IT security is from Mars, software security is from Venus

Inger Anne Tøndel, Martin Gilje Jaatun, Daniela Soares Cruzes

Abstract: In smaller software companies, the divide between IT security and software security can result in software security not being prioritized. A formal security champion role in the development team and collaborative risk-based security activities are potential ways to reduce this divide and bring a more proactive approach to software security.

More than ten years ago, van Wyk and McGraw [1] called out for aligning information security and software development. At that time, there was a disconnect between security and development that led to software being development without any understanding of technical security risk, and thus software with security weaknesses that should have been avoided. Even though the software security landscape has changed a lot in the past ten years, with increasing exposure of software and growing attention to security issues, this disconnect is still present in software companies (Table 1).

We have studied software security practices and challenges in 23 public organizations in Norway [2] through interviews with employees having various roles related to security in these organizations. This includes CISOs or other personnel with IT security or network security roles in the organizations, as well as software architects. The organizations we studied vary in size, but most can be considered small or medium sized companies (SMEs). The public organizations additionally varied in how much software was internally developed and what role the organization itself had in the development; whether they did development in-house, hired external developers, acquired bespoke software from vendors, or a combination of the above. The majority had their own software developers, who often worked together with hired external developers.

In the study we identified various ways in which the disconnect between security and development is still prominent in these smaller organizations, as well as reasons why this is so. In the following we explain these reasons further, and provide some suggestions for how to move forward from here. We found that to the extent that IT security professionals are involved in the development, this involvement is not strategic, and the man-hours put into this interaction is very limited. Additionally, a lot of the good work that is done on IT security and network security in the organization, does not seem to influence software development – it rather is seen as irrelevant for the development. This goes for business-level information security risk analysis as well as for penetration testing.

Table 1. To what extent do the organizations and projects follow the recommendations of van Wyk and McGraw on information security professionals' involvement in software development projects?

Recommendation van Wyk and McGraw	IT security professionals' practice	Software development professionals' practice
Abuse cases: Security professionals have knowledge of attacks, and should participate together with developers in creating abuse cases	In general, IT security practitioners are *not* involved in creating abuse cases, but they may have conversations with developers about threats and security requirements.	Only very few create abuse cases.
Business risk analysis: Information security professionals know security impacts first hand for similar business applications, and can thus provide answers to questions on incident costs.	Perform overall risk analysis for the whole organization, but these are often considered by developers to not be relevant for the development projects.	Several organizations do risk analysis related to development projects, but these do not necessarily cover security risks. Only a few have clear routines to do software security risk analysis related to the development projects.
Architectural risk analysis: A security analyst that is also a technology expert (covering application, underlying platform, frameworks, languages, etc.) can provide important perspectives on risks, weaknesses and mitigation strategies.	-	When security architects are involved in the projects, these may evaluate risk related to architectural decisions and follow up on security principles. This is however seldom done unless security is a clear priority.
Test planning: Risk based testing scenarios would benefit from experiences of incident handlers. Security professionals are good at "thinking like an attacker".	-	Testing mainly covers functionality.
Code review: This step is best left in the hands of the development organization.	-	Code review is commonly performed, but related to code quality in general (no specific focus on security).
Penetration testing: This is usually the domain of information security and incident handling organizations, but for software development a more inside -> out approach should be taken	Several organizations do penetration testing, but not necessarily directed at the software they develop. Initiatives for penetration testing often come from outside the development organization.	A few do penetration testing at main release, or if they suspect major security issues.

Recommendation van Wyk and McGraw	IT security professionals' practice	Software development professionals' practice
Deployment and operations: Information security expertise can help safely setting up the application in a secure operational environment; access controls, event logging and monitoring, etc.	Security is in general a much higher priority and concern in operation (network security). This part of the organization usually has routines to stay updated on attacks and security risk.	Different culture among developers and operations when it comes to security. This may lead to frictions; e.g. developers believe operations put up too many hindrances.

So, where does it go wrong? The evidence we have collected points to three main reasons why software security is not given priority, as summarized in Table 2. These reasons concern both the IT security and development tribes. In the following we go into each of these reasons, explain the challenges we identified and provide suggestions for moving forward.

Table 2: Key reasons why software security is not given priority, both among IT security professionals and in the development organization

Unclear responsibilities and expectations on software security	Risk perception	Lack of approaches that fit the software development daily activities
• No one is given explicit responsibility for software security • Optimistic assumptions on competence and interest of developers and contractors on security	• Software security not important for internal systems • Security is about confidentiality • Contractors and developers can be trusted	• Approaches to security that worked for waterfall-based development do not work as well with agile • IT security people not involved in sprint meetings or other key decision making points

Unclear responsibilities: Where does IT security stop and software security begin?

Commonly, information security is defined as safeguarding the confidentiality, integrity and availability of information, and IT security is broadly defined as information security in IT systems. In today's businesses, this information is in large part processed by software systems, thus software security is essential for information security. Information security management standards, such as ISO/IEC 27001, include controls on system acquisition, development and maintenance. It is therefore not a surprise that in the organizations, IT security personnel are often given some responsibility for software security.

McGraw defines software security as "the idea of engineering software so that it continues to function correctly under malicious attack" [3]. Table 3 provides an overall comparison between the information security and software security fields. The fields are clearly related. Still, there are major differences in the formal requirements to, and the organization of, the work. van Wyk and McGraw recommended that a fruitful cooperation between information security people and developers could help developers understand what they're up against and potential impacts on the business. To achieve such a fruitful cooperation, they recommended that information security professionals should be involved in some of the software security touchpoints (Table 1). But van Wyk and McGraw were very clear that this required skills and initiative from the IT security side. Gaining the necessary understanding of software development in order to contribute with security in a meaningful way, and in a way that is respected by the developers, is non-trivial. This included understanding the craft of developing software, as well as what are the goals driving the development and its race towards faster time to market.

Table 3. IT security and software security compared

	ITSEC	SWSEC
FORMAL RESPONSIBILITIES	Place in the hierarchy; a formal role with responsibilities.	No formal position. Autonomous teams
MATURITY	Standards are adopted. Mature tools.	Standards not often used. Less mature tools.
DRIVERS	Risks; incidents; standards and legislation.	Requirements (legislative and customer demands); software vulnerabilities.
RESTRICTIONS	Costs-effectiveness; management buy-in.	Time to market.
GENERAL MINDSET	Sceptic and risk averse.	Optimistic – build things.
SPEED	Two paces: i) race against attackers (patch, update signatures, etc.) ii) ISMS – plan, do, check, act – longer cycles (e.g. risk analysis once a year)	Agile, DevOps, continuous delivery -> need to keep up with this pace.

Software security rely on individual initiative

Most organizations in our study point to IT security people as the ones having responsibility also for software security. However, IT security people seem not to be highly involved in software development, and for them, their responsibility for software security is unclear. As a result, the responsibility is fragmented, and it is not possible to clearly hold anyone accountable for software security. Some IT security professionals stated that, in the end, the developers are responsible for their own part of the system, and that in their organization, other security activities and goals had been given priority. Consequently, software security had not been given attention. Many organizations seem to rely quite heavily on their contractors to take care of software security, and do not really follow up on them regarding security issues. They rely on contractors to identify security requirements, and assume that they have an overview of security risk and perform the right activities to address this risk.

IT security professionals may occasionally discuss security issues with developers, but they do not follow up on how this is dealt with in the development. IT security professionals view architects as important and potential allies in the software security work. However, in practice the architects seldom take on this role as a security ally. The architects often come from external contractors, and are thus mainly concerned with getting the job done, and following the product owner's orders. Interviewees talk about situations where the architect does not take responsibility for security, and instead points to the CISO or similar role. Since architects are considered to be a primary influencer on whether there is, e.g., performed design or architecture review related to security, software security currently relies on individual initiative and interest of security among the architects. On the plus side, since architects typically are seasoned developers with significant experience, it is likely that they will have more software security knowledge than the average developer, as our studies indicate that software security knowledge is correlated with years of experience [4]. This means that architects should be well placed to fill a role in software security, but this responsibility needs to be assigned explicitly by management.

The CISO as a change agent for software security?

For software security to gain momentum, someone needs to ask for more software security to drive change in practices. If assuming that the disconnect between IT security personnel and developers is the main reason why software security is not happening, one could say that the initiative for software security should come from the IT security side. This is also very much what van Wyk and McGraw build on in their article, providing recommendations for information security experts on how to become more involved in development. However, we don't see evidence that this is happening.

So, if neither the security nor the developer side is pushing for more software security, who should? As a general rule, management is key in driving change in organizations [5]. They are in a position to push security in the organization, either information security or software security. In the organizations studied there is a big difference in the awareness and push from managers on information security compared to software security. At the time of the first part of the study (2013), all the public organizations were required to implement an information security management system (ISMS). At the same time, large public development projects could run without software security being a main consideration, and without anyone being given clear responsibility for software security.

Though information security practitioners and developers often have a common technical background, the former rarely have strong development expertise. Thus, it is generally recommended that responsibility for software security should be assigned to someone from the development side. Based on data from the BSIMM study [6], the first step in a software security initiative should be the formation of a software security group (SSG), responsible for carrying out and facilitating software security in the organization. This group should ideally consist of software security people, alternatively developers that can be taught about security. The SSG should have people with deep coding skills as well as architects, and people with good communication skills. It has been stated that network security people usually fail in this type of role [6].

If looking at organization charts, CISOs (and other information security or network security people) are usually not located anywhere near the development organization. It has previously been shown how the silo structure of organizations can limit communication about and learning from cyber incidents [7]. Similarly, organizations may find that the brilliant information security competence they have in-house does not benefit the development at all. In the organizations studied, developers are not included in security forums that, e.g., discuss attacker trends and risks, and IT security people are only occasionally in interaction with the development organizations. In the organizations that rely quite heavily on external contractors for development, the linkage and proximity of the IT security and development people are an even bigger challenge.

Security Champions can be the bridge between IT security and Software Security

Just telling software developers and IT security people that they need to play together has not been working – a decade's worth of empirical evidence tells us this. Another change agent is necessary, and we propose that this role can be filled by *software security champions* [8, 9]. As mentioned before, establishing a security champion program also needs to be management-driven. Developers with an above-average interest in software security need to be identified or hired, and care must be taken that every team has at least one security champion. This will require management support and funding. However, this person must NOT be an external IT security expert – it is instrumental that the security champion is a developer, contributing to the development process and the quest toward "done". The BSIMM [6] also highlights the important role of the security champions, but uses the term "satellite" instead, suggesting that they are somewhat secondary to the SSG. We believe that for SMEs, it will be more beneficial to start with the software security champions, and possibly migrate to creating an SSG if and when the organization reaches the requisite size.

Once the security champions are in place, they can serve as the bridge between the CISO and the developers – their security knowledge should let them understand the "security-speak" of the CISO, and their developer chops and positive contributions to the fight against the windmills represented by the backlog should ensure a sympathetic ear among their fellow developers. For organizations that have an SSG, this just adds another layer to the organization chart, as shown in *Figure* 1. It is important that the security champions organizationally are placed in the same hierarchy as other developers, ultimately reporting to the Vice President of Development or similar role. We believe the same holds true for the SSG – even though it should have clear lines to the CISO, it is still part of the development organization

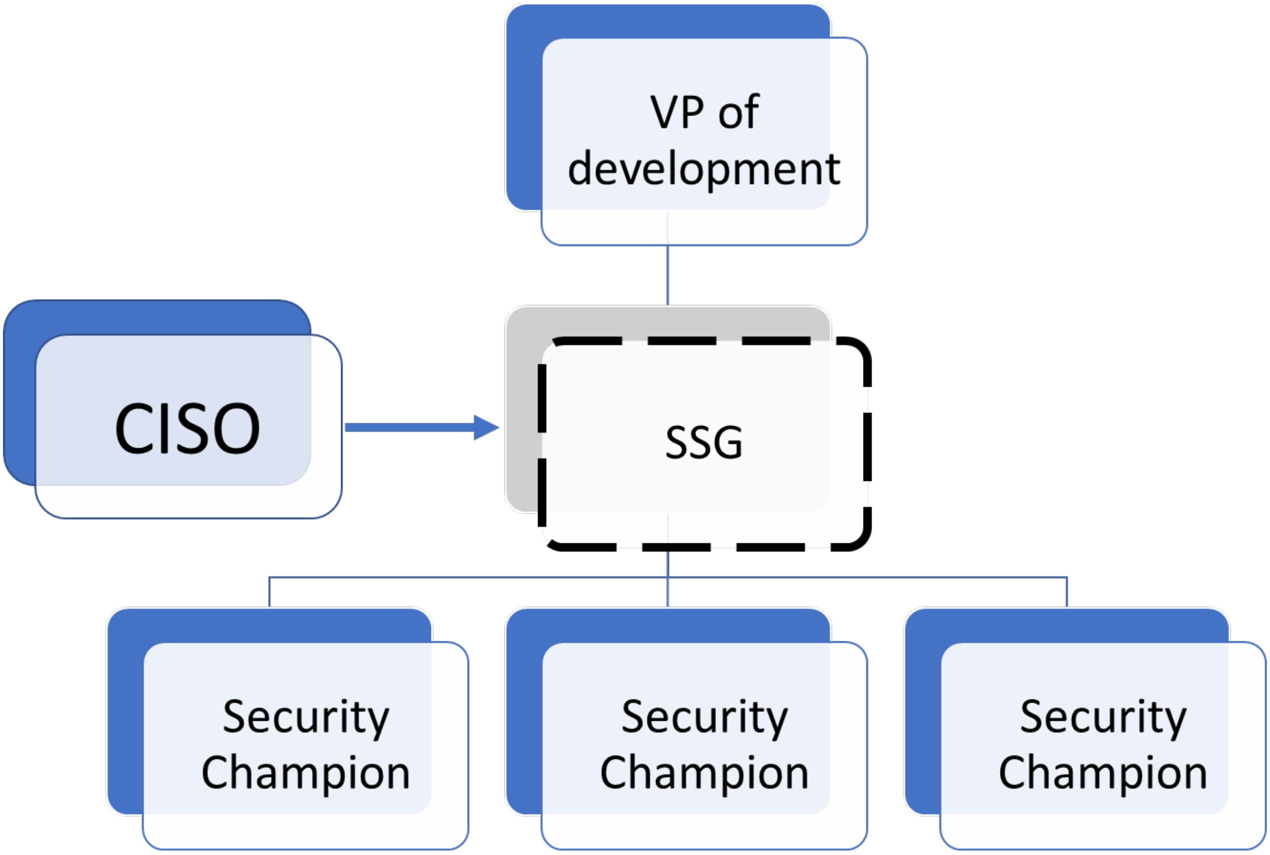

Figure 1: Shoehorning security into the organization chart

Risk perception

Despite the common mantra in literature and in security circles that all security work should be risk based, we did not find much evidence that software security follows a risk-based approach in the organizations we studied. Instead the approach can be characterized as compliance-based or accidental.

The organizations in our study were up to date on legal requirements, network security and new risks in IT. This however does not mean that the software development initiated by these organizations benefited from that competence. Though the studied organizations often have forums and similar to follow up on the latest cyber security threats, none seem to have clear routines to inform developers about new or evolving threats. Only a few organizations have identified what their most important product is, or which kinds of attacks they are most afraid of related to their software.

When the systems under development are not to be open for external users, and thus not directly available on the Internet, security is not considered to be particularly important. This is the case for many of the systems developed by or for these organizations, and in general, security does not seem to be prioritised. Instead, efficiency (i.e., getting the job done) is considered the main priority both from those procuring and those developing the software. Additionally, the organizations seem to have significant trust in the software developers, including contractors, that they all have good intentions. Thus, they do not feel the need to have mechanisms in place to check for rogue code, and do external security tests, etc.

IT security personnel are not necessarily involved in making decisions on what level of security is needed for the project. Software security is more likely to be considered for development of new products, than for improvements of existing products, and if the need for security is rather obvious (e.g. health information), as this can spur the involvement of security experts in the project. If the need is not obvious, security may not be considered – unless something triggers a sudden jump in security attention.

The main trigger for security activities in the development projects are accidentally detected security vulnerabilities and legal requirements. In the one project we studied that had security as a high priority, this was solely because of strict legal requirements to the type of data the software should handle. Because of these legal requirements, security was given priority in the budget and security architects were included in the development teams. Legal requirements are additionally stated as a reason for doing risk analysis. In some cases, risk analysis is first performed after receiving audit remarks that this is lacking. For projects without a clear approach to security, security activities come a bit incidental and late (if at all).

Risk centred activities as an antidote

As development organizations increasingly become aware of the need to address security during development efficiently and effectively, without hampering their agile approach to development, they need to make assessments on what type of security activities to include, i.e., what security activities pay off. We advocate that companies would benefit from taking a risk-based approach in their selection of software security activities, to ensure they are conscious about how much security and what type of security is most needed in their specific project. Making such decisions is challenging and requires security competence. However, there are many activities that development projects can adopt that make such decisions more available to the team, also in cases where they lack deep security knowledge.

The most often mentioned activity is Threat analysis/Threat modeling. In such activities the development teams can be supported by checklists or mnemonics such as STRIDE, or they can even utilize game-based approaches such as OWASP Cornucopia [10] and Microsoft EoP [11]. Alternatively, including developers in risk analysis activities can make these analyses more relevant to the team and increase the security awareness of those that participate. Protection Poker [12], a game-based approach to risk estimation, offers a way to easily integrate security risk analysis into agile development practices. These activities require little preparation, require relatively low effort to perform, and contribute to building security awareness in the whole team. Essentially, they are structured ways to talk about security risk, and may be the little push that is required to spend more time on software security activities. Including IT security personnel in the activities can further improve the quality of the security discussions they foster, and increase awareness and understanding on both sides; making the development teams more knowledgeable and aware of security issues, and increase IT security personnel's understanding of the challenges in development.

Security in Software Development Practices: Speed, Data, Ecosystems

It is clear that many IT security professionals and developers have conflicting goals. While the IT security people in general express a concern that software security may not be adequately handled in their organization, both groups explain that the main priority during development is functionality. Additionally, in cases where security requirements are identified early on, time pressure in the projects can cause security requirements to be postponed, even past deployment. With traditional waterfall development methods, all requirements were in the contract and had to be fulfilled. Now, important security decisions are made in short sprint meetings, localized in the scope of the sprint, and the impression is that whenever there is a conflict over time and budget in a sprint, security is sacrificed in order to realize functionalities. Most organizations we have studied are struggling to integrate security into the agile way of working. One participant explained that information security is not included until the end of the project, as information security personnel is not invited to participate before then, but at that point developers do not like it when she introduces new security requirements.

According to an article by Bosch [13], there are three key factors driving the future of software engineering:

- *speed:* continued success depends on the organization's ability to respond quickly to customer requests, changing market priorities, new competitors, etc.
- *data:* data collection is now cheap and easy, and organizations that are able to make smart and timely decisions based on collected data will benefit
- *ecosystems:* as a result of increased speed and data, companies will more frequently change their role and position in their ecosystem, thus organizations need to intentionally, proactively and effectively manage changing relationships

These drivers impact the whole organization, not only the development teams. For software development, there has long been a drive towards less documentation and faster delivery of working code, through agile development, DevOps, and continuous delivery. Where traditional organizations relied on functional organizational hierarchies, businesses now move towards cross-functional teams and self-management. CISOs and similar organizational roles need to find their way in an organization structure more centered on autonomous teams, where speed of decisions is key to maintain competitiveness. We find that this aligns well with the suggestion that the CISO exerts influence on a distributed band of security champions, possibly via the SSG, without removing their organizational ties to the development organization.

Agile is all about value and functionality; what you can show to the customer in the next sprint. Security tends to be considered a non-functional requirement, and with the main emphasis on functionality, such non-functional requirements are easily sacrificed on the altar of progress when backlog priorities are set. In case of security incidents, there is a need for rapid response, but the overall challenges are more towards getting management approval of needed, longer-term security investments. However, we expect that as organizations move toward DevOps and continuous deployment, IT security will automatically need to be tighter integrated with development, and their horizons will shift accordingly.

It can be argued that Ops are in general more security aware, since they have to deal with daily intrusion attempts and manage firewalls, antivirus tools and intrusion detection systems. The DevOps mantra is "you build it – you run it", which implies that the developers will be much closer to the sharp end, also when an incident should happen. This also means that developers need to be involved in incident response drills, ultimately resulting in better response and quicker fixing of security bugs and flaws.

Along with the drive towards value and functionality fast, there is a growing attention to the fact that software needs to be developed with security in mind. Cyber incidents are visible in media, and this increases awareness among managers as well as customers. It is thus likely that in the future also non-security-critical software will need to consider security as an important quality attribute, and that customers will pose security requirements to the software they acquire. This in itself can increase the drive for security in the development organization without IT security people needing to take the role of a change agent for software security.

Acknowledgements

This work was supported by the *Science of Security in Agile Software Development* project (SoS-Agile), funded by the Research Council of Norway (grant number 247678).

References

[1] K. R. van Wyk and G. McGraw, "Bridging the gap between software development and information security," *Security & Privacy, IEEE,* vol. 3, pp. 75-79, 2005.

[2] I. A. Tøndel, M. G. Jaatun, D. S. Cruzes, and N. B. Moe, "Risk Centric Activities in Secure Software Development in Public Organisations," *International Journal of Secure Software Engineering (IJSSE),* vol. 8, pp. 1-30, 2017.

[3] G. McGraw, "Software security," *IEEE Security & Privacy,* vol. 2, pp. 80-83, 2004.

[4] T. D. Oyetoyan, M. G. Jaatun, and D. S. Cruzes, "A Lightweight Measurement of Software Security Skills, Usage and Training Needs in Agile Teams," *International Journal of Secure Software Engineering,* vol. 8, p. 27, January 2017.

[5] H. Assal and S. Chiasson, "Security in the software development lifecycle," presented at the Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018), 2018.

[6] G. McGraw, S. Migues, and J. West, "Building Security In Maturity Model (BSIMM9)," Synopsys October 2018.

[7] A. Ahmad, J. Hadgkiss, and A. B. Ruighaver, "Incident response teams–Challenges in supporting the organisational security function," *Computers & Security,* vol. 31, pp. 643-652, 2012.

[8] V. Asthana, K. Beckers, M. Ifland, J. Martin, N. Ozmore, I. Tarandach*, et al.*, "Software: Security Takes a Champion," http://safecode.org/wp-content/uploads/2019/02/Security-Champions-2019-.pdf 2019.

[9] A. Antukh. (2017). *Security Champions Playbook*. Available: https://www.owasp.org/index.php/Security_Champions_Playbook

[10] M. Thompson and H. Takabi, "EFFECTIVENESS OF USING CARD GAMES TO TEACH THREAT MODELING FOR SECURE WEB APPLICATION DEVELOPMENTS," *Issues in Information Systems,* vol. 17, 2016.

[11] A. Shostack, "Elevation of privilege: Drawing developers into threat modeling," in *2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[12] L. Williams, M. Gegick, and A. Meneely, "Protection poker: Structuring software security risk assessment and knowledge transfer," in *International Symposium on Engineering Secure Software and Systems*, 2009, pp. 122-134.

[13] J. Bosch, "Speed, Data, and Ecosystems: The Future of Software Engineering," *Software, IEEE,* vol. 33, pp. 82-88, 2016.

Biographies

Inger Anne Tøndel received her MSc from the Norwegian University of Science and Technology (NTNU) in 2004. Since then she has been a research scientist at SINTEF Digital. Currently she is also a PhD candidate at NTNU. She has published more than 50 papers in peer reviewed journals and conferences. Her research interests include software security, security requirements, information security risk management, cyber insurance and smart grid cyber security.

Martin Gilje Jaatun is a Senior Scientist at SINTEF Digital in Trondheim, Norway. He graduated from the Norwegian Institute of Technology (NTH) in 1992, and received the Dr.Philos degree in critical information infrastructure security from the University of Stavanger in 2015, where he is now an adjunct professor. Previous positions include scientist at the Norwegian Defence Research Establishment (FFI), and Senior Lecturer in information security at the Bodø Graduate School of Business. His research interests include software security, security in cloud computing, and security of critical information infrastructures. He is vice chairman of the Cloud Computing Association (cloudcom.org), vice chair of the IEEE Technical Committee on Cloud Computing (TCCLD), an IEEE Cybersecurity Ambassador, and a Senior Member of the IEEE.

Daniela Soares Cruzes is a senior scientist at SINTEF Digital. Previously, she was adjunct associate professor at the Norwegian University of Science and Technology (NTNU). She also worked as a research fellow at the University of Maryland and Fraunhofer Center for Experimental Software Engineering-Maryland. Dr. Daniela Cruzes is the project manager of the SoS-Agile (Science of Security for Agile software Development) project funded by the Research Council of Norway (2015-2020). Her interests are agile software development, software security, software testing processes, empirical research methods, theory development and synthesis of software engineering studies.