International Journal of Secure Software Engineering
Volume 8 • Issue 4 • October-December 2017

Risk Centric Activities in Secure Software Development in Public Organisations

Inger Anne Tøndel, Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway & SINTEF Digital, Trondheim, Norway

Martin Gilje Jaatun, SINTEF Digital, Trondheim, Norway

Daniela Soares Cruzes, SINTEF Digital, Trondheim, Norway

Nils Brede Moe, SINTEF Digital, Trondheim, Norway

ABSTRACT

When working with software security in a risk-centric way, development projects become equipped to make decisions on how much security to include and what type of security pays off. This article presents the results of a study made among 23 public organisations, mapping their risk-centric activities and practices, and challenges for implementing them. The authors found that their software security practices were not based on an assessment of software security risks, but rather driven by compliance. Additionally, their practices could in many cases be characterised as arbitrary, late and error driven, with limited follow up on any security issues throughout their software development projects. Based on the results of the study, the authors identified the need for improvements in three main areas: responsibilities and stakeholder cooperation; risk perception and competence; and, practical ways of doing risk analysis in agile projects.

KEYWORDS

Agile Development, Empirical Study, Public Organisations, Risk Analysis, Risk Centric Activities, Risk Communication, Risk Management, Software Security

1. INTRODUCTION

Today, nearly all sectors of society depend on software systems to operate efficiently. As the dependency on software has grown, so have the threats towards these systems and the potential consequences of incidents. Though network security measures (such as firewalls and anti-virus software) can improve the security of the software systems, these only address the symptoms of the real problem: software that is crippled with vulnerabilities (McGraw, 2006).

Building security into the software, through adopting software security activities and measures in the development process, is a direct and effective way of dealing with cyber threats towards software systems. This, however, adds to the development time and cost, and this addition needs to be justified. Working towards 100% secure systems is not feasible, thus it is necessary to identify which part of the software is more critical regarding security and which activities will be most efficient and effective in

DOI: 10.4018/IJSSE.2017100101

Copyright © 2017, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

securing the software product. Taking a risk centric approach to software security means to identify what are the major risks of the particular software that is developed, and use this knowledge of risk to guide decisions regarding software security. This is commonly recommended by current secure Software Development Lifecycles (SDLs), frameworks and maturity models (Chandra, 2008; Howard & Lipner, 2006; McGraw, 2006; McGraw et al., 2016).

In many ways, security can be considered to be in conflict with the current trend of "continuous development" (Fitzgerald & Stol, 2017), reducing efficiency by delaying delivery of new features (at least in the shorter term, though costs may be saved through having to provide fewer fixes later). Agile software development uses an iterative approach to software construction, aimed at reducing development time, and prioritising value, while improving software quality and inherently reducing risk (Cockburn and Highsmith 2001). It is clear that people issues are the most critical in agile projects and that these must be addressed if agile is to be implemented successfully (Cockburn and Highsmith 2001). Even though agile methods claim to be risk driven (Beck, 2000; Eclipse, 2016), some authors have observed that risk management has been neglected in project management of agile projects (Hijazi et al., 2012; Ibbs & Kwak, 2000; Junior et al., 2012; Raz et al., 2002). It may be more difficult to establish a working process for software security activities in agile development compared to waterfall-based development, where you could more easily have mandatory or recommended security activities for the different software development phases (ben Othmane et al., 2014; Jaatun et al., 2015; Microsoft, 2009). Oyetoyan et al. (2017) provide a brief overview of secure SDLs and conclude that traditional approaches to software security do not necessarily work well with agile development processes. Additionally, security is largely a systemic property, and with agile development it can be more of a challenge to have a complete view of the final system (ben Othmane et al., 2014). At the same time, agile development may come with some opportunities regarding security, e.g. to adapt to new security threats and to have ongoing interaction with customers about security.

Risk centric software security is very much related to the way developers address security in the projects. Still, other roles in an organisation (e.g. procurers, legal experts and information security experts) can have major influences on a development project's approach to security and can have important parts to play when it comes to identifying and understanding risk, and in making risk-based decisions in the projects. About ten years ago, van Wyk and McGraw (2005) pointed out the important role of security experts in influencing and supporting the work on security in development projects. There has however not been much research on the interaction between security experts and development projects in agile development since then.

In this article, we address the following research question: How can current software organisations work with software security in a risk centric way? As implied by this research question, we study software security within development practices that are in major adoption today, meaning our context is agile development. However, whereas agile methods are centred on the activities of teams, we take a more holistic approach, including the perspectives of organisations and projects. To answer the research question, we make a mapping of the risk centric activities, practices and challenges for implementing them among 23 public organisations in Norway. This sector has been chosen for study for three reasons. First, this sector has experienced a strong security push from the authorities, causing them to prioritise security management in the organisations. As a consequence, the importance of having someone being responsible for security has been emphasised, something that makes this sector an interesting case to study when it comes to organisational influences on risk centric software security. Second, this sector's access to legal experts makes them aware of legal requirements on security, something that increases the likelihood that software security is given some attention in software development. Third, we had easy access to this sector through cooperation with the Norwegian Agency for Public Management and eGovernment (Difi). The organisations studied have adopted agile practices for software development for some time. In the organisations, we have talked mainly with information security people, as these are in general given broad responsibility for all issues regarding IT security, including software. To ensure that the development project perspective is included, two projects have been studied in more detail from the viewpoint of the software architects.

The article is structured as follows. Section 2 gives an overview of current research on risk management in agile development. Section 3 describes the research method used in the study. Section 4 presents the results of the study, whereas the implications of these results are discussed in Section 5, with an emphasis on making recommendations for research and practice. The threats to validity are also discussed. Section 6 concludes the paper.

2. CURRENT STATE OF THE ART ON RISK CENTRIC SOFTWARE SECURITY

In this section, we start with explaining what type of activities we would expect to see in software development if a risk management approach is taken to the software security work. Then we move on to present current experiences on how risk management fits with the agile approach to software development.

2.1. Software Security Risk Management

Software projects come with many uncertainties, including time-to market, stakeholder expectations and budget (Islam et al., 2014). Such uncertainties lead to project risks. Software risk management is a tool that can be used to manage and reason about these risks in a structured manner. Islam et al. points out that despite the existence of several risk management methods particularly suited for software projects, current research on software risk management shows that these are not well applied. Practitioners' concern is the tangible development cost that lead to project deliverables and thus direct benefits. The impact of applying an overall risk management method on a software project is unclear (Bannerman, 2008).

Security risks are one type of risk that software products face today. Within the area of cyber security, there exist many standards, guidelines and research papers that suggest different ways of managing risk and performing risk assessments. Some of the major ones are ISO/IEC 27005 (ISO/IEC, 2011), OCTAVE Allegro (Caralli et al., 2007), and the NIST Risk Management Framework (RMF) (NIST, 2010). ISO/IEC 27005 and OCTAVE Allegro are concerned with information security risk management and take an organisational approach. RMF specifies a process for integrating organisational risk management activities into system development. These documents claim that a systematic approach to information security risk management is necessary for ensuring that security activities are aligned with the organisational goals and objective, and that organisational needs for security are identified and addressed in an effective and timely manner.

If looking more closely at the type of activities recommended in ISO/IEC 27005, OCTAVE Allegro and RMF, one can identify common activities that one could expect to see when working risk centric in an organisational context. The most obvious activity is that of performing a risk analysis, including setting the scope of the analysis. Recommendations for how to do risk analysis constitute a large part of these documents (e.g. as in OCTAVE Allegro where steps 1-7 of 8 could be categorised as being part of the risk analysis activity). A risk analysis process naturally supports another activity found in all these three documents, namely making decisions on how to treat the risk. These two risk management activities (analysing risk and making decisions on what to do with the risk) can be identified in ISO/IEC 27005, OCTAVE Allegro and RMF by looking at the steps they recommend. By closer reading, one can however identify two other main activities: follow up of risk, and communication of risk. ISO/IEC 27005 contains process steps for "risk monitoring and review" and for "risk communication and consultation". Though not specific steps in the methodology, OCTAVE Allegro includes senior management sponsorship and training as important preparatory activities. RMF, which is concerned with integration of risk management and development, moves beyond just analysing the risk, by describing implementing, assessing and monitoring controls.

Communicating essential risk information to senior management is emphasised in RMF. To sum up, key activities of risk management are:

- **Risk Analysis (RA):** Including characterising the system and the risk appetite, identifying threats towards the system and assess the associated risk;
- **Risk Treatment Decisions (RTD):** Leading to requirements on controls and/or security activities needed;
- **Risk Treatment Follow Up (RTFU):** Intended to assess whether the treatments are implemented and work as intended, and to monitor changes in risks;
- **Risk Communication (RC):** Towards senior managers, but also to make sure information relevant for risk understanding (e.g. legal requirements, changes in threats) are shared between projects and between organisational units.

Risk analysis activities in software security are motivated by similar arguments as in ISO/IEC 27005, Octave Allegro and RMF; to more effectively and less expensively identify security vulnerabilities and risks and establish mitigations (Howard & Lipner, 2006), and to make better trade-off-decisions and prioritise development efforts based on risk (Chandra, 2008; McGraw, 2006). In addition, the awareness raising among project teams (Chandra, 2008) is considered important, especially when it comes to improved understanding of what factors may lead to negative outcomes (Chandra, 2008) and the ability to think like an attacker (McGraw et al., 2016). Threat modelling is even stated to be *"The Cornerstone of the [Security Development Lifecycle]*, and the threat model *"the major [Security Development Lifecycle] artefact"* that *"must be used as a baseline for the product"* (Microsoft, 2009). As such, major software security frameworks, maturity models and secure SDLs include activities very much related to risk management. Table 1 shows how the Building Security In Maturity Model (BSIMM) (McGraw et al., 2016), the OWASP Software Assurance Maturity Model (OpenSAMM) (Deleersnyder et al., 2017), the seven touchpoints for software security (McGraw, 2004) and Microsoft's secure SDL (Howard & Lipner, 2006) all contain activities that are related to the four key risk management activities described above.

Understanding and assessing security risk is known to be a complex challenge. A risk-based approach usually implies having an overview of the criticality of the various software assets, understanding potential threats and vulnerabilities (also from an attacker perspective) and being able to provide estimates on likelihood and consequences of the different types of incidents that can harm the software and impact the service the software delivers to its users. It is also necessary to understand how the various risks can be mitigated effectively. Risk analysis, and especially quantitative risk analysis, has been characterised by some as *"a modern fairytale"* in the domain of information security, as there is no overview of all threats and not sufficient data to estimate probability and consequences. To compound the challenge, there is typically not adequate method support (Oppliger, 2015). If this is the case for information security, it is likely also the case (and maybe even more so) for software security. There is limited empirical data available on what makes risk management difficult, both for information security and software security. A review of risk analysis methods for IT systems (Sulaman et al., 2013) identified a lack of evaluation of risk analysis methods. Despite the mantra that all security work should be risk based, a study among information security professionals (Jourdan et al., 2010) unveiled that as many as 25% stated that risk analysis was never or rarely performed for their department or organisation. A main challenge that has been identified regarding information security risk assessments is the estimation of likelihood and cost of information security risks, due in part to limited historical data available and constantly changing risk factors (Cybenko, 2006; Fenz & Ekelhart, 2010; Gerber & Von Solms, 2005; Rhee et al., 2012; Tøndel et al., 2015). Information is an intangible asset where it is "extremely difficult if not impossible to determine precise value" (Gerber & Von Solms, 2005), and many losses are never discovered and reported (Rhee et al., 2012).

Table 1. Overview of how the key risk management activities are included in key software security frameworks, maturity models and secure SDLs

	BSIMM	OpenSAMM	Touchpoints	Microsoft SSDL
Risk analysis	-Attack Models (Intelligence) -Architecture Analysis (SSDL Touchpoints)	-Threat Assessment (Construction)	-Abuse cases -Risk analysis	-Perform Security and Privacy Risk Assessments (requirements) -Perform Attack Surface Analysis/ Reduction (design) -Use Threat Modeling (design)
Risk treatment decisions	-Standards and Requirements (Intelligence)	-Security Requirements (Construction) -Secure Architecture (Construction)		-Establish security and privacy requirements (requirements) -Establish design requirements (design)
Risk treatment follow up	-Strategy and Metrics (Governance) -Code Review (SSDL Touchpoints) -Security Testing (SSDL Touchpoints) -Penetration Testing (Deployment)	-Strategy & Metrics (Governance) -Design Review (Verification) -Code Review (Verification) -Security Testing (Verification)	-Risk based security tests -Static analysis (tools) -Penetration testing -External analysis	-Perform static analysis (implementation) -Perform dynamic analysis (verification) -Perform Fuzz Testing (verification) -Conduct Attack Surface Review (verification) -Conduct final security review (release)
Risk communication	-Training (Governance)	-Education and Guidance (Governance)		-Core security training (training)

2.2. Software Security Risk Management in Agile Development

Risk management can be said to be treated implicitly in agile development projects (Odzaly et al., 2017; Tavares et al., 2017). As explained by Nelson et al. (2008), one such implicit risk management technique is to prioritise tasks in the beginning of the iteration. As such, high-risk tasks can be prioritised, something that can reduce overall project risk. But as Nelson et al. point out, risk management is broader than prioritising high-risk tasks. There is a need to follow up on the risk and take additional actions if necessary.

As the guidance provided by agile methods when it comes to risk management can be said to be "very general" (Nyfjord & Kajko-Mattsson, 2008), research has aimed to tailor risk management to agile development projects, with several approaches being suggested (Odzaly et al., 2017). Few studies have however covered integration of risk management with agile software, considering the organisational level (Nyfjord & Kajko-Mattsson, 2008; Odzaly et al., 2017).

On security risk management for agile projects, the research papers are even fewer. There are few studies on practices, and few risk analysis methods specifically tailored towards agile development. One of the more relevant studies available is an action research study at Ericsson, (Baca et al., 2015) of the effects of implementing a security enhanced agile software development process (SEAP). This process included several software security activities (e.g. code review, penetration testing), however the study focuses solely on two key aspects of SEAP: Adding more security resources in the project and the development teams, and performing incremental risk analysis. The introduction of SEAP was found to improve identification and handling of risk, and because of this, the risk-management is found to be more cost-efficient than with the approach previously used by Ericsson. This is claimed to be due to security issues now being dealt with in a more distributed fashion, and thus more issues are solved directly by the team. The details of how risk analysis and risk management was with SEAP is not available, apart from describing that frequency of risk analysis was being increased, the scope for each analysis was being reduced, and the approach was becoming more distributed.

The most notable risk analysis method available that is specifically tailored to agile development is Protection Poker (Williams et al., 2010), a game for risk assessment to be used by agile teams. Evaluations of adoption of Protection Poker in real development projects is however sparse. There is one study available where Protection Poker was used by one team at RedHat (Williams et al., 2010), but this study focused more on awareness and knowledge raising through using the techniques, and not on adoption. And despite positive evaluation results, the team studied stopped using Protection Poker sometime after the study was ended.

We are not aware of any studies or specific methods that aim to understand or solve software security risk management in agile development, taking an organisational approach. Such research is however much needed, as effects of security interventions in agile development have been found to be dependent on organisational factors (Poller et al., 2017).

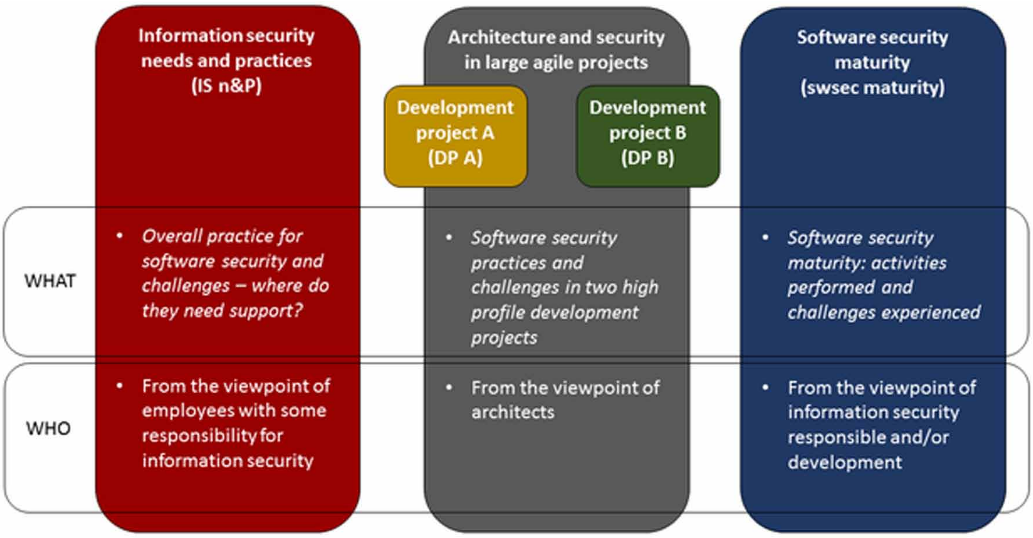
3. RESEARCH METHOD

This section gives an overview of the research method used for the study. It starts with describing the study goal and study design, explaining how the study consists of three separate sub-studies. Then it moves on to explaining the sub-studies in more detail, before describing the approach to analysis.

3.1. Research Questions and Study Design

This study is a combination of three individual studies (sub-studies) that have been performed over the span of two years and that address a common theme. Overall, the study is motivated by the vital role a risk centric approach is considered to have in the literature when it comes to achieving cost effective software security and it tackles the research question "How can current software organisations work with software security in a risk centric way?" Figure 1 shows how the three individual sub-studies

Figure 1. Overview of the individual sub-studies


Information security needs and practices (IS n&P)
Architecture and security in large agile projects
Development project A (DP A)
Development project B (DP B)
Software security maturity (swsec maturity)
WHAT
• Overall practice for software security and challenges – where do they need support?
• Software security practices and challenges in two high profile development projects
• Software security maturity: activities performed and challenges experienced
WHO
• From the viewpoint of employees with some responsibility for information security
• From the viewpoint of architects
• From the viewpoint of information security responsible and/or development

together address this research question from three different angles. The first sub-study studied software security as part of the overall information security management practices in the organisations, from the viewpoint of people working on information security. By having this as the first sub-study we were able to get a high-level view of the overall challenges and status on an organisational level. However, this sub-study deliberately ignored the developer viewpoint, and thus missed a central perspective on the topic studied. Thus, in the second sub-study we studied two high-profile public development projects in more detail, aiming to get an understanding of how security was handled in the projects. Together, sub-study one and sub-study two gave an overview of practices and challenges related to software security from both the viewpoint of security people and software architects in the development projects but lacked a more structured overview of the software security activities adopted in the organisations. As such, sub-study three aimed to get such an overview by mapping their software security activities to those described in the BSIMM framework (McGraw et al., 2016). The three sub-studies together made us able to identify and map risk centric activities and practices in the organisations and understand challenges of implementing a risk centric approach to software security.

As explained in the introduction, the decision to study development in the Norwegian public sector was made based on three factors: a security push in this sector had forced them to prioritise security management in the organisations, legal expertise in the organisations made them aware of any compliance requirements on security, and we had easy access to study participants. The Agency for Public Management and eGovernment (Difi) was our partner in two of the sub-studies (sub-study 1 and 3), contributing with financing as well as helping in participant recruitment. This made it possible to get access to a high number of organisations within a sector where security was receiving growing attention.

Table 2 gives an overview of key facts about the sub-studies. As can be seen the full study has been performed over a period of two years. As some organisations participated in more than one sub-study, the total number of public organisations studied is 23, in addition to two software companies (consultants/contractors) that had a central role in public software development projects.

Table 2. Key facts about the sub-studies

	Information Security Needs and Practices (IS n&p)	**Architecture and Security in Large Agile Projects (DPA, DPB)**	**Software Security Maturity (swsec Maturity)**
When	2013	2013	2015
Public organisations	13	1	20
Software companies (consultants/contractors)	-	2	For some organisations, consultant developers gave input to the questionnaire
Data collection method	Focus group interviews	Group interviews and documentation	Questionnaire with follow up interview
Study participants	Information security/network security	Architects	Some information security, some development
Level of focus	Organisation	Project	Activities

3.2. Sub-Study 1: Information Security Needs and Practices (IS n&p)

In the first sub-study, the aim of the study was to identify public organisations' needs for support regarding information security, and the study covered several topics of which software security was one. The study was performed using the focus group technique (Stewart & Shamdasani, 2014). A focus group can be understood as a semi-structured group interview, and the technique is well suited to identify areas of improvement based on the experience of the participants. By bringing the participants together, rather than performing individual interviews, the participants relate to the opinions of the others in a conversation, something that brings out more information.

Invitations to participate in the study were sent to 26 public organisations, out of which 13 agreed to participate. The invitations to participate were sent by Difi, who also selected which organisations to invite. The main criterion used in the selection was that the participating organisations should include a mixture of organisations regarding both size and security maturity. In addition, a few organisations that were known to have received critical remarks from the Office of the Auditor General of Norway (Riksrevisjonen) regarding information security were invited to participate. The invitations requested the participation of personnel that had some degree of responsibility for information security in their organisations.

In total, three focus group interviews related to software security were performed, and each focus group interview lasted in total three hours. One group consisted of organisations that were believed to be mature in their information security work. The other two groups were more mixed in terms of participants. Few of the participants knew each other from before, and we used brainstorming techniques in the start of each focus group to build trust among the participants, as a successful focus group is dependent on participants that are willing to share experiences.

All groups followed the same process and interview guide. After a short introduction to the study, we performed a short brainstorming where participants addressed the following question: What works well and what is challenging in the work with information security in your organisation? Then we started on the group interview, covering the following topics: 1) security culture and management buy-in 2) information security management systems and risk management, and 3) software development. In the software development part, the participants were asked about when in the process (requirements, development, deployment) information security people and activities were involved, and challenges were discussed. In addition to the questions from the interviews, the later focus groups were presented with important findings from previous groups and asked to comment on those findings.

Two researchers facilitated all the focus groups. One of these was responsible for taking detailed notes from the discussion. In addition, all conversations were recorded. After each focus group, the researchers reflected about how the groups functioned; whether everybody participated in the discussion, if they agreed a lot or disagreed. Observations regarding group dynamics have been taken into account in the analysis of the results, in addition to other context information regarding background and experiences of participants and maturity of their organisations. After each focus group, a summary was made and sent to the participants for comments. The summary included a recapitulation of the most interesting points of the discussion, as well as unstructured and anonymised notes from the discussion.

3.3. Sub-Study 2: Architecture and Security in Large Agile Development Projects (DP A and DP B)

In the second sub-study, one very large agile project was selected as the primary case to study. In the following we call this project "Development project A". This project ran for four years and consumed roughly 800 000 man-hours and was among the largest agile software projects in Norway at that time. In total three organisations were involved in the development, this was the public organisation itself and two contractor organisations. For more details on this case and the study design see (Dingsøyr et al., 2017).

In this sub-study, we performed interviews on how architecture and security was handled in the project. We organised three group interviews, one for each organisation. The reason for having only one organisation in each interview was that we wanted to capture potential differences between the organisations. Participants to the group interviews were recruited to the study by asking the involved organisations to invite the most relevant people for the topic. For the group interviews on software architecture and security only one of the contractors was able to participate with personnel that had worked on the Development project A, resulting in two group interviews on this topic. In total six persons from the project participated in these two interviews; three persons from the public organisation and three from one of the contractors. All these had served in software architect roles at some time during the project, ranging from team architect to chief architect. The interviews were performed by two researchers, recorded and transcribed.

The contractor that could not participate with personnel from Development project A, instead participated in the study with one experienced software architect that was responsible for another high-profile public development project. In the following we call this project "Development project B". This agile project was considerably smaller than Development project A, with only two teams and one contractor, and one year of development. However, this project had a considerably higher focus on security issues, due to the sensitivity of the data handled.

3.4. Sub-Study 3: Software Security Maturity (swsec Maturity)

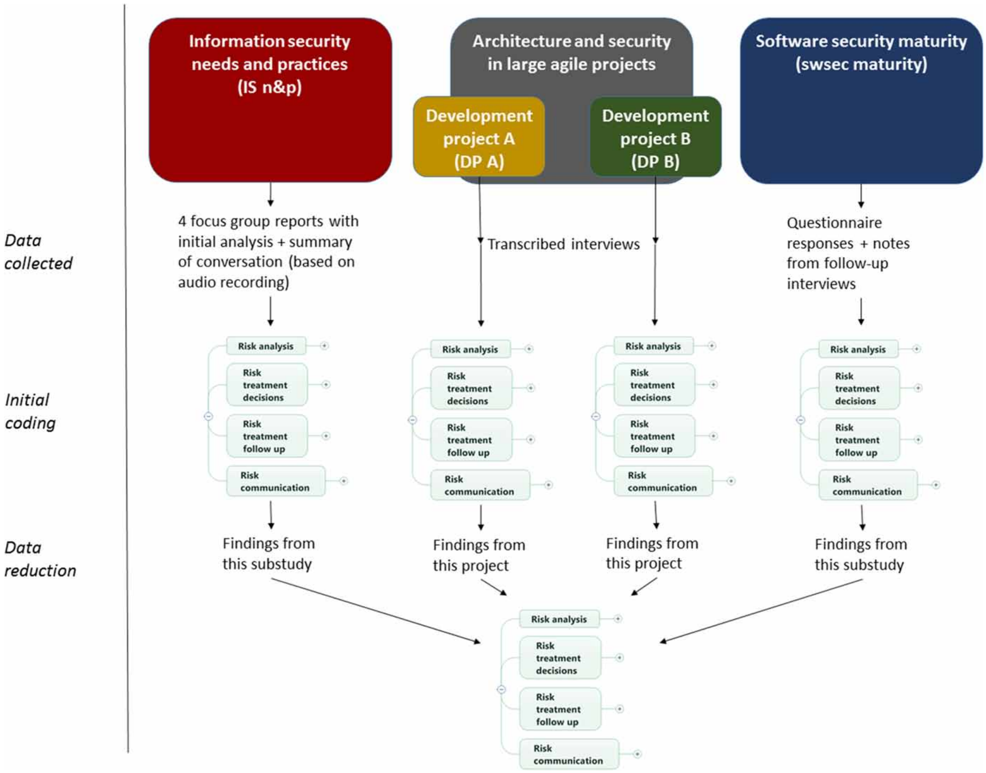
The third sub-study was performed about one and a half years after the other two sub-studies, and was aimed at measuring the software security maturity among Norwegian public organisations (Jaatun et al., 2015). The study instrument consisted of a questionnaire based on the Building Security In Maturity Model (BSIMM) as documented in the BSIMM V report (McGraw et al., 2013). The main function of BSIMM is to serve as a yardstick to determine where an organisation stands compared with other organisations. The questionnaire tells us what activities the organisation has in plac and based on how well they cover the various practices, we can determine the maturity level of each organisation. The questionnaire was followed up by an interview, in order to verify the answers and clarify potential misunderstandings in filling out the questionnaire. In most cases, minor updates to the questionnaire was made based on the interviews. Each interview was performed by one researcher, they were conducted online, and were recorded. Three researchers cooperated on performing the interviews, and these researchers discussed any ambiguities in the questionnaire beforehand to ensure that their assessments of activities were as similar as possible.

The questionnaire was distributed via email to 32 Norwegian public organisations which we had reason to believe had ongoing software development activities. 20 of these organisations returned fully filled-out questionnaires. For seven of the responses, the questionnaire had been filled out in cooperation by representatives involved in software development and in general IT security work. In the other cases, the response was made either by people working on information security or on IT in general (six responses), by people working on software development (five responses), or the main responsibility of the respondent was unclear based on the job title (two responses). In most cases, at least one of the respondents had a managing role in the organisation, e.g., information security manager, IT manager, group leader or architect. The resulting questionnaire responses were analysed to find the maturity, and the results from the analysis of the questionnaire is documented elsewhere (Jaatun et al., 2015). In this paper, we have also analysed the notes from the follow-up interviews, and use these to shed more light on the questionnaire responses and the software security practices reported.

3.5. Analysis

The three studies were aimed at identifying existing practices and challenges when it comes to software security in these organisations. To understand these practices and challenges, we coded the data based on the assumption that risk management is a key to making decisions regarding software security, as depicted in Figure 1. The overall process of analysis is illustrated in Figure 2. For the analysis, we used Mind Manager, with one separate mind map for each study. To ease comparison of data from the different studies, the overall structure of each mind map was the same: risk assessment,

Figure 2. Overview of process for analysing the data


Information security needs and practices (IS n&p)
Architecture and security in large agile projects
Development project A (DP A)
Development project B (DP B)
Software security maturity (swsec maturity)
Data collected
4 focus group reports with initial analysis + summary of conversation (based on audio recording)
Transcribed interviews
Questionnaire responses + notes from follow-up interviews
Initial coding
Risk analysis
Risk treatment decisions
Risk treatment follow up
Risk communication
Data reduction
Findings from this substudy
Findings from this project
Findings from this project
Findings from this substudy

risk treatment decisions, risk monitoring, risk communication. Then the data within each of these overall topics was organised into categories based on the coding. After coding and creating a mind map for each sub-study, key findings from each of the categories and for each study were identified, and these were again used to organise the data into a mind map that included the key findings and data from all the studies, and analyse these together.

4. RESULTS

In this section, we give an overview of practices and experiences from the organisations and projects studied when it comes to risk analysis, risk treatment decisions, risk treatment follow up and risk communication. Then we provide an overview of triggers and barriers not directly related to risk management that seem to be important for software security in the studied organisations.

4.1. Practice Adoption

Table 3 gives an overview of the main findings on adoption of risk centric practices in the studied organisations. The state of adoption is summarised in the findings-column with a number. The meaning of this number differs between the sub-studies. For sub-study 2 (DP A and DP B) the adoption is related to an individual project, and shows if the project adopts the practice throughout the project (2), adopts the practice to some extent (e.g. does it in an ad-hoc manner or partially) (1) or does not adopt the practice (0). For sub-study 1 (IS n&p) and sub-study 3 (swsec mat) the practice adoption relates to organisations, not projects, and the number assigned depends on the practices of the studied organisations overall. Thus, a practice is said to be adopted (2) if most of the organisations (80% or

Table 3. Overview of adoption of risk management activities

Main Findings		**Activity**	**Practice (2 = Adopted; 1 = Sometimes/to Some Extent; 0 = Not Done; '-' = Unclear/No Data)**				**Notes on Adoption**
			IS n&p	**DP A**	**DP B**	**Swsec mat**	
Risk Analysis	Risk analysis practices vary greatly among organisations (RA1) Legal requirements are a driver for performing risk analysis (RA2) Risk analyses are often not centred on software security issues (RA3)	Characterize system	-	1	2	1	No clear process for this. Guided by risk perception where confidentiality has the main priority (less focus on integrity and availability).
		Identify threats/ threat modeling	-	0	-	1	Some organisations are starting to do this more, but not common practice yet.
		Analyse risk	1	0	2	1	Often not relevant (high level, not specific on security risk). Determination of security needs is more driven by compliance than risk.
Risk treatment decisions	Arbitrary, late and error driven (RTD1) No one fights for software security (RTD2) Legal requirements dictate specific security measures – creates tension (RTD3)	Security require-ments	1	1	2	1	In many cases, security requirements are considered rather late in the process.
		Secure design and architecture	-	0	2	1	
Risk treatment follow-up	Most trust the vendors and the developers to follow up security (RTFU1) Limited security testing or review (RTFU2) Time pressure results in security requirements being postponed (or even dropped) (RTFU 3)	Code review	-	0	1	1	When code review is done, its focus is not security, but other code quality aspects.
		Design review	-	0	-	1	
		Security testing	1	0	2	1	Most testing is done on general functionality.
		Metrics	-	0	-	0	
		Monitor changes in risk	-	0	-	1	Organisations monitor changes in network security risks. Monitoring of changes in risks related to sw development is dependent on developer interest.
Risk communication	Lack of training in software security risks (RC1) Silo structure prevents spread of knowledge (RC2)	Training in sw security	0	-	0	1	
		Sharing of risk information within organisation	1	-	-	1	Legal expertise and security expertise in the organization do not necessarily benefit development projects.
		Security people involved in sw projects	1	-	2	1	Some involvement, but in most cases this involvement is very limited in terms of effort.

more) in the sub-study have adopted the practice, and it is partially adopted (1) if more than two of the organisations have adopted the practice. Note that in sub-study 1 we did not ask the organisations about their practices individually, thus practice adoption is based on our impressions from the discussions in the group interviews. Mind maps that show the basis for the findings summarised in the table can be found in Appendix A. In Table 3 we describe the findings in more detail.

Practices and routines for *risk analysis* varied greatly among the studied organisations (RA1). The risk analysis practices were most thoroughly covered by the swsec maturity sub-study (sub-study 3) where we specifically asked each organisation about their practices. In this sub-study, risk analysis was by no means an uncommon practice related to the projects. Only one organisation was clear that they never do such an analysis related to security. However, at the same time only three of the interviewed organisations stated that they always (or most of the time) do risk assessments related to software security in all development projects. In most organisations in this sub-study, security risk analysis was done only for some of the projects, and the organisations did not seem to have any clear strategy for deciding when a risk analysis was needed - the process seemed to be ad hoc and dependent on key people (such as software architects) and their interest in and awareness about security. A few informants claimed that security is more likely to be handled when they procure systems (instead of developing the systems in-house) and for new systems (as opposed to improvement of existing systems), but this appeared to be an observation of own practices rather than an intentional strategy. The IS n&P sub-study (sub-study 1) supports the impression that software architects and legal experts are key to having security needs considered in the projects.

In the swsec maturity sub-study we found that although the organisations self-reported that they do perform risk analysis related to development, such risk analysis did not necessarily cover software security risk in the projects (RA3). Risk analysis performed related to the specific development projects could be centred on other types of project risk, ignoring security risks. Risk analysis performed related to security risks could have been performed on an organisational level but were not considered relevant on the project level. Thus, current risk analysis practices do not necessarily improve software security.

In the organisations studied, the strongest driver we found for performing risk analysis related to security is legal requirements (RA2). In the swsec maturity study (sub-study 3), three of the organisations referred to legal requirements when they talked about their risk assessment practices, and one clearly stated that an audit was the trigger for performing risk assessments. Other motivations are not expressed to the same extent by the interviewees in this sub-study. In the development projects studied (sub-study 2) this effect of legal requirements is shown in practice. Development project B is the only one of the two with clear legal requirements on security, and the only one where risk analysis was done regularly throughout the project. Although development project A was high profile, security risks were not systematically analysed in the project, and the software architects we talked to did not seem to be particularly concerned about, nor updated on, the security of the system.

Risk treatment decisions are often not made based on a process that ensures a thorough understanding of risk. In the organisations studied it seems a bit arbitrary whether or not security is considered for the projects (RTD1). Although some security issues may be considered early on, important security issues may still be left out. Security is in many cases included a bit late and inclusion is error-driven. In the development projects studied this is especially the case in development project A, where security issues that were accidentally discovered was a main reason for the security efforts that the interviewees told us about. Some "surprises" are however hard to avoid, and this was also the case in development project B (where security was considered throughout the project) where the interviewee stated that "…we took some measures towards the end, where it did just strike us 'oh, we just have to secure this.'" In this project, however, most security measures were initiated in a more planned and proactive way. All sub-studies agree that software architects have a potentially important role when it comes to security in the projects, but this is dependent on their personal initiative and interest in security. In practice, few software architects seem to have security as a main interest, and their explicit responsibilities when it comes to security are limited. Security people are sometimes involved, but seem to be passive, either waiting to be invited or participating in the beginning and then leaving the project to fend for itself. Some state that they trust vendors to take care of security, including identifying how much and what type of security is needed.

The responsibility for identifying and deciding on security requirements for the development projects seems fragmented (RTD2). In the swsec maturity study, the interviewee from one organisation pointed to a hired consultant as the one having an interest in software security. An interviewee from a different organisation stated that it is difficult to identify who in the organisation is responsible for software security, and the interviewee considered each developer to be responsible for their part of the code. In the IS n&p substudy, participants pointed out that security is often considered a technical thing that is assumed to be covered since technical people are involved. In the same sub-study, they however explain that they are seeing an improvement in the way security is handled. This is related to people from the business side beginning to get more concerned about security. As they provide input to requirements, they have a potential role in bringing security requirements to the projects, although their competence on security is low. In general, it seems unclear in most organisations where the responsibility of information security people ends and the responsibility of the development part of the organisation starts when it comes to software security.

Legal requirements are an important source for security requirements in the development projects in the organisations studied (RCT3). Though legislation clearly motivates security efforts, there are two main problems with this. First, the legal requirements may have unintended effects when it comes to security. The architect from development project B explained that requirements to physically delete data (not only mark it as deleted) increased the risk that data is lost. Additionally, legal requirements were resulting in more complex solutions, something that is not necessarily beneficial for security. Second, there is a need to balance security and other issues. The architect from development project B stated: "Getting the legal people on board was perceived as one of the greatest challenges…" The same architect characterised security people as extreme, wanting to remove network connections etc. Making compromises is challenging when dealing with legal requirements. However, the architect explained that they sometimes had to go to the legal experts to challenge legal requirements and postpone such requirements to later iterations.

Risk treatment follow up is not done in any structured manner in most of the studied organisations. Activities such as testing and code review are common, but they are rarely considering security aspects (RTFU2). Developers and vendors are trusted to take care of security, without this being followed up by most organisations (RTFU1). Interviewees in the swsec maturity sub-study offer statements such as (paraphrased) I am sure our vendor has routines in place and I have raised this issue with the developers, but I am not sure what developers do about it. In many of the organisations in this sub-study, external vendors do most of the development, something that is mentioned as one hindrance for involvement and follow up by security people. However, one of the more mature organisations in the swsec maturity sub-study was aware that security needs to be followed up on more closely, as expressed by one interviewee (paraphrased): Contract terms state that the vendors should follow the security rules. But it is the daily practice that matters. That is why it is essential to have a process to follow up quality requirements regarding security.

Time pressure and agile development is considered a challenge when it comes to following up on security throughout the development projects (RTFU3). In the IS n&p sub-study, participants explained that "…it was easier when projects were run using the waterfall development model, because then all requirements were there before the project started, and the vendors had to deliver everything in the requirements. Today important decisions that involve information security are made in sprint meetings…" (paraphrased from focus group discussion). Security people do not seem to be present in these meetings, and thus their influence on these decisions is very limited. The swsec maturity sub-study confirmed that existing regimes on software security do not work as well as before due to agile development, further hampering ability to follow up on risk treatment in these projects.

Risk communication is in general not addressed in a structured manner in the studied organisations. The training activities were most thoroughly identified in the swsec maturity sub-study. There it was found that none of the studied organisations have a structured approach to software security training (RC1). They may have general security training for employees, but this does not cover software security. In the swsec maturity sub-study one of the interviewees stated (paraphrased): New employees have to receive a mandatory security introduction and sign the security policy, but there is nothing about software security there. And since the developers are hired consultants none of them has to go through this process. In some organisations, employees have been sent to courses or conferences where software security has been a topic. A few are also aware of training activities at the vendors, but in most cases any software security training is ad hoc and dependent on developer interest.

As has been pointed out above, legislation is an important source for security requirements. Of all the organisations in the swsec maturity sub-study, 85% self-report that they have an overview of regulations. However, in the follow up interviews it becomes clear that although the organisation may have a legal department with a clear overview of legislative requirements, this does not necessarily benefit the development projects. In the same way, security policies may be present at an organisational level, without this affecting the development, and security expertise and follow up on new threats that is done on an organisational level does not necessarily lead to this knowledge being available to development projects (RC2). To illustrate, we paraphrase the following statements from the swsec maturity interviews: We have a forum to discuss cyber-attacks in operation, but not sure if things from this forum get to developers. I hope so. And, our organisation has many policies that ensure we are compliant, but I am not sure how much this impacts the coding.

4.2. Triggers and Barriers for Software Security

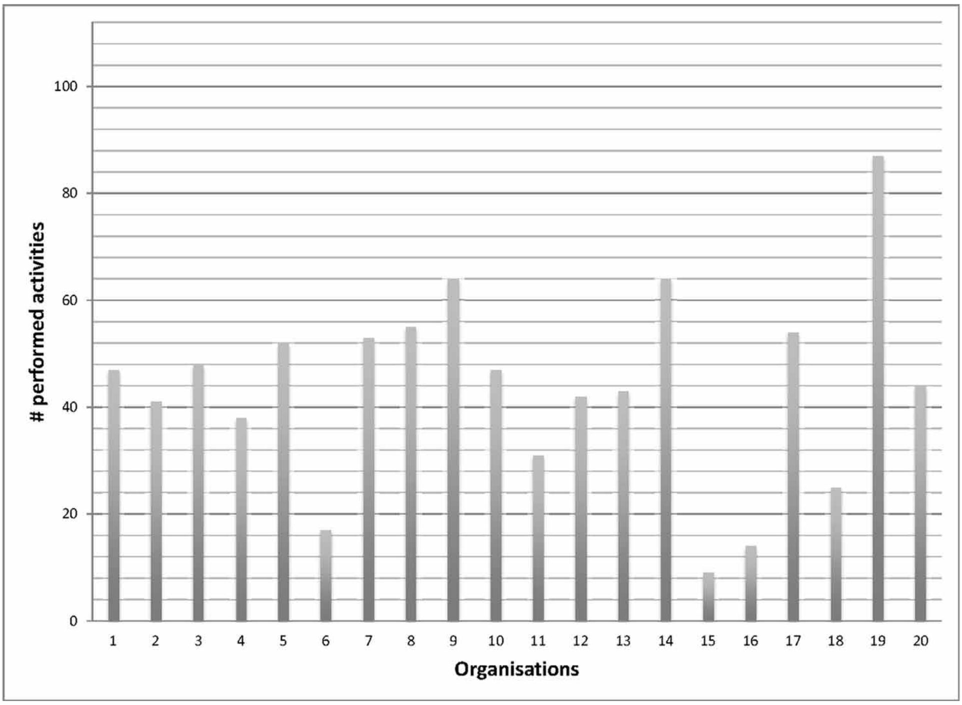
As can be observed from the results described in the above subsection, the results from this study downplay the importance of risk analysis in current software security practices in many of these organisations. It is suggested that any risk analysis performed is often not relevant, and that the motivation is mainly legal compliance and not improved security per se. Additionally, risk analysis is not put forward as an important basis for making decisions about what security measures and activities are needed in the projects. Follow-up of any such decisions is also not risk-based, and very few have any clear processes to review and test for security issues and ensure that any security requirements are adequately dealt with in the final software. As can be seen in Table 3, adoption of typical risk management activities (as described in section 2.1) is low, except in development project B. Although the software security approach of these organisations does not seem to be risk centric, they do report that they perform a number of security activities, as can be seen in Figure 3 that gives an overview of the questionnaire responses in the swsec maturity sub-study (for more details on these responses, see (Jaatun et al., 2015).

As risk management does not seem to be the most important starting point for software security in the studied organisations, we have identified from the data other activities or factors that are stated as possible triggers or barriers for software security by the interviewees (see Table 4). These have been identified by going through the coded data looking for what interviewees describe as causes for or hindrances for software security in their organisations and projects.

As described in the previous sub-section, external requirements clearly represent a trigger for performing risk analysis, and they impact the security requirements of projects as well as the effort put into security. Development project B had quite strict legal requirements related to the data the product was to handle, and this is likely the major reason why this project had a relatively strong focus on security. Additionally, the information security people we talked to in the IS n&p sub-study (sub-study 1) also saw closer interaction with the legal department as a possible way to push more security to the projects.

As also mentioned in the previous sub-section, detection of security-related errors made in code clearly triggers a burst in security attention in order to fix discovered issues. Hearing about security

Figure 3. Self-reported adoption of BSIMM activities in swsec maturity sub-study (questionnaire responses)



problems in previous projects can also increase general attention about security. However, when it comes to issues related to risk perception and attitude and competence on security, awareness seems to be low in the majority of the projects in these organisations. One of the reasons stated for this is that many of the systems they develop are meant only for internal use and will not be directly accessible from the internet. And they state that in cases where they develop open systems, these often only present data that is considered open as well, thus security is not considered important. The exception to this among the projects we have studied is development project B (sub-study 2), where the resulting project was to be accessible from the internet and contained sensitive data, and in this case, security was made a priority. In the study, we have not made any efforts to identify security risks to the software the organisations develop, in order to verify such claims. However, it is worth noting that the security experts in the IS n&p sub-study stated that awareness about the confidentiality aspects of security is much higher than awareness about integrity and availability aspects. The same information security people additionally experienced that knowledge of security was low among many of the developers and procurers, who consider security to mainly be a technical issue that will automatically be covered. Still, the overall study also gives indications that knowledge about security is improving among developers, as some of the people we talked with in the swsec maturity study (sub-study 3) perceived developers as knowledgeable about security. Furthermore, although it seems that interest in security among software architects and developers may be a bit low based on the data we have, it also shows that security can be perceived as something that makes a project interesting and challenging from a technical viewpoint.

In the organisations we have studied, the responsibility for software security seems unclear and fragmented. Few can really claim to have a software security group (SSG), i.e., a group that includes roles such as security architect or security tester. In many organisations, the responsibility for software security is considered part of the responsibility for information security, but in that case, it runs the risk of getting deprecated. Some of the informants were clear that in their organisation other security

Table 4. Triggers (↑) and barriers (↓) for software security activity in the projects

Factor	IS n&p	DP A	DP B	swsec Maturity
External requirements	↑ Legal requirements and Interaction with legal department		↑ Legal requirements	↑ Legal requirements trigger risk analysis
Errors made	↑ Security mishaps, also from previous projects	↑ Security mishaps detected	↑ Security mishaps detected	
Risk perception	↓ Systems not directly connected to the internet	↓ Systems not directly connected to the internet		↑ Project with obvious security needs
	↓	↓		↓ Systems not used by externals ↓ Open data
Responsibility	↓ Unclear responsibilities for sw security		↑ Security included as user stories ↑ Budget for security	↓ Unclear responsibilities for sw security ↓ Rely on contractors to deal with security – limited follow up
Architects	↓ Architects do not take responsibility for security	↓ Architects not that interested in security		↓ Few architects with security interest
Changes	↓ Time pressure ↓ Agile development		↓ Need for progress to fulfil contract ↓ Balancing security with other needs	↓ CISOs do not have resources to follow up on sw security
Attitude and competence	↓ Lack of security knowledge among developers ↓ Procurers lack awareness about security ↓ Security is considered mainly a technical issue	↓ Lack of awareness of security implications	↑ Security was considered part of what made the project technically interesting and challenging	↓ No/little training in sw security
Other issues				↑ New products

activities and goals had been prioritised, and consequently software security had not been given attention. Architects are seen as potential allies in a security push, but as of now they seldom have explicit responsibility for security. By and large, it is currently not possible to clearly hold anyone accountable for software security.

When changes need to be made throughout the project, security is not necessarily considered. More often than not, time pressure in the projects results in security requirements being deprecated in order to reach other project goals. This is considered a bigger problem with agile development than in more traditional waterfall-based development models.

5. DISCUSSION

Based on the results of these studies we have identified three main areas that is important to consider in order to improve current practice and to guide further research. These are: responsibilities and stakeholder cooperation; risk perception and competence, and; practical ways of doing risk analysis.

In the following subsections, these areas are discussed and some recommendations are made. Towards the end of the section we provide a discussion of the validity of the results

5.1. Responsibilities and Stakeholder Cooperation

The empirical basis for the discussion of responsibilities and stakeholder cooperation can be found in Table 5. This table shows the most relevant results from the four risk management areas identified in Section 2, including findings from Table 3 (main findings, practice adoption) and Table 4 (triggers, barriers). In this, and the following, tables, a '-' in one row indicates that there is no relevant finding for this topic. In some cases, rows are merged, and in that case that means that a finding is relevant for more than one risk management area.

One major obstacle for software security in the studied organisations is the unclear responsibilities for software security. Responsibility could be put either on the security experts or on the development projects, but we recommend that responsibility is given to the projects or someone close to development. The reason for this recommendation is twofold. First, when software security is seen as part of information security, it risks being deprecated. This happened in the studied organisations. Due to a push from the government on implementing an Information Security Management System (ISMS), i.e. ISO/IEC 27001, employees having the role of Chief Information Security Officer (CISO) or similar were under pressure to improve their ISMS and relevant practices, while the pressure to improve software security were not as strong. Thus, it is not a big surprise that software security was not given priority by CISOs. As pointed out by van Wyk and McGraw (2005), the alignment of information security and development would require a large effort from information security people. In the organisations we studied, CISOs and other information security employees were already struggling with limited resources and thus unable to properly address all important security tasks. Second, the competence needed to work with information security at an organisational level does not necessarily match the competence needed to work with security in development projects. The BSIMM is clear in its recommendation that the ones responsible for software security should have a development background, stating "Starting with network security people and attempting to teach them about software, compilers, SDLCs, bug tracking, and everything else in the software universe usually fails to produce the desired results. Unfortunately, no amount of traditional security knowledge can overcome a lack of experience building software." (McGraw et al., 2016) Training in security may however be needed for the people that is given this role.

Assigning clear responsibilities for software security does not however remove the need to improve cooperation among a variety of stakeholders related to software security in a project. As can be seen from Table 5, communication and cooperation between the participants in the development project, security people and legal experts is seen as a challenge. This is challenging in two ways: Getting these different roles to interact to have the competence of the legal experts and security experts benefit the development projects, and having these roles understand each other's perspectives so that good trade-offs can be made. This seems to be even more challenging when development is done by vendors/contractors, limiting the interaction. It is important that organisations are aware of both the benefits and challenges of having these roles involved in development projects and make arrangements to support cooperation.

It's been more than ten years since van Wyk and McGraw (van Wyk & McGraw, 2005) called out for aligning information security and software development. They pointed out that the disconnect between security and development led to software development without any understanding of technical security risk, and thus software with security weaknesses that should have been avoided. In their article, van Wyk and McGraw discussed the role of information security in the software security touchpoints and provided recommendations to information security people that wanted to play a part in improving software security. The role of security teams for adoption of secure development tools was further studied by Xiao et al. (2014) as part of interviews with 42 professional software developers. They found that the relationship between the security team and the developers can act

Table 5. Empirical basis for discussion and recommendations on responsibilities and stakeholder cooperation

Area	Main Finding	Relevant Triggers	Relevant Obstacles	Notes on Adoption
Risk analysis	-	-	-	Lack of clear processes
Risk treatment decisions	No one fights for software security (RTD2)	-	Unclear responsibilities for sw security. Architects lack interest in security and do not take responsibility for security.	-
	Legal requirements dictate specific security measures - creates tension (RTD3)	Legal requirements and interaction with legal department	Balancing security with other needs.	-
Risk treatment follow up	Most trust the vendors and the developers (RTFU1)	-	Rely on contractors to deal with security – limited follow up. CISOs do not have resources to follow up on sw security.	Monitoring of changes in risk related to sw development is dependent on developer interest.
	Time pressure results in security requirements being postponed (or even dropped) (RTFU3)	-	Need for progress to fulfil contract Time pressure Agile development	-
Risk communication	Silo structure (RC2)	-	-	Legal expertise and security expertise in the organisations does not necessarily benefit development projects. Limited involvement by security people in the projects.

as a driver for security tool adoption, as well as a hindrance, depending on the circumstances. Only six of 17 security teams interacted often with developers. The other security teams focused more on operational security and was often found to be not helpful in development. When the security team interacted with developers, this caused more security activities to be performed by the developers due to social pressure, and in turn developers felt more responsible for the security of their code. The opposite effect was found in a company where only certain individuals were given training in use of security tools.

Involvement can be done and supported in many ways, e.g. through routines and by creating meeting places. In any case, it is important to ensure the voices of legal experts, security experts, and potentially other types of stakeholders related to security, is heard at key decision points in the projects to ensure that also their concerns are taken into account in the decisions on what risk should be accepted. Trade-offs will have to be made, but these trade-offs should be made based on awareness of the potential consequences of the choices available to the project.

5.2. Risk Perception and Competence

In the organisations studied, it seems clear that confidentiality is what is mainly considered when thinking about the needs for security in the projects. We found little awareness in the data that security is also about integrity and availability. As can be seen from Table 6, many of the obstacles when it comes to software security in the organisations is related to risk awareness; stakeholders have the opinion that security is not needed because the systems are only to be used internally or only contain open data. These are clearly factors that reduce the risk. However, it should not automatically lead to the conclusion that there is no need to consider security for such systems. Some security breaches are performed by insiders (Jang-Jaccard & Nepal, 2014). Some systems that start out as internal, are later wrapped to have a web interface etc. Open data may have requirements when it comes to integrity and availability. In addition, systems that only process open or otherwise insensitive data might still be

Table 6. Empirical basis for discussion and recommendations on risk perception and competence

Area	Main Finding	Relevant Triggers	Relevant Obstacles	Notes on Adoption
Risk analysis *Risk treatment decisions* *Risk treatment follow up*	-	Security considered part of what made the project technically interesting and challenging.	Systems not directly connected to the internet. Systems not used by externals. Open data. Lack of security knowledge among developers Procurers lack awareness about security. Security is mainly considered a technical issue. Lack of awareness of security implications.	-
Risk communication	Lack of training in software security risks (RC1)	-	No/little training in sw security	Training in sw security is almost non-existent in the organisations.

used as a step in an attack, which eventually gains access to other systems in the company. Thus, the organisations need to consider a broader set of security properties in their evaluations of security needs.

To improve security awareness, there is a need to do something about the current lack of training in software security among all stakeholders that are somehow involved in development, it being developers, software architects or procurers. This does not necessarily mean that all developers must attend a software security course (though this will surely have its benefits). Participation in risk analysis is an excellent way of increasing both security knowledge and awareness (Wheeler, 2011). Evaluations of Protection Poker (Williams et al., 2010) show that using a collaborative technique for discussing software security risks in the whole team raises awareness on security and spreads knowledge on security within the team. This corresponds to the experiences reported by Kongsli (Kongsli, 2006) on using misuse stories and automatic testing of security in the development of web applications. Effects experienced included increased security awareness in the team, raised collective ownership of security issues, and moving security activities such as system hardening and penetration testing to earlier iterations. However, to effectively raise awareness while also achieving a risk analysis with good quality, there is a need to include at least one person that is knowledgeable about software security. People with such competence thus need to be made available to the development projects to be used as a resource in security activities throughout the project. This may involve the need to train persons to fulfil this role, if such competence is not already available in the organisations. As also stated above, it is important to note that information security professionals do not automatically fit this role, as they may not have the necessary competence to understand the specifics of software development (van Wyk & McGraw, 2005).

5.3. Practical Ways of Doing Risk Analysis

In the organisations and projects studied, the work on software security does not seem to be risk centric, with a few exceptions (development project B (sub-study 2) and possibly a few of the organisations in the swsec maturity sub-study (sub-study 3)). Instead, organisations take a more compliance-based approach to security, with legal requirements as a main driver for security requirements and risk assessments. A compliance-based approach comes with its benefits, as was also discussed in the IS n&p sub-study (sub-study 1) in the context of organisational security work (not software security). In general, the participants in the focus groups did not agree on what would be most beneficial; a risk based or a compliance-based approach to information security. It was pointed out that a risk-based approach requires more competence in order to be confident that major risks are taken care of. In cases where such competence is lacking, a checklist-based approach or an approach that mainly is based on legal requirements may result in better security. Legal requirements can then be considered a specification of a minimum security level that all organisations have to comply with.

Although a compliance-based approach is easier to apply, a large amount of literature and standards recommend the risk centric approach (Caralli et al., 2007; Chandra, 2008; Howard & Lipner, 2006; ISO/IEC, 2011; McGraw, 2006; McGraw et al., 2016; NIST, 2010; Wheeler, 2011). We would like to point out three main disadvantages with a compliance-based approach to security (based on Wheeler (2011)). First, there is no one-size-fits-all when it comes to security. Different organisations and software have different needs for security. If the security work is solely based on recommendations from checklists, one will likely improve security, but not in the most cost-effective way. In addition, one is not confident that the most important measures for this particular software is addressed sufficiently. Second, security threats are changing fast, thus checklists can be quickly outdated. In order to adequately react to changing threats, there is a need for competence and awareness beyond what you get from checklists. Third, with checklists, those responsible for addressing risks as well as other stakeholders are not trained in considering business needs and the role of security in fulfilling those. Participating in risk analysis is a great way to increase competence and awareness about security. Awareness of business needs is essential to make good decisions on what security measures to take and to ensure security is considered important by managers.

Based on literature and based on the findings from this study we thus recommend moving towards a more risk centric approach to software security. But in doing this the organisations and projects need improved ways to assess risk in the development projects.

Table 7 gives an overview of the main results from the study that we base these recommendations on. The studied organisations' current approaches to software security do not seem to give adequate confidence that software security is addressed sufficiently, as current software security efforts seem largely to be arbitrary, late and error driven (RTD1). Taking into account the limited testing and security review practiced (RTFU2), it is likely that many security issues go undetected with today's

Table 7. Empirical basis for discussion and recommendations on practical ways of doing risk analysis

Area	Main Finding	Relevant Triggers	Relevant Obstacles	Notes on Adoption
Risk analysis	Risk analysis practices vary greatly among organisations (RA1)	Legal requirements New products Project with obvious security needs	-	For risk analysis, see RA1-3. Very low adoption of threat modeling. -
	Legal requirements are a driver for performing risk analysis (RA2)			
	Risk analyses are often not centred on software security issues (RA3)			
Risk treatment decisions	Arbitrary, late and error driven (RTD1)	Detection of errors made	-	Some security requirements are made, but may come late in the process.
Risk treatment follow up	Limited security testing or review (RTFU2)	-	-	-
Risk communication	-	-	-	-

practice. Though errors were a trigger for security activities also in development project B where security was followed up in a more continuous and structured manner throughout the project, the effect of errors as a trigger for security activities seem to have been stronger in development project A where security was not given as much attention. Legal requirements were an important trigger in development project B, but in project B this led to a more risk centric approach to security with adoption of activities that could be considered part of risk management (see Table 3).

Current approaches to risk analysis in the studied organisations seem however to be inadequate for software security (RA1-3). First, it is not clear when risk analysis should be performed. Second, the analyses that are done seem not to be that useful for software security work. Based on this, it seems that the organisations could benefit from lightweight processes for determining the need for security in a project, to determine what level of risk analysis is needed as well as other security activities. Such a lightweight process need to take into account the full security properties, not only confidentiality and legal requirements. Additionally, there is a need for risk analysis methods that fit software projects and that can be done in a more continuous manner throughout the project and at a level useful for development. Protection Poker (Williams et al., 2010) is one example of a risk assessment technique tailored to agile development and that would potentially fit the needs of these organisations, and that has positive evaluation results in a real development setting. However, there are not many other such techniques to choose from. Instead companies are left with adapting more general risk assessment techniques to fit the needs of their agile development projects. More research is needed in how risk assessment can be done efficiently and effectively in agile projects.

5.4. Threats to Validity

The discussion of threats to validity of this study is based on the recommendations of Cruzes and ben Othmane regarding threats to validity in empirical software security research (Cruzes & ben Othmane, 2017). It is important to highlight that qualitative studies such as the one that we performed rarely attempt to make universal generalisations. Instead, they are more concerned with characterising, explaining, and understanding the phenomena in the contexts under study. Cruzes and ben Othmane base their recommendations on Lincoln and Guba (1985), that substituted reliability and validity with the parallel concept of trustworthiness. Trustworthiness again consists of four aspects: credibility, transferability, dependability, and confirmability, with credibility as an analogy to internal validity, and transferability as an analogy to external validity.

Credibility refers to "the quality of being convincing or believable, worthy of trust" (Cruzes & ben Othmane, 2017), and dependability refers to "stability and reliability of data over time and conditions" (Cruzes & ben Othmane, 2017). The credibility and dependability of this study are closely related, and highly linked to study design decisions, in particular decisions regarding scope and depth of the study. In the following we discuss three main design decisions made and their impact on validity, namely the decision to study several organisations at a high level instead of one or a few organisations at a more detailed level, the decision to gather the perspectives of roles outside the development teams, and the decision to have the study organised as three sub-studies spanning two years.

In this study, we have studied 23 organisations, but these have not been studied in detail. We rely on self-reporting of practices, and thus on the people we talk with adequately reporting both practices and challenges. In the focus groups as well as in the interviews we got the impression that the people we talked to were honest about their practices, also telling about challenges they faced. However, we have not aimed to check that what they told us was in fact true using additional empirical sources. Usually, only one person from each organisation was interviewed, thus we only got one individual's perspective on their software security work. We could have chosen to go more in depth in the organisations, including more peoples' perspectives, but this would have come at the cost of the number of organisations we would have the capacity to study. Studying as many as 23 organisations from the same sector makes us able to understand the practices and challenges of this sector as a whole, not only that of individual organisations.

In the IS n&p sub-study (sub-study 1) and in the swsec maturity sub-study (sub-study 3), most of the people we talked to were not deeply involved in development, but rather had roles related to network security or information security in the organisation. One reason for this study design decision is our research question, where we take a more holistic approach to understanding risk centric practices to software security, including also the organisational aspect. This emphasis on the opinions of security people outside of the development projects is however a limitation, as we may risk to not adequately understand the actual practices of the development projects. By studying two development projects in more detail, we overcome some of this limitation. However, it is important to note that by collecting the viewpoints of security people, also outside development, is important because it corresponds to the way software security is currently handled in these organisations. When in the swsec maturity sub-study we asked to talk with those responsible for software security, we were directed to information security people in many cases. If we had decided to only study the projects, we would have missed the important perspective of the security people and their interaction with the projects when it comes to software security.

The study performed in order to do this mapping consists of three sub-studies performed over two years. The study does not aim to identify changes that may have happened during this time. However, we are not aware of any major external factors that would impact the software security work during these two years. The focus in the three sub-studies are not the same but differ in what is studied and who is used as informants. This is a strength in the way that the software security practices in the sector is studied from different angles. However, none of the studies study risk centric activities exclusively. Rather, the overall practices and challenges are aimed captured. As we did not bring with us a list of risk centric activities to look for in the companies, we may have missed some of their risk centric activities. However, studying the organisations strictly through the lens of such a list of risk centric activities could result in overlooking practices that we had not included in the list beforehand, thus obscuring the organisations' practices and approach to software security.

Transferability of study results refers to "the degree to which the results of the qualitative research can be generalized or transferred to other contexts or settings. It depends on the degree of similarity between sending and receiving contexts" (Cruzes & ben Othmane, 2017). In this case, it should not be done without taking into account the particular context of this study; public organisations in Norway. Public organisations may behave differently than private software companies in some respects. Although private companies act as contractors to the studied public organisations, the organisations themselves have different goals than what is common for private companies. The high emphasis on legal compliance found in this study is an example of a factor that can be stronger because of the sector and the type of systems developed. However, with the upcoming enforcement of the General Data Protection Regulation (GDPR) legal compliance will most likely become more important for software companies in general.

Confirmability refers to "neutrality; that is, findings must reflect the participants' voice and conditions of the inquiry, and NOT the researcher's bias, perspective, or motivations" (Cruzes & ben Othmane, 2017). In the three sub-studies, several researchers have been involved in data collection, and no one researcher has taken part in all data collection. This is both a strength and a weakness. By having several researchers involved, any preconceptions of one individual researcher have less impact on the data collection. However, with several researchers involved there is the challenge of coordinating these researchers to ensure the data collection is consistent among the different companies. This was especially important in the swsec maturity sub-study, where phone interviews were done by several researchers individually, and in this sub-study we took measures to ensure the involved researchers had a similar understanding of key concepts used in the interview guide and of the goal of the study. In the other two sub-studies, data collection was done by the same researchers throughout the whole sub-study. Richness of data has been ensured through recording of interviews in sub-studies 1 and 2. In sub-study 2, the data analysed was the transcribed interviews, while in sub-study 2 the recording was used to enrich notes taken during the focus group sessions. These notes, including initial conclusions from the focus group, was sent to the focus group members for comments a short time after the focus group. In sub-study 3, richness of data was ensured by having questionnaires that was followed up by an interview. Besides, we have created mind-maps of the data for abstraction of the results, where all results can be traced back to the original source of information.

6. CONCLUSION

This study of software security practices in public organisations, with a mapping of their risk centric activities, has revealed that software security practices were not mainly based on an assessment of software security risks, but rather driven by compliance. Their practices could also in many cases be characterised as arbitrary, late and error driven, with limited follow up on any security issues throughout the development projects. We have identified a need for: more practical ways of doing risk analysis; improved risk perception and competence; and clearer responsibilities and improved stakeholder cooperation.

We recommend that organisations move towards a more risk centric approach to software security, as the current compliance-based approach does not give adequate confidence that important security issues are addressed sufficiently. In doing this, organisations would benefit from lightweight processes for determining the need for security in a project, making sure the full security properties are considered. Additionally, there is a need for risk analysis methods that fit agile software projects, and that can be done in a more continuous manner. Key stakeholders in the development project should be involved in risk analysis, to increase awareness of security. Responsibility for software security in a project should be clearly assigned, and should preferably be given to someone close to the development. However, in addition to assigning responsibility, there is a need to arrange for legal and security experts to have understanding of and interaction with development projects. When decisions are made that impact risk acceptance, there should be routines in place to make sure experts on legal and security issues have a chance to share their perspectives on the issue.

ACKNOWLEDGMENT

This work has been supported in part by the SoS-Agile: Science of Security in Agile Software Development project, funded by the Research Council of Norway (grant number 247678). Two of the sub-studies (sub-study 1 and 3) have been funded by the Norwegian Agency for Public Management and eGovernment (Difi). We would like to thank all the company and project representatives that participated in the study. We would also like to thank Tor Erlend Fægri and Svein Hallsteinsen who performed the interviews in the second sub-study, and Karin Bernsmed who performed some of the interviews in the third sub-study. Thanks to Prof. Guttorm Sindre for input on the contents of the paper.

REFERENCES

Baca, D., Boldt, M., Carlsson, B., & Jacobsson, A. (2015, August 24-27). A Novel Security-Enhanced Agile Software Development Process Applied in an Industrial Setting. *Paper presented at the 2015 10th International Conference on Availability, Reliability and Security*.

Bannerman, P. L. (2008). Risk and risk management in software projects: A reassessment. *Journal of Systems and Software*, *81*(12), 2118–2133. doi:10.1016/j.jss.2008.03.059

Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley Longman Publishing Co., Inc.

ben Othmane, L., Angin, P., Weffers, H., & Bhargava, B. (2014). Extending the Agile Development Process to Develop Acceptably Secure Software. *IEEE Transactions on Dependable and Secure Computing, 11*(6), 497-509. doi:10.1109/tdsc.2014.2298011

Caralli, R. A., Stevens, J. F., Young, L. R., & Wilson, W. R. (2007). *OCTAVE Allegro: Improving the Information Security Risk Assessment Process (CMU/SEI-2007-TR-012 ESC-TR-2007-012)*. Software Engineering Institute at Carnegie Mellon University.

Chandra, P. (2008). Software assurance maturity model. Retrieved from Cruzes, D. S., & ben Othmane, L. (2017). Threats to Validity in Empirical Software Security Research. In L. ben Othmane, M. G. Jaatun, & E. Weippl (Eds.), Empirical Research for Software Security: Foundations and Experience (pp. 277-302). CRC Press.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133.

Cybenko, G. (2006). Why Johnny Can't Evaluate Security Risk. *IEEE Security and Privacy*, *4*(1), 5. doi:10.1109/MSP.2006.30

Deleersnyder, S., Win, B. D., & Glas, B. (2017). *Software Assurance Maturity Model - How To Guide - A Guide to Building Security Into Software Development*. Retrieved from https://github.com/OWASP/samm/blob/master/v1.5/Final/SAMM_How_To_V1-5_FINAL.pdf

Dingsøyr, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2017). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*. doi:10.1007/s10664-017-9524-2

Eclipse. (2016). Eclipse Process Framework (EPF). Retrieved from http://www.eclipse.org/epf/

Fenz, S., & Ekelhart, A. (2010). Verification, validation, and evaluation in information security risk management. *IEEE Security and Privacy*, (2): 58–65.

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, *123*(Suppl. C), 176–189. doi:10.1016/j.jss.2015.06.063

Gerber, M., & Von Solms, R. (2005). Management of risk in the information age. *Computers & Security*, *24*(1), 16–30. doi:10.1016/j.cose.2004.11.002

Hijazi, H., Khdour, T., & Alarabeyyat, A. (2012). A Review of Risk Management in Different Software Development Methodologies. *International Journal of Computers and Applications*, *45*(7), 8–12.

Howard, M., & Lipner, S. (2006). *The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software* (Vol. 2016). Microsoft Press.

Ibbs, C. W., & Kwak, Y.-H. (2000). Assessing project management maturity. *Project Management Journal*, *31*(1), 32–43.

Islam, S., Mouratidis, H., & Weippl, E. R. (2014). An empirical study on the implementation and evaluation of a goal-driven software development risk management model. *Information and Software Technology*, *56*(2), 117–133. doi:10.1016/j.infsof.2013.06.003

ISO/IEC. (2011). ISO/IEC 27005: 2011Information technology–Security techniques–Information security risk management.

Jaatun, M. G., Cruzes, D. S., Bernsmed, K., Tøndel, I. A., & Røstad, L. (2015). *Software Security Maturity in Public Organisations. In Information Security* (pp. 120–138). Springer.

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80*(5), 973–993. doi:10.1016/j.jcss.2014.02.005

Jourdan, Z., Rainer, R. K., Marshall, T. E., & Ford, F. N. (2010). An Investigation Of Organizational Information Security Risk Analysis. *Journal of Service Science*, *3*(2), 10.

Junior, I. H. F., Azevedo, R. R. d., Moura, H. P. d., & Silva, D. S. M. d. (2012, August 27-30). Elicitation of Communication Inherent Risks in Distributed Software Development. *Paper presented at the 2012 IEEE Seventh International Conference on Global Software Engineering Workshops*. doi:10.1109/ICGSEW.2012.18

Kongsli, V. (2006). Towards agile security in web applications. *Paper presented at the Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry* (Vol. 75). Sage.

McGraw, G. (2004). Software security. *IEEE Security and Privacy*, *2*(2), 80–83. doi:10.1109/MSECP.2004.1281254

McGraw, G. (2006). *Software security: building security in* (Vol. 1). Addison-Wesley Professional.

McGraw, G., Migues, S., & West, J. (2013). *Building Security In Maturity Model (BSIMM-V)*. Retrieved from http://bsimm.com

McGraw, G., Migues, S., & West, J. (2016). *Building Security In Maturity Model (BSIMM7)*. Retrieved from http://bsimm.com

Microsoft. (2009). *Security Development Lifecycle for Agile Development*. Retrieved from http://www.microsoft.com/en-us/SDL/Discover/sdlagile.aspx

Nelson, C. R., Taran, G., & de Lascurain Hinojosa, L. (2008). Explicit Risk Management in Agile Processes. In P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming: 9th International Conference, XP 2008*, *Limerick, Ireland*, *June 10-14* (pp. 190-201). Berlin: Springer.

NIST. (2010). *Guide for Applying the Risk Management Framework to Federal Information Systems - A Security Life Cycle Approach* (Special Publication 800-37). Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r1.pdf

Nyfjord, J., & Kajko-Mattsson, M. (2008). Integrating Risk Management with Software Development: State of Practice. *Paper presented at the International MultiConference of Engineers and Computer Scientists*, Hong Kong. Retrieved from http://www.iaeng.org/publication/IMECS2008/IMECS2008_pp878-884.pdf

Odzaly, E. E., Greer, D., & Stewart, D. (2017). Agile risk management using software agents. *Journal of Ambient Intelligence and Humanized Computing*. doi:10.1007/s12652-017-0488-2

Oppliger, R. (2015). Quantitative Risk Analysis in Information Security Management: A Modern Fairy Tale. *IEEE Security and Privacy*, *13*(6), 18–21. doi:10.1109/MSP.2015.118

Oyetoyan, T. D., Jaatun, M. G., & Cruzes, D. S. (2017). A Lightweight Measurement of Software Security Skills, Usage and Training Needs in Agile Teams. *International Journal of Secure Software Engineering*, *8*(1), 27. doi:10.4018/IJSSE.2017010101

Poller, A., Kocksch, L., Türpe, S., Epp, F. A., & Kinder-Kurlanda, K. (2017). Can Security Become a Routine?: A Study of Organizational Change in an Agile Software Development Group. *Paper presented at the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, Portland, OR. doi:10.1145/2998181.2998191

Raz, T., Shenhar, A. J., & Dvir, D. (2002). Risk management, project success, and technological uncertainty. *R & D Management*, *32*(2), 101–109. doi:10.1111/1467-9310.00243

Rhee, H.-S., Ryu, Y. U., & Kim, C.-T. (2012). Unrealistic optimism on information security management. *Computers & Security*, *31*(2), 221–232. doi:10.1016/j.cose.2011.12.001

Stewart, D. W., & Shamdasani, P. N. (2014). Focus groups []: Sage Publications.]. *Theory into Practice*, *20*.

Sulaman, S. M., Weyns, K., & Höst, M. (2013). A review of research on risk analysis methods for IT systems. *Paper presented at the 17th International Conference on Evaluation and Assessment in Software Engineering*. doi:10.1145/2460999.2461013

Tavares, B. G., da Silva, C. E. S., & de Souza, A. D. (2017). Risk management analysis in Scrum software projects. *International Transactions in Operational Research*. doi:10.1111/itor.12401

Tøndel, I. A., Line, M. B., & Johansen, G. (2015). Assessing information security risks of AMI: What makes it so difficult? *Paper presented at the 1st International Conference on Information Systems Security and Privacy 2015*, Angers, France.

van Wyk, K. R., & McGraw, G. (2005). Bridging the gap between software development and information security. *Security & Privacy, IEEE*, *3*(5), 75–79. doi:10.1109/MSP.2005.118

Wheeler, E. (2011). *Security Risk Management* (1st ed.). Boston: Syngress.

Williams, L., Meneely, A., & Shipley, G. (2010). Protection Poker: The New Software Security "Game". *IEEE Security and Privacy*, *8*(3), 14–20. doi:10.1109/MSP.2010.58

Xiao, S., Witschey, J., & Murphy-Hill, E. (2014). Social influences on secure development tool adoption: why security tools spread. *Paper presented at the Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, Baltimore, MD. doi:10.1145/2531602.2531722

APPENDIX

Mind Maps of Findings

Figures 4-7 provide an overview of the main findings from the study, as well as which sub-studies the findings come from.

Figure 4. Overview of key findings on risk analysis

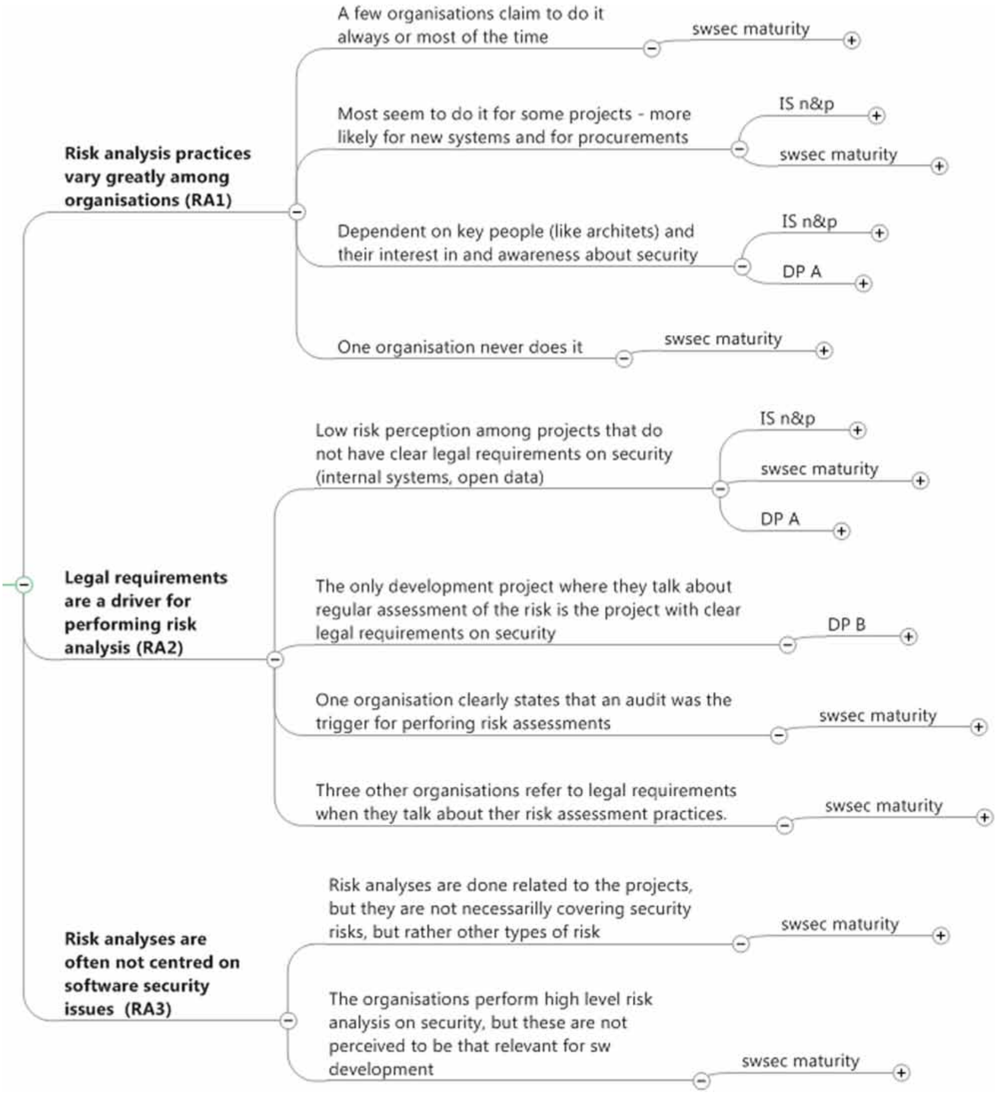

Risk analysis practices vary greatly among organisations (RA1)
A few organisations claim to do it always or most of the time
swsec maturity
Most seem to do it for some projects - more likely for new systems and for procurements
IS n&p
swsec maturity
Dependent on key people (like architets) and their interest in and awareness about security
IS n&p
DP A
One organisation never does it
swsec maturity
Legal requirements are a driver for performing risk analysis (RA2)
Low risk perception among projects that do not have clear legal requirements on security (internal systems, open data)
IS n&p
swsec maturity
DP A
The only development project where they talk about regular assessment of the risk is the project with clear legal requirements on security
DP B
One organisation clearly states that an audit was the trigger for perforing risk assessments
swsec maturity
Three other organisations refer to legal requirements when they talk about ther risk assessment practices.
swsec maturity
Risk analyses are often not centred on software security issues (RA3)
Risk analyses are done related to the projects, but they are not necessarilly covering security risks, but rather other types of risk
swsec maturity
The organisations perform high level risk analysis on security, but these are not perceived to be that relevant for sw development
swsec maturity

Figure 5. Overview of key results on risk treatment decisions

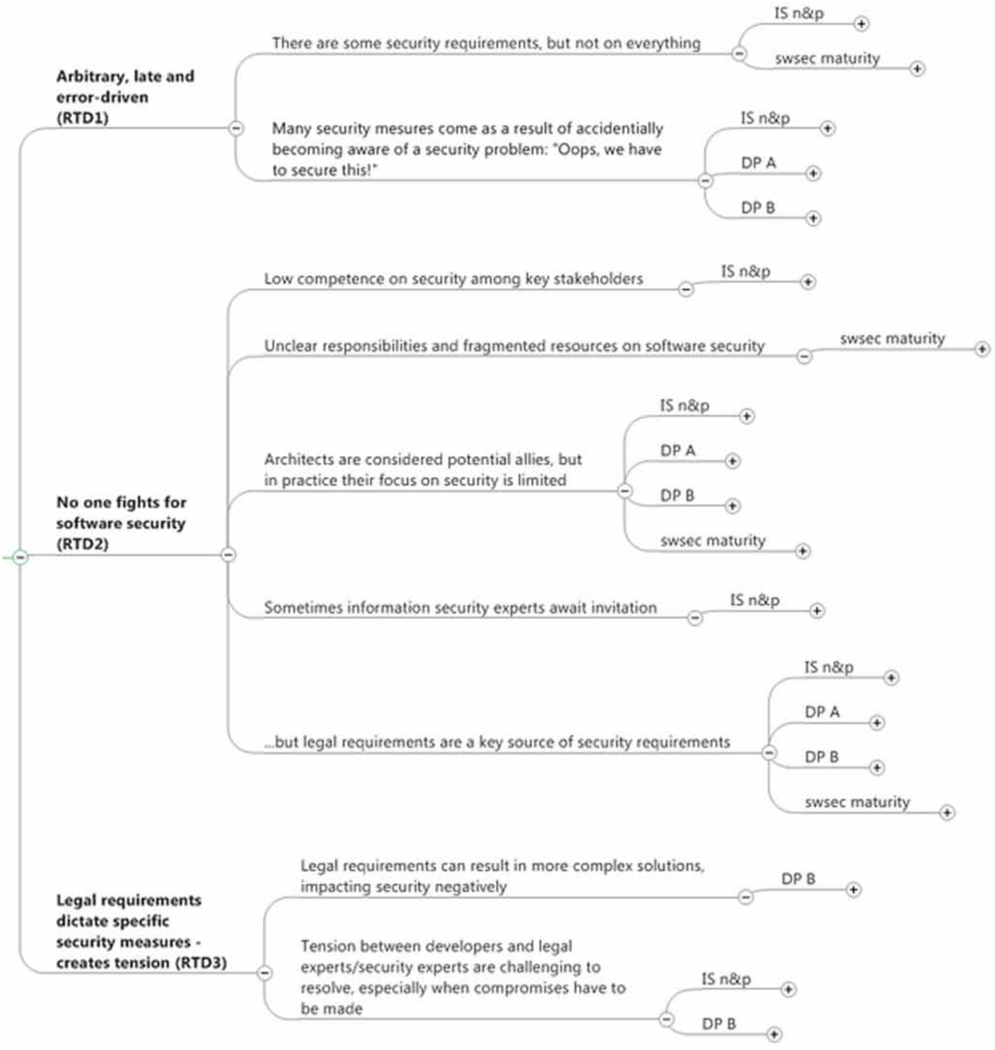

Arbitrary, late and error-driven (RTD1)
There are some security requirements, but not on everything
IS n&p
swsec maturity
Many security mesures come as a result of accidentially becoming aware of a security problem: "Oops, we have to secure this!"
IS n&p
DP A
DP B
No one fights for software security (RTD2)
Low competence on security among key stakeholders
IS n&p
Unclear responsibilities and fragmented resources on software security
swsec maturity
Architects are considered potential allies, but in practice their focus on security is limited
IS n&p
DP A
DP B
swsec maturity
Sometimes information security experts await invitation
IS n&p
...but legal requirements are a key source of security requirements
IS n&p
DP A
DP B
swsec maturity
Legal requirements dictate specific security measures - creates tension (RTD3)
Legal requirements can result in more complex solutions, impacting security negatively
DP B
Tension between developers and legal experts/security experts are challenging to resolve, especially when compromises have to be made
IS n&p
DP B

Figure 6. Overview of key results on risk treatment follow up

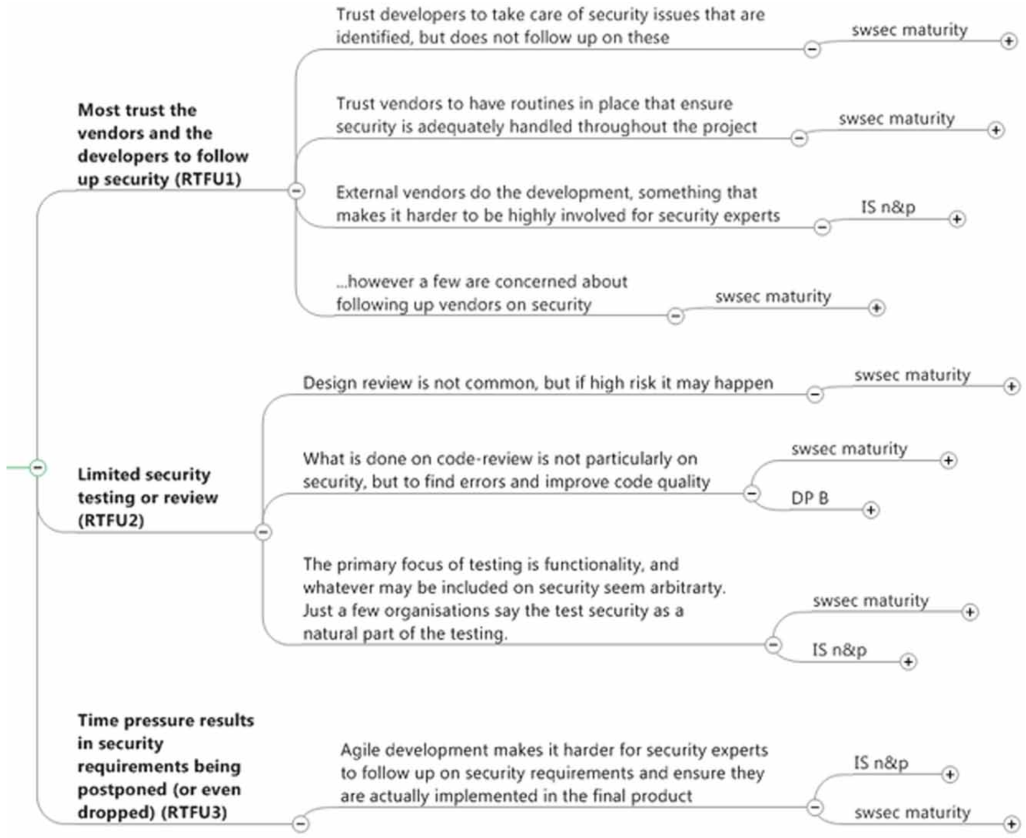
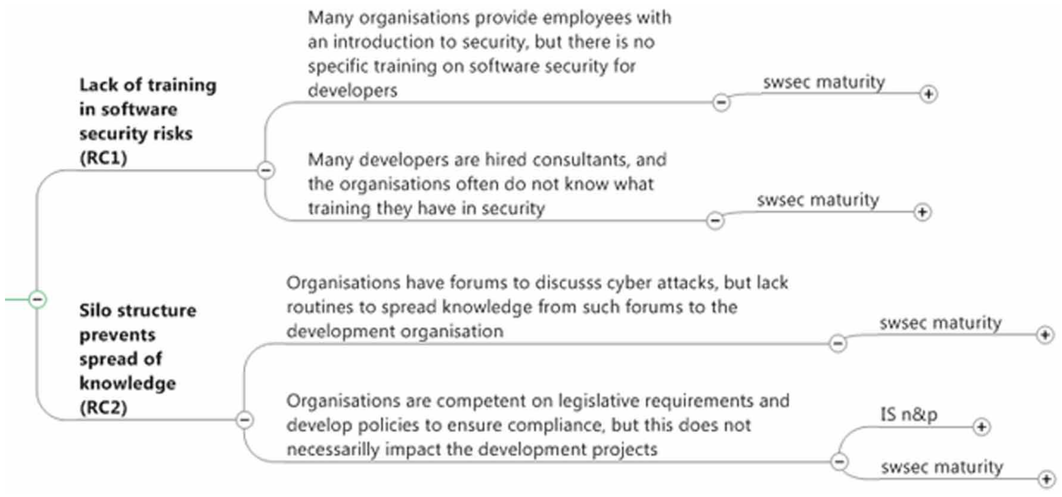

Most trust the vendors and the developers to follow up security (RTFU1)
Trust developers to take care of security issues that are identified, but does not follow up on these
swsec maturity
Trust vendors to have routines in place that ensure security is adequately handled throughout the project
swsec maturity
External vendors do the development, something that makes it harder to be highly involved for security experts
IS n&p
...however a few are concerned about following up vendors on security
swsec maturity
Limited security testing or review (RTFU2)
Design review is not common, but if high risk it may happen
swsec maturity
What is done on code-review is not particularly on security, but to find errors and improve code quality
swsec maturity
DP B
The primary focus of testing is functionality, and whatever may be included on security seem arbitrarty. Just a few organisations say the test security as a natural part of the testing.
swsec maturity
IS n&p
Time pressure results in security requirements being postponed (or even dropped) (RTFU3)
Agile development makes it harder for security experts to follow up on security requirements and ensure they are actually implemented in the final product
IS n&p
swsec maturity

Figure 7. Overview of key results on risk communication


Lack of training in software security risks (RC1)
Many organisations provide employees with an introduction to security, but there is no specific training on software security for developers
swsec maturity
Many developers are hired consultants, and the organisations often do not know what training they have in security
swsec maturity
Silo structure prevents spread of knowledge (RC2)
Organisations have forums to discusss cyber attacks, but lack routines to spread knowledge from such forums to the development organisation
swsec maturity
Organisations are competent on legislative requirements and develop policies to ensure compliance, but this does not necessarily impact the development projects
IS n&p
swsec maturity

Inger Anne Tøndel is a PhD candidate at the Department of Computer Science at the Norwegian University of Science and Technology (NTNU) and a Research Scientist at SINTEF Digital. She got her master's degree in Telematics from NTNU in 2004. Her research interests include software security, information security risk management, cyber insurance and cyber security in smart grids.

Martin Gilje Jaatun is a Senior Scientist at SINTEF Digital and an Adjunct Professor at the University of Stavanger. He graduated from the Norwegian Institute of Technology (NTH) in 1992 and received the Dr.Philos. degree from the University of Stavanger in 2015. 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 IEEE TCCLD, and a Senior Member of the IEEE. He is also an IEEE Cybersecurity ambassador, and Editor-in-Chief of the International Journal of Secure Software Engineering.

Daniela Cruzes is a researcher scientist at SINTEF. Previously, she was adjunct associate professor at the Norwegian University of Science and Technology (NTNU). She worked as a researcher fellow at the University of Maryland and Fraunhofer Center for Experimental Software Engineering-Maryland. Dr. Daniela Cruzes received her PhD in experimental software engineering from the University of Campinas - UNICAMP in Brazil in 2007. Her research interests are empirical software engineering, research methods and theory development, synthesis of SE studies, software security, software testing and agile and DevOps.

Nils Brede Moe works with software process improvement, intellectual capital, and agile and global software development as a senior scientist at SINTEF. His research interests are related to organizational, socio-technical, and global/distributed aspects. His publications include several longitudinal studies on self-management, decision making, innovation, and teamwork. He has co-edited the books Agile Software Development: Current Research and Future Directions and Agility Across Time and Space: Implementing Agile Methods in Global Software Projects. His thesis was, From Improving Processes to Improving Practice - Software Process Improvement in Transition from Plan-driven to Change-driven Development. He holds an adjunct position at the Blekinge Institute of Technology in Sweden.