depend on any other parties, e.g., where we are not the only one officially managing the network and resources? Which type of attacks will we try to detect? Will we revoke a device after it was registered? How do we assure the integrity of the data about devices? Can we revoke registration of a device? Which situations should we do that?
	Incident management	What incidents can happen? What procedures do we need to have in place for incident management? Do we need to train for any specific incidents?
	Physical security	What are the components that we need to physically protect? How will we protect these components? What is it that we are assuming that makes the device secure in the platform? Can we revoke registration of a device? Which situations should we do that? Are there recommendations that we need to give to the customers?
	Availability protection	Which types of assets are important to be available all the time? When do we need to take backup? Who is responsible of backup of what? What are the requirements to backup of data?

(*continued on next page*)

cases where other security activities had been done that were not on the list, these were also informed about and discussed. Then followed an open discussion about security concerns that should be noted down and addressed. After this open discussion, they opened the excel sheet where they had recorded all previously identified security concerns. This excel sheet contained a short description of each concern, in addition to information on which application it concerned, what was the status, what was the priority, and who (if any) was responsible. Then, after going through the excel sheet, they decided on a set of activities to focus on in the next period. Throughout the meeting facilitator took notes that were shown on screen. Notes could be taken directly in a meeting memo, or in the excel sheet.

The meeting in MediumCo and SmallCo had a different structure. These meetings made use of support material that offered a set of areas to cover and questions to aid in making decisions. These were organised in a Confluence page in MediumCo and an excel sheet in SmallCo. The security areas were adapted from Firesmith (2003) for use in MediumCo, and the questions were initially developed based on the needs of the first project where

Table 11 (*continued*).

	Area	Supporting questions
	Data security and integrity	Who will take care of the integrity of the data that we get from third parties? Do we need to worry about this? On cases of corruption of the data, how do we recover? Which types of procedures for database protections for the back end we will have? How are we planning to make sure the data on each device is correct and updated? Do we need? How are we planning to have backup in case the data gets corrupted?
	Third party component analysis	Which third party components will be involved here and how this can affect security? What are the entry points from other systems? Where is data integrity most important? For what data, for what functionality, for what input?
Release notes	Release notes to operations	NA
	Release notes to customers	NA

these meetings were applied. Fig. 10 shows how the supporting Confluence page looked like in MediumCo. Table 11 gives an overview of all the security areas covered in the support material, in the form used by SmallCo. MediumCo and SmallCo used the same security areas but had slightly different support for the overall concerns (in MediumCo covered by the Section Background in the Confluence page, see Fig. 10). Moreover, the supporting questions had been made less company and technology-specific before bringing them to SmallCo.

In a typical meeting in MediumCo and SmallCo, they started with going through the action points from the previous meeting before the facilitator selected a few security areas to focus on in the meeting. These were then discussed, and notes were taken (visible on screen) on decisions, concerns, and open issues identified in the discussions. As part of this they identified action points. To exemplify, one meeting in MediumCo had already identified in the agenda three areas from the checklist they wanted to discuss (privacy, network security, and key management). Then in the meeting they discussed the associated questions and any already noted concerns on Confluence, and updated the documentation in Confluence on these issues. Then, due to extra time, they moved on to discuss a few more security areas as well. In the initial meeting, the main emphasis was on the open issues in the beginning of the template (e.g., what are our main concerns to security in this project?) while later meetings went more into detail on the different areas, and revisited previous decisions and discussions. The exception to this approach was when the meeting series was brought to the new team in MediumCo, and the team managed to go through the full template in one meeting.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jss.2022.111477.

References

Ahmad, M.O., Markkula, J., Oivo, M., 2013. Kanban in software development: A systematic literature review. In: 2013 39th Euromicro Conference on Software Engineering and Advanced Applications. http://dx.doi.org/10.1109/SEAA.2013.28.

Assal, H., Chiasson, S., 2019. 'Think secure from the beginning': A survey with software developers. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. http://dx.doi.org/10.1145/3290605.3300519.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., 2001. Manifesto for agile software development. https://agilemanifesto.org/.

Behutiye, W., Karhapää, P., López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A.M., Rodríguez, P., Franch, X., Oivo, M., 2020. Management of quality requirements in agile and rapid software development: A systematic mapping study. Inf. Softw. Technol. 123, 106225. http://dx.doi.org/10.1016/j.infsof.2019.106225.

Bernsmed, K., Cruzes, D.S., Jaatun, M.G., Iovan, M., 2022. Adopting threat modelling in agile software development projects. J. Syst. Softw. 183, 111090. http://dx.doi.org/10.1016/j.jss.2021.111090.

Bishop, D., Rowland, P., 2019. Agile and secure software development: An unfinished story. Issues Inf. Syst. 20 (1).

Crawley, B., Glas, B., Jenkins, B., Cooper, C., Kefer, D., Parekh, H., Dileo, J., Ellingsworth, J., Kennedy, J., Kisserli, N., Duarte, P., Arriada, S., Kravchenko, Y., 2020. Presenting OWASP SAMM - OWASP SAMM V2.0 - Core Model Document. OWASP. https://github.com/OWASP/samm/blob/master/Supporting%20Resources/v2.0/OWASP-SAMM-v2.0.pdf.

Cruzes, D.S., Jaatun, M.G., Bernsmed, K., Tøndel, I.A., 2018. Challenges and experiences with applying microsoft threat modeling in agile development projects. In: 2018 25th Australasian Software Engineering Conference. ASWEC, http://dx.doi.org/10.1109/ASWEC.2018.00023.

Davison, R., Martinsons, M.G., Kock, N., 2004. Principles of canonical action research. Inf. Syst. J. 14 (1), 65–86. http://dx.doi.org/10.1111/j.1365-2575.2004.00162.x.

Davison, R.M., Martinsons, M.G., Ou, C.X., 2012. The roles of theory in canonical action research. MIS Q. 763–786. http://dx.doi.org/10.2307/41703480.

Deterding, N.M., Waters, M.C., 2021. Flexible coding of in-depth interviews: A twenty-first-century approach. Sociol. Methods Res. 50 (2), 708–739. http://dx.doi.org/10.1177/0049124118799377.

Firesmith, D.G., 2003. Common Concepts Underlying Safety Security and Survivability Engineering. The Software Engineering Institute, Carnegie-Mellon University, https://resources.sei.cmu.edu/asset_files/TechnicalNote/2003_004_001_14198.pdf.

Fitzgerald, B., Stol, K.-J., 2014. Continuous software engineering and beyond: Trends and challenges. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering. http://dx.doi.org/10.1145/2593812.2593813.

Heeager, L.T., Nielsen, P.A., 2018. A conceptual model of agile software development in a safety-critical context: A systematic literature review. Inf. Softw. Technol. 103, 22–39. http://dx.doi.org/10.1016/j.infsof.2018.06.004.

Jarzębowicz, A., Weichbroth, P., 2021. A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices. In: Przybyłek, A., Miler, J., Poth, A., Riel, A. (Eds.), Lean and Agile Software Development, Vol. 408. LASD 2021, Springer, Cham, pp. 91–110. http://dx.doi.org/10.1007/978-3-030-67084-9_6.

Kocksch, L., Korn, M., Poller, A., Wagenknecht, S., 2018. Caring for IT security: Accountabilitie, moralities, and oscillations in IT security practices. In: Proc. ACM Hum.-Comput. Interact. 2.CSCW. http://dx.doi.org/10.1145/3274361.

Kongsli, V., 2006. Towards agile security in web applications. In: OOPSLA '06: Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications. ACM, http://dx.doi.org/10.1145/1176617.1176727.

Lindsjørn, Y., Sjøberg, D.I.K., Dingsøyr, T., Bergersen, G.R., Dybå, T., 2016. Teamwork quality and project success in software development: A survey of agile development teams. J. Syst. Softw. 122, 274–286. http://dx.doi.org/10.1016/j.jss.2016.09.028.

Maxwell, J.A., 2013. Qualitative research design: An interactive approach. In: Applied Social Research Methods Series, vol. 41, Sage publications.

McGraw, G., Allen, J.H., Mead, N., Ellison, R.J., Barnum, S., 2013. Software Security Engineering: A Guide for Project Managers. Carnegie Mellon University, Software Engineering Institute, https://apps.dtic.mil/sti/pdfs/ADA617944.pdf.

Migues, S., Erlikhman, E., Ewers, J., Nassery, K., 2021. BSIMM12 2021 Foundations Report. Synopsis. https://www.bsimm.com/content/dam/bsimm/reports/bsimm12-foundations.pdf.

Miles, M.B., Huberman, A.M., Saldaña, J., 2018. Qualitative Data Analysis: A Methods Sourcebook. Sage publications.

Moe, N.B., Dingsøyr, T., Rolland, K., 2018. To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development. Int. J. Inf. Syst. Project Manag. 6 (3), 45–59. http://dx.doi.org/10.12821/ijispm060303.

Nayak, K., Marino, D., Efstathopoulos, P., Dumitraş, T., 2014. Some vulnerabilities are different than others. In: International Workshop on Recent Advances in Intrusion Detection. Springer, http://dx.doi.org/10.1007/978-3-319-11379-1_21.

Newton, N., Anslow, C., Drechsler, A., 2019. Information security in agile software development projects: A critical success factor perspective. In: 27th European Conference on Information Systems. ECIS.

Oueslati, H., Rahman, M.M., Othma, Lb, 2015. Literature review of the challenges of developing secure software using the agile approach. In: 2015 10th International Conference on Availability, Reliability and Security. http://dx.doi.org/10.1109/ares.2015.69.

Palombo, H., Tabari, A.Z., Lende, D., Ligatti, J., Ou, X., 2020. 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.

Petersen, K., Gencel, C., 2013. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In: 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement. http://dx.doi.org/10.1109/IWSM-Mensura.2013.22.

Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131–164. http://dx.doi.org/10.1007/s10664-008-9102-8.

Schwaber, K., 2004. Agile Project Management with Scrum. Microsoft Press.

Stray, V., Sjøberg, D.I.K., Dybå, T., 2016. The daily stand-up meeting: A grounded theory study. J. Syst. Softw. 114, 101–124. http://dx.doi.org/10.1016/j.jss.2016.01.004.

Strode, D.E., Huff, S.L., Hope, B., Link, S., 2012. Coordination in co-located agile software development projects. J. Syst. Softw. 85 (6), 1222–1238. http://dx.doi.org/10.1016/j.jss.2012.02.017.

Tøndel, I.A., Cruzes, D.S., Jaatun, M.G., 2020. Achieving good enough software security: The role of objectivity. In: EASE '20: Proceedings of the Evaluation and Assessment in Software Engineering. pp. 360–365. http://dx.doi.org/10.1145/3383219.3383267.

Tøndel, I.A., Cruzes, D.S., Jaatun, M.G., Rindell, K., 2019a. 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. http://dx.doi.org/10.1145/3339252.3340337.

Tøndel, I.A., Cruzes, D.S., Jaatun, M.G., Sindre, G., 2022. Influencing the security prioritisation of an agile software development project. Comput. Secur. 118, 102744. http://dx.doi.org/10.1016/j.cose.2022.102744.

Tøndel, I.A., Jaatun, M.G., 2020. Towards a conceptual framework for security requirements work in agile software development. Int. J. Syst. Softw. Secur. Prot. (IJSSSP) 11 (1), 33–62. http://dx.doi.org/10.4018/IJSSSP.2020010103.

Tøndel, I.A., Jaatun, M.G., Cruzes, D.S., Williams, L., 2019b. Collaborative security risk estimation in agile software development. Inf. Comput. Secur. 26 (4), 508–535. http://dx.doi.org/10.1108/ICS-12-2018-0138.

Tuladhar, A., Lende, D., Ligatti, J., Ou, X., 2021. An analysis of the role of situated learning in starting a security culture in a software company. In: USENIX Symposium on Usable Privacy and Security. SOUPS 2021.

Türpe, S., Poller, A., 2017. Managing security work in scrum: Tensions and challenges. In: The International Workshop on Secure Software Engineering in DevOps and Agile Development. SecSE, pp. 34–49.

van der Veer, R., 2019. SAMM agile guidance. https://owaspsamm.org/guidance/agile/.

Villamizar, H., Kalinowski, M., Viana, M., Fernández, D.M., 2018. A systematic mapping study on security in agile requirements engineering. In: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, IEEE, http://dx.doi.org/10.1109/SEAA.2018.00080.

Weir, C., Becker, I., Blair, L., 2021. A passion for security: Intervening to help software developers. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice. ICSE-SEIP, http://dx.doi.org/10.1109/ICSE-SEIP52600.2021.00011.

Weir, C., Becker, I., Noble, J., Blair, L., Sasse, M.A., Rashid, A., 2020a. Interventions for long-term software security: Creating a lightweight program of assurance techniques for developers. Softw. - Pract. Exp. 50 (3), 275–298. http://dx.doi.org/10.1002/spe.2774.

Weir, C., Rashid, A., Noble, J., 2020b. Challenging software developers: Dialectic as a foundation for security assurance techniques. J. Cybersecur. 6 (1), http://dx.doi.org/10.1093/cybsec/tyaa007.

Wieringa, R.J., 2014. Design Science Methodology for Information Systems and Software Engineering. Springer.

Wieringa, R., Moralı, A., 2012. Technical action research as a validation method in information systems design science. In: Design Science Research in Information Systems. In: Advances in Theory and Practice. DESRIST, vol. 212, Springer, http://dx.doi.org/10.1007/978-3-642-29863-9_17.

Williams, L., Gegick, M., Meneely, A., 2009. Protection poker: Structuring software security risk assessment and knowledge transfer. In: Engineering Secure Software and Systems. ESSoS 2009, Springer, http://dx.doi.org/10.1007/978-3-642-00199-4_11.

Williams, L., Meneely, A., Shipley, G., 2010. Protection poker: The new software security game. IEEE Secur. Priv. 8 (3), 14–20. http://dx.doi.org/10.1109/msp.2010.58.

Wohlin, C., Aurum, A., 2015. Towards a decision-making structure for selecting a research design in empirical software engineering. Empir. Softw. Eng. 20 (6), 1427–1455. http://dx.doi.org/10.1007/s10664-014-9319-7.

Yin, R.K., 2018. Case Study Research and Applications, sixth ed. Sage.

Inger Anne Tøndel is a Ph.D. candidate at the Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim. Recently, she held a position as a senior research scientist at SINTEF Digital, Trondheim, Norway, and her research interests include software security, security requirements, information security risk management, cyber insurance, and smart-grid cybersecurity. She now works as a senior advisor at the Norwegian Directorate of e-health. Tøndel received an M.Sc. in telematics from NTNU in 2004.

Daniela Soares Cruzes is a professor at the Department of Computer Science, NTNU, Trondheim, Norway. Her research interests are agile software development, software security, software-testing processes, empirical research methods, theory development, and synthesis of software-engineering studies. Cruzes received her Dr.Ing. in electrical and computer engineering with emphasis in empirical software engineering at the University of Campinas, Brazil, in 2007. She has two postdoctoral studies, one at the Fraunhofer Center at the University of Maryland, College Park, and one at the Norwegian University of Science and Technology, Trondheim. She is a member of committees with various highly ranked international conferences and journals.