Knut Formo Buene
Helga Tenold Fridtun

Investigating trust relationships between software development teams and information security stakeholders

Master's thesis in Communication Technology
Supervisor: Maria Bartnes
Co-supervisor: Roy Myhre
June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Kunnskap for en bedre verden

Knut Formo Buene
Helga Tenold Fridtun

Investigating trust relationships between software development teams and information security stakeholders

Master's thesis in Communication Technology
Supervisor: Maria Bartnes
Co-supervisor: Roy Myhre
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Title:	Investigating trust relationships between software development teams and information security stakeholders
Students:	Knut Formo Buene & Helga Tenold Fridtun

Problem description:

The DevOps approach for software development has become widespread in the IT industry and continues to be a popular approach for software development. DevOps emphasises rapid releases with a high number of software iterations. The DevOps approach, stemming from agile development, differs from a traditional waterfall approach, where the development phases are planned out in advance. Pre-planned phases related to security would provide the development team and security practitioners the opportunity for ensuring that sufficient security measures are completed. This is not emphasised in DevOps and has been a cause of concern in the security community.

The fast-paced nature of software development in DevOps has led to traditional security measures being considered unsuitable. If security cannot keep up with the pace of software development in DevOps, there will be software vulnerabilities making their way out into the real world. There can be severe consequences to software vulnerabilities and misconfigurations, and this is well-documented in the media, which reports about data breaches and exploits frequently. Security has to be part of the entirety of the software development cycle. Several frameworks have been proposed, such as DevSecOps, which focuses on integrating security and DevOps and left-shifting security work. However, there have been several challenges related to teamwork and culture. Different work cultures, needs and perspectives exist among security practitioners and developers. Psychological safety and trust are linked to both team and individual learning behaviours. A lack of trust is associated with a decrease in productivity, information sharing and team morale.

There is a need for more knowledge on human-related challenges with integrating agile development and security. In this project, we aim to investigate how trust relations in a software development team and with relevant outside stakeholders relate to how information security work is conducted in the software development team. The objective is to understand current practices and challenges and develop recommendations for building trust between the software development team and the external responsible security actors outside the team. With broader insights into the human-related issues with integrating security work into agile methodologies, this can ultimately help in reaching the goal of creating more secure software.

Date approved:	2022-02-14
Supervisor:	Maria Bartnes, NTNU, IIK & Sintef
Cosupervisor:	Roy Myhre, Sopra Steria

Abstract

Agile development has gained widespread adoption in the industry and attention in the academic communities. The development methodology has introduced many advantages for the software delivery of the IT solutions, such as a greater focus on interdisciplinary cooperation and early time to market. However, the fast-paced nature of agile development has resulted in concerns regarding the security of the IT solutions being developed. Security practitioners have reported being unable to keep up with the pace of software releases and issues related to cooperation between security practitioners and development teams. There is a need for more research on human-related factors to improve the issues with collaboration. This study investigates trust relationships between development teams and security stakeholders to determine how it affects the work with information security. High levels of team trust have been shown to have beneficial effects on cooperation and team performance. To further investigate trust relationships and the consequences, we have conducted a multiple-case study with semi-structured interviews. Our findings suggest that the trust relationships between the development team and the stakeholder affect the work process related to information security work. The challenges associated with trustworthiness factors are primarily encountered in one direction of analysis; the development team being the trustor and the security stakeholder being the trustee. Challenges are found to be related to four perceived trustworthiness factors: ability, predictability, transparency and benevolence. The consequences of these challenges related to information security work were found to be primarily process-related; developers taking shortcuts, inefficient feedback cycles and a lack of motivation to engage in security work. We provide four recommendations for building trust between the development team and the security stakeholder to improve the work-related information security processes.

Sammendrag

Smidig utvikling har de siste tiårene fått stor oppmerksomhet i IT-industrien og forskningsmiljøet. Utviklingsmetodikken har introdusert mange fordeler for programvareleveranse, blant annet et større fokus på tverrfaglig samarbeid samt raske og hyppige leveranser. Men smidig utvikling har også ført til bekymringer knyttet til sikkerheten til IT-systemene og programvaren utviklet med metodikken. Sikkerhetspersonell rapporterer om at de ikke klarer å holde følge med det høye utviklingstempoet til smidig utvikling. De rapporterer også om problemer med samarbeid mellom ansatte i sikkerhetsroller og utviklere. Smidig utvikling er en metode som baserer seg mye på menneskelige faktorer og samarbeid, og det trengs mer forskning på dette feltet for å løse samarbeidsproblematikken. Tillit i team har vist seg å ha positive konsekvenser for samarbeid og effektivitet. Denne studien undersøker tillitsrelasjoner mellom utviklerteam og sikkerhetsaktører rundt teamet og hvordan tillitsrelasjonene påvirker arbeidet med informasjonssikkerhet i teamene. For å undersøke dette, har vi gjennomført en multiple-case studie med semi-strukturerte intervjuer. Våre funn indikerer at tillitsrelasjoner mellom utviklerteam og sikkerhetsaktører påvirker arbeidsprosessen med informasjonssikkerhet. Utfordringene relatert til troverdighetsfaktorer ble stort sett funnet i én retning av forholdet: når utviklerteamet er den som skal stole på sikkerhetsaktøren. Vi har funnet utfordringer relatert til fire troverdighetsfaktorer; evne, forutsigbarhet, transparens og velvilje. Funn om konsekvenser viser at det var mest prosessrelaterte konsekvenser; utviklere tar snarveier, ineffektiv kommunikasjon mellom aktørene og mangel på motivasjon for å engasjere seg i informasjonssikkerhetsarbeid. Til slutt, foreslår vi anbefalinger for å bygge tillit mellom utviklerteamet og sikkerhetsaktøren for å bedre arbeidet med informasjonssikkerhet i utviklerteam.

Preface

This master thesis was prepared in the spring of 2022 at the Norwegian University of Science and Technology (NTNU), Faculty of Information Technology and Electrical Engineering (IE), Department of Information Security and Communication Technology (IIK). This work concludes our five years as students in the master's programme in Communication Technology.

We would like to thank our responsible supervisor Maria Bartnes and our co-supervisor Roy Myhre. They have been of great help throughout the project, provided valuable feedback and helped us get in touch with participants. We would also like to thank Daniela Soares Cruzes for giving us the idea of the topic area and Inger Anne Tøndel for her feedback and opinions on the topic area and methodology. Additionally, we would like to thank the participants for sharing their experiences and contributing with their expert opinions.

A special thanks to our friends and families who have been of great support over the last five years.

Helga Tenold Fridtun & Knut Formo Buene
TRONDHEIM, NORWAY
JUNE 2022

Contents

Appendices

List of Figures

List of Tables

List of Acronyms

CISO Chief Information Security Officer.

CIT Critical Incident Technique.

GDPR General Data Protection Regulation.

GSE Global Software Engineering.

IE Faculty of Information Technology and Electrical Engineering.

IIK Department of Information Security and Communication Technology.

NIST National Institute of Standards and Technology.

NSM The Norwegian National Security Authority.

NTNU Norwegian University of Science and Technology.

RTR Risk-taking in Relationship.

RVA-analysis risk and vulnerability analysis.

Chapter 1 Introduction

Over the last two decades, there has been a paradigm shift in the approach to software development. The IT industry has moved from the traditional waterfall approach to the usage of agile methods. The traditional waterfall method is characterised by planning before execution and detailing the entire development life cycle before implementation. Agile development shifts the focus from processes and tools to individuals and interactions and from following a plan to responding to change [Agile manifesto]. Several frameworks have evolved from the agile method, such as DevOps and DevSecOps. DevOps has introduced advantages such as shorter delivery times, faster deployment, automation, earlier time to market, and a strong focus on work culture and collaboration in teams. It also focuses on breaking down information silos, a known problem in software development. DevOps has gained widespread adoption throughout the IT industry and continues to receive attention in the industry, such as the reports presented by Puppet [Pup21] that yearly evaluates the current state of DevOps in the industry. A cause for concern regarding security for agile development is the lack of dedicated time to prioritise security efforts. With its preplanned project phases, the traditional waterfall method was capable of dedicating phases to ensuring that security is handled, which is not the case for agile development and DevOps. Several attempts have been made to prioritise security in the agile methodology, but this has proven difficult. Security professionals report being unable to keep up with the pace and frequency of deliveries in agile development, which can be detrimental to information security [RZBS21].

A report from 2019 by The Norwegian National Security Authority (NSM) detailing the risks Norway is facing claims the country's digital risks are growing, along with the value of the information that Norwegian companies manage. The country's most prominent threat actors are state intelligence and criminal organisations. Organisations in the critical infrastructure sector are especially at risk for cyber security incidents during digitisation phases. Consequently, these organisations also have the most significant focus on cyber security. The report suggests that Norwegian businesses have gained more awareness about security but points to the need for the work to continue. In addition to preparing technical solutions against security incidents, organisations need good leadership and thorough work processes to avoid exploitation [NN19].

Integrating security in the life cycle process of software development is important to the ultimate goal of creating secure and robust IT services and software applications. If vulnerable IT solutions make their way into production and the hands of consumers, it will have significant negative impacts on several stakeholders. Companies, regardless of the company size and the maturity of information security, are affected by attacks that can have severe consequences due to vulnerabilities in software. The mainstream media frequently reminds us of the consequences and repercussions of these security incidents. Consequences for companies operating in critical infrastructure include the possible loss of life, misuse of sensitive information and significant economic losses, underlining the importance of the topic area. Companies hold more valuable digital information than ever, such as information about end-users or specific markets. The General Data Protection Regulation (GDPR) guidelines have the purpose of protecting user-data privacy, and huge fines might be waiting for companies that practice unlawful data collection or processing. IT companies depend on good reputations to grow their customer bases and remain regarded as respectable actors in the market. IT consultancy companies are dependent on good reputations in order to win offers. Moreover, with the ever-increasing pace of software releases and widespread digitisation in many organisations, integrating security in the software development process is becoming more important than ever.

DevSecOps is an approach that has been proposed as a solution to the problem of integrating security in the development life cycle. DevSecOps aim to integrate the security discipline along with development and operations. It focuses on building security into the product by integrating security in pipelines which provide continuous integration, continuous delivery and deployment (CI/CD) [Atl21a]. Whether security is an integrated component in DevOps is debated. However, many attribute DevSecOps as the approach to which security is a central component. DevSecOps has gained attention in the research communities, where there have been efforts looking into challenges and solutions for implementing the approach in the organisation [RZBS21; SC20]. Some of the common challenges are related to human factors, collaboration, and communication [SC20; TLH19]. The human aspect and cooperation across different disciplines are important factors when adopting agile methodologies. The academic literature also calls for more research on the human aspects of such methodologies [SMH18; RZBS21; SC20], as it is one of the less-studied perspectives of DevSecOps.

Research on trust in teams shows that it has beneficial effects. High degrees of trust are often found in high-performing teams, and it has shown to have positive effects on team creativity [BFG15], team performance and team satisfaction [CRT01]. Trust in teams is also strongly related to team members' attitude towards the organisation they are employed at [Cos03]. Trust has been researched in the context of Global Software Engineering (GSE), where it is viewed as essential for applying agile work methods in the globally distributed team [JGŠ10; NBV06]. However, trust relationships between software development teams and security stakeholders that work with the team have not, to our knowledge, been researched extensively in non-distributed teams. Some common characteristics of the security stakeholders are that they work with the team and dictate information security requirements. The topic of interest for this study is therefore to investigate trust relationships among the development team and information security stakeholders, and how the trust relationship between them affects the work with information security in the team. The research questions for the master's thesis are the following.

> *RQ1: How do the trust relationship between the software development team and the information security stakeholder affect the information security work in the team?*
>
> *RQ1.1: What challenges related to trustworthiness factors exist between the team and the information security stakeholder?*
>
> *RQ1.2: If challenges are present - what are the consequences related to information security work in the team?*
>
> *RQ1.3: What are some recommendations for building trust between the team and the information security stakeholder?*

Trust is essential for high-performance in other team contexts, which motivates us to investigate this in our context as well. By exploring these research questions, we want to contribute to a broader understanding of the human-related issues of cooperation between security practitioners and the development team through the perspective of trust. By highlighting the information security related consequences of challenges with trust relationships, we aim to bridge these two disciplines of information security and trust to give a broader perspective to the current practices and solutions in software development. We also want to contribute with recommendations to stakeholders in software development (such as the team, security stakeholders and facilitating roles) to raise awareness of trust-building in essential cooperative relationships.

We have used a multiple-case study approach with semi-structured interviews to answer these research questions. There are four different cases. The first three cases are from organisations in the public sector, the telecommunication sector, and the critical infrastructure sector, respectively. Each case has participants consisting of developers in the team and security stakeholders. The last case is a collection of individuals consisting of developers and one security stakeholder who work in separate companies and do not share any work relations. All the participants apply work processes that originate from the DevOps approach.

The report is structured as follows. Chapter 2 outlines relevant research on the topic of trust research, as well as a trust framework which we are applying in our study. The chapter also includes related work to development approaches such as agile, DevOps and DevSecOps. The overall research methodology is presented in Chapter 3. Chapter 4 presents the findings of our study, and Chapter 5 addresses the research questions as well as a discussion of the limitations of the study. Finally, Chapter 6 concludes our work, and we give recommendations for future work in the area.

Chapter 2
Background

An extensive amount of research has been conducted on agile work methodologies. This chapter presents literature on agile development approaches such as DevOps and DevSecOps. The theoretical framework of trust and relevant concepts and definitions of trust are also presented in this chapter. Related work of trust in team contexts is also introduced.

2.1 Agile development, DevOps and DevSecOps

The current state of agile methodology, DevOps and DevSecOps were reviewed along with related work and relevant background material in a specialisations project preceding the master's thesis [FT21]. Relevant material from the specialisation project will supplement the section below.

Agile development was first proposed in 2001 and addresses a new way of software development as opposed to the previous traditional methods [Agile manifesto]. It addresses the problems within teams and processes related to rapid change. Specifically, it suggests that teams can be more effective in responding to change if it can reduce costs related to moving information between people and also decreasing the time between decision-making on a change until the consequences of that decision are seen in action [CH01]. Many of the methods within agile movements are implemented in the DevOps approach.

There are four main ceremonies in agile methodology that are essential for maintaining agility in software development [SKF21]. These are meetings with defined duration, frequency, and goals during a sprint. A sprint is an iteration of typically two to four weeks. The first ceremony is the daily stand-up meeting, which is a short status meeting where team members can also address blockers. The second ceremony is sprint planning, which is a meeting held once in each sprint to plan out the sprint's goal and the backlog of this sprint. At the end of the sprint, the team will conduct a sprint review and a sprint retro. In the review, the team discuss and show the work completed in the sprint. In the retrospect, they discuss the team performance in that particular sprint with a focus on increasing effectiveness [Monday].

The DevOps approach is derived from agile development and aims to integrate development and operations in software delivery [EGHS16]. The approach emphasises people, collaboration and culture among development, operations and stakeholders to improve customer value [SH21]. Some of the highly emphasised features in DevOps are automation, fast deployment and infrastructure monitoring. It is an organisational change where previously siloed work functions are merged into multidisciplinary teams [EGHS16]. DevOps has achieved widespread adoption in the development industry. The people, culture and teams are important factors in DevOps, as it is a human-centred movement. The team can contain a diversity of roles. However, the teams usually consist of developers and IT operations that work collaboratively throughout the product life cycle to achieve faster delivery times, more seamless deployment and increase the quality of the software [Atl21b].

DevSecOps (Development, Security and Operations) is an approach derived from DevOps, where values from DevOps are applied to teams where security is an active and integrated part of the development process. The motivation for DevSecOps is to have security incorporated into the continuous integration, to avoid the more traditional approach of only engaging in security activities towards the end of a project [Atl21a]. The values of DevSecOps principles foster collaboration between development and security in the early days of development and security practitioners actively being used in the life cycle. Catching vulnerabilities near the end of a project can lead to costly corrections. Integrating security measures in the entire life cycle will lower risk and time spent on vulnerabilities as well as make it easier to understand risk and create policies and procedures [MC17]. Automating security is also one of the key principles of DevSecOps as it allows security controls to be fast and scalable, making it possible for security mechanisms to keep up with development [MC17].

Providing concise definitions and distinctions of DevOps and DevSecOps has shown to be complicated. Some terms that are used synonymously with DevSecOps are SecDevOps and DevOpsSec [SC20]. Mature DevOps practitioners ensure that testing, deployment and validation of requirements are addressed continually throughout the development process. It would allow for fast recovery if an incident were to happen. This affirms the saying “DevSecOps is DevOps done right” [ASMA22]. Even though one could argue that security is a natural part of DevOps, the term DevSecOps helps draw attention to addressing security concerns in the early stages of software development [SC20].

2.2 Information security work

The National Institute of Standards and Technology (NIST) define information security as *"protecting information and information systems from unauthorised access, use, disclosure, disruption, modification, or destruction in order to provide integrity, confidentiality, and availability"* [NIST01]. Information security is a multi-faceted field, with several aspects which can be investigated. Some of these aspects are IT security and software security. IT security can be defined as information security in IT systems [TJC20]. This involves efforts to follow standards and policies, mitigate risks, and ensure the data being processed by the software is secure. Software security pertains to creating software that, under malicious attacks, remains operating correctly [McG04]. Software security is a central part of well-functioning IT security [TJC20]. The study's participants stem from both the IT security and software security fields. The study will investigate the trust relationship between developers and information security stakeholders, focusing on processes and collaboration within the realm of IT security. In the thesis' research questions and further discussion, these efforts will be referred to as information security work.

2.3 Software developers

Developers in a software development team are a diverse group of people. Senior developers are experienced in software development and have profound knowledge of the entire development cycle in addition to coding. They often have more responsibility for the solutions and can delegate tasks to the team [Full Scale]. On the other side, junior developers are entry-level programmers with less development experience. Developers might work with the solution's back-end, which refers to the core of the IT solutions. This can include database management, creating APIs and programming on the core system. Front-end developers work on the interface on which the end-user interacts. They ensure the end-user can use the functions and features of the solution [Kenzie Academy].

2.4 Information security stakeholders

A security stakeholder may comprise different roles or a group of people. Examples of security stakeholders may include security and solutions architects, security experts or IT security specialists. Common characteristics of the security stakeholder are that they work with the software development team and set some premises for the information security requirements in the team. Some IT organisations may implement these roles if information security is of high priority (such as critical infrastructure companies). This role can operate as a mediator between the development team(s) and the organisation's security department. This role also has the responsibility of conveying security requirements from the security leader group of an IT organisation (which may be a Chief Information Security Officer (CISO) or the group the CISO is leading) to the software development teams.

A security champion is another form of a security stakeholder. However, because they usually reside in the team and perform development tasks, they are not regarded as security stakeholders in this study but rather as development team members.

A penetration tester is a security practitioner that helps organisations identify and resolve security vulnerabilities affecting their solutions, systems or products [Cyber Degrees]. They simulate cyberattacks and try to find security vulnerabilities in the system before malicious actors can find and exploit them. They are often consultants that are contracted to a project, but some organisations employ them full-time.

2.4.1 IT operations team

IT operations teams or departments can also be considered security stakeholders in IT organisations, as they often work within the information security domain. The overall responsibility includes maintaining the IT infrastructure in the organisation. This may comprise setting up and maintaining firewalls for the organisation's internal networks. They are also responsible for ensuring that organisational policies are implemented across development teams [BMC Blogs]. Usually, they manage the employees' work computers and decide the configurations and policies for the computers, and may also do general user support inside the organisation. There is an important distinction between the operations team and the "ops" in DevOps. DevOps focus on integrating development and operations in order to have continuous delivery of software solutions. However, the operations team is responsible for the organisation's overall IT, which also comprises network configurations.

2.4.2 CISO

The CISO is responsible for coordinating the security policies throughout the organisation and remains the most influential security stakeholder of an organisation. With the increasing focus on IT security in organisations, the CISO in the organisation is moving from a technical role to an executive role. A key factor to a successful CISO is to have the right combination of soft skills and hard skills [vYSVR21]. Good communication, leadership, and interpersonal skills are considered important skills to have [SvYVR21; vYSVR21]. Being able to trust the peers as a CISO is important [Bad21].

2.4.3 Challenges related to the human factor in DevSecOps

Challenges and proposed solutions related to human factors in DevSecOps were explored in the specialisation project [FT21]. Relevant background material from the specialisation project will be included and used as supplementary material in Subsections 2.4.3 and 2.4.4.

Earlier literature has identified several challenges related to adopting DevSecOps in an organisation. These challenges relate to organisational change and trust [SC20; RZBS21]. For an organisation to succeed at DevSecOps, possible changes in skills, culture, tools, processes, standards and practices should be considered.

Trust is one of the foundations for DevSecOps to function beneficially and should be established between the DevOps team and the security team in order to implement security practices on a daily basis. Security practitioners that are not well integrated with the team might be viewed as having a lack of respect and not to be taken seriously [SC20]. Insecurities stemming from a lack of security knowledge amongst developers and unwillingness to help developers by security practitioners hinder both of them from trusting each other [McG18].

Several challenges within DevSecOps have been outlined in an extensive literature review [RZBS21] in regards to people-related issues and cooperation. One of the main reported issues was the lack of inter-team communication between the development and security teams. Developers reported feeling attacked when given feedback by security personnel, who would point out security flaws in their work. They often felt they had to make security considerations that would take away the autonomy from their daily work [TLH19]. Siloed workflows are still reported as being an issue in the IT industry, especially regarding the separation between security practitioners and software developers. The inclusion of a security role in the team was deemed a necessary countermeasure [SC20]. A lack of security knowledge amongst the developers was also found to be an issue. This, in part, may be due to the divide between the education related to software security and software development. Challenges related to organisational culture were also reported, especially regarding a lack of security responsibility within the organisation and management. This responsibility was lacking in the development teams as well.

Other challenges and issues were related to practices in a DevSecOps setting, focusing on rapid releases. The traditional security practices have often been performed manually, and the literature review report on the difficulty in adopting these in a continuous and automated manner to fit the DevOps approach [RZBS21]. As a result, there is a gap between the security practitioners being hesitant to rely on automated processes and development going forward rapidly. Some of these practices include being in compliance with standards, privacy by design, risk analysis, threat modelling, and risk management [RZBS21]. These practices can be time-consuming and need some form of human input that opposes the desired development speed of DevOps. These activities include assessing and verifying security requirements, measuring security and performing security testing. It was reported that there is a lack of proper tools to support the automation of these activities. As a result, teams face trade-offs between the pace of development and security.

Quick feedback and seamless communication are essential features in DevSecOps because it supports traceability, enabling fault localisation and problem-solving. However, quick feedback loops have shown to be difficult due to traditional methods of data gathering in security handling and cultural differences between developers and security practitioners [RZBS21].

2.4.4 Proposed solutions related to the human factor in DevSecOps

Building Security In Maturity Model (BSIMM) is a framework with the objective of planning, measuring and executing software security through studying implemented security measures and practices in partnering members [BSIMMa]. The framework makes it possible for an organisation to compare its security efforts to what is outlined in BSIMM and possibly detect areas that need attention [BSIMMc]. This is done according to a maturity model which can concern employees, culture, security practices or technology. One of the strategies outlined in BSIMM is the creation and fostering of satellites. These are often referred to as security champions and are employees with a high interest in information security [BSIMMb]. It is one of the most common recommendations on how to ensure that security has higher priority in the team due to the security champion acting as a bridge between development and security teams [RZBS21; TJC20]. Often the security champion will be one of the developers from the development team, and a positive side-effect of this is that the security champion will not be viewed as an outsider who is there to judge or criticise their code [RZBS21].

Different solutions have been proposed to mitigate the security issues encountered within DevSecOps teams. A left shift in security, meaning including security tasks and people from the beginning, is recommended in DevSecOps. This can include having an appropriate feedback framework to support a short feedback cycle or engaging developers more in security tasks. The collaboration also needs to be controlled and follow standards, especially an established way of communication and clear separations of tasks between interdisciplinary team members. Clear communication channels are also important when dealing with risk information and evaluation results. If guidelines of communication are in place, this can increase trust amongst the stakeholders [RZBS21].

Implementing security training for knowledge sharing will help raise awareness of security. For instance, proper training on how to use the relevant security tools or having basic knowledge of security threats are helpful. Security training could be in the form of online activities or security awareness sessions like dojos [RZBS21; SH21].

2.5 Trust

Trust is often viewed as a key success factor in organisations today to ensure effective collaboration. Trust has been researched to a great extent in different disciplines such as computer science, social sciences, management and psychology [TSS22], and many scholars seem to agree on the importance of trust in the various disciplines [Sch06]. Trust is a multidimensional concept and can be investigated on different levels, such as individual, team, and organisational levels.

2.5.1 Definitions

As trust has been researched through the perspective of numerous academic disciplines, many different definitions of trust have been proposed. Some are more recognised than others, and one of the most commonly cited definitions is by Mayer et al. They define trust in an organisational setting as the *"willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party"* [MDS95]. The majority of subsequent definitions of trust centre around the two main parts of Mayer's definition: how willing one is to become vulnerable to the actions of another party and the expectation of being treated well by that party [FG12; MT11]. Mayer et al. described trust in relationships between two parties. Such relationships between two parties are referred to as dyadic relationships. Rousseau et al. have proposed another definition: *"Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behaviour of another"* [RSBC98]. He emphasises that trust is not a behaviour or a choice but a psychological state that can cause or result from such behaviours. Hence, he supports the thought that trust is a characteristic of a relationship rather than a characteristic of the individual. In our study, we are using the definition of Mayer et al.

2.5.2 Concepts of trust

An extensive literature review conducted by Fulmer and Gelfand in 2011 [FG12] gathered research on trust conducted between 2000 and 2011. The paper analysed 375 papers published in 20 different journals. The authors introduced a multilevel-multireferent framework, which separates between trust existing *at* certain levels and *in* certain referents. This is shown in Figure 2.1.

Table 2.1: A multilevel-multireferent framework of trust [FG12]

		Referent	
		Individual referent	Team referent
Level of analysis	Individual level	An individual's trust in another individual	An individual's trust in another team
	Team level	The aggregated trust in a team towards an individual	The aggregated trust in a team towards another team

Trust existing at a certain level refers to the level in the organisation the analysis is applied. The paper evaluates trust in three organisational levels (we have only included two levels in Figure 2.1). In an ascending manner, these levels are trust at the individual, team, and organisational levels. At the individual level, trust refers to the individual's degree of trust. Trust at the team and organisation level refers to the joint degree of trust existing among members of a unit. As a result, team-level and organisational level trust is the aggregate of trust among team members or organisations, respectively. In our research, we study trust at the individual level in an individual referent. Ideally, we would be researching trust at the team level in the individual referent. However, as we are interviewing only parts of the development team, we cannot research the aggregated trust of an entire development team.

The two parties involved in a trust relationship are the *trustor* and *trustee*. The trustor is the individual in the relationship who puts trust in another party. The other party who receives this trust from the trustor is the trustee. In the multilevel-multireferent model, the trustee is referred to as the referent. At each analysis level described above, the paper details three possible referents: interpersonal, team and organisational referent. Trust in an interpersonal referent is trust placed in a person, like, for instance, a friend, office manager or stranger. The team referent is trust placed in a group of people who share a common goal. Trust in the organisational referent is the trust placed in an organisation.

2.5.3 Propensity to trust

The trust relationship is influenced by the traits of the trustor and trustee. Mayer et al. describe some factors of the trustor that influence the trust relationship. The most influential of these factors is the trustor's propensity to trust. This refers to how inclined the trustor is to place trust in someone. The degree of someone's propensity to trust others differs based on their personality type, technical experience and cultural background.

2.5.4 Perceived trustworthiness

There are aspects of the trustee which affect the dyadic trust relationship. One is more likely to trust someone if the person has some given personality traits. These traits are collectively viewed as the trustee's perceived trustworthiness. Mayer et al. detail three key aspects of the trustee that affect trustworthiness. These are the trustee's ability, integrity and benevolence [MDS95].

A trustee's *ability* refers to the technical skills or competence that would warrant trust being placed in the individual. Such competence could give the person a degree of authority in a particular field or area. A trustee who possesses a certain skill is more likely to be trusted with tasks related to that skill. *Benevolence* pertains to the image of the trustee's inclination toward the trustor. The trustee's benevolence positively affects its trustworthiness if the trustor feels the trustee has a certain selfless inclination towards themselves, regardless of its own gain. A benevolent trustee thus implies that the trustee looks out for the interests of the trustor as opposed to acting opportunistically. The aspect of the trustee's *integrity* centres around the personal principles the trustor considers the trustee is following. If these principles and values align with the trustor, he/she is more likely to trust the trustee [MDS95].

2.5.5 Mayer's model of trust

Mayer et al. introduce a proposed model of trust between two individuals. The model can be seen in Figure 2.1. The trust relationship is divided into the antecedents of trust, the trust itself and the consequences of trust. The antecedents of trust are the factors that must be in place to facilitate trust. The model shows the factors of a person's perceived trustworthiness as the first part. This, together with the trustor's propensity to trust, affects the trust placed in the trustee. The next part of the model is risk-taking in relationship. This refers to the amount of risk the trustor is willing to accept when trusting someone. Trust will coax the risk-taking in a relationship, where the amount of trust affects the amount of risk. Risk-taking is influenced by the trustor's perception of the risks linked to the situation.

There is a risk involved when placing trust in someone. This is illustrated in the models' subsequent phase. The level of trust the trustor has in the trustee affects its risk-taking behaviour. Risk-taking in Relationship (RTR) is the relationship-specific risk the trustor is willing to take. Risk-taking is further affected by the situation's perceived risks. For risk-taking behaviour to occur, the trustor's level of trust towards

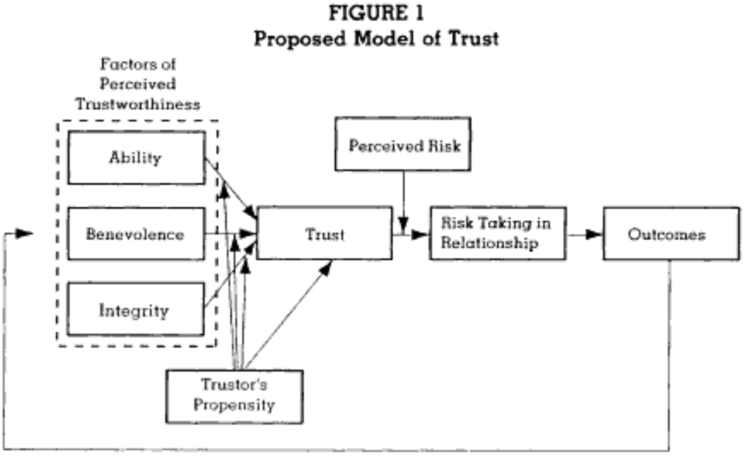

Figure 2.1: Mayer's Model of Trust [MDS95]

the trustee must exceed the perceived risks involved. The situation outcome leads back to the trustees' perceived trustworthiness factors, affecting the future trust relationship between the parties and further risk-taking behaviour by the trustor.

So how do these perceived trustworthiness factors ultimately create a solid foundation to create trust in a dyadic interpersonal relationship? McKnight et al. connect the concepts [MC01]. They include predictability as one of the main factors within trust, in addition to ability, benevolence and integrity. Let us consider the case of someone providing help for someone else. The trustee might be willing (benevolence) to help but might not always be able (ability). That is why the combination of these is much stronger than one of them alone. If the trustee also provides help to serve the trustor in an honest and ethical manner (integrity), then the trustor is more inclined to trust the trustee. If the trustor also consistently follows agreements (predictability), there is a good foundation for the trustor to trust the trustee.

2.5.6 Taxonomy of team trust

The work by Mayer et al. investigated antecedents and consequences of trust relationships in dyads. More recent studies have extended the research of dyadic trust into trust in teams. A study by Brauer et al. investigated trustworthiness factors as antecedents and risk-taking behaviour as consequences of team trust in virtual and face-to-face teams. The study expanded on Mayer's three factors of perceived trustworthiness, as seen in Figure 2.1, to include the two additional main categories; predictability and transparency. They argue that investigating trust in a team setting

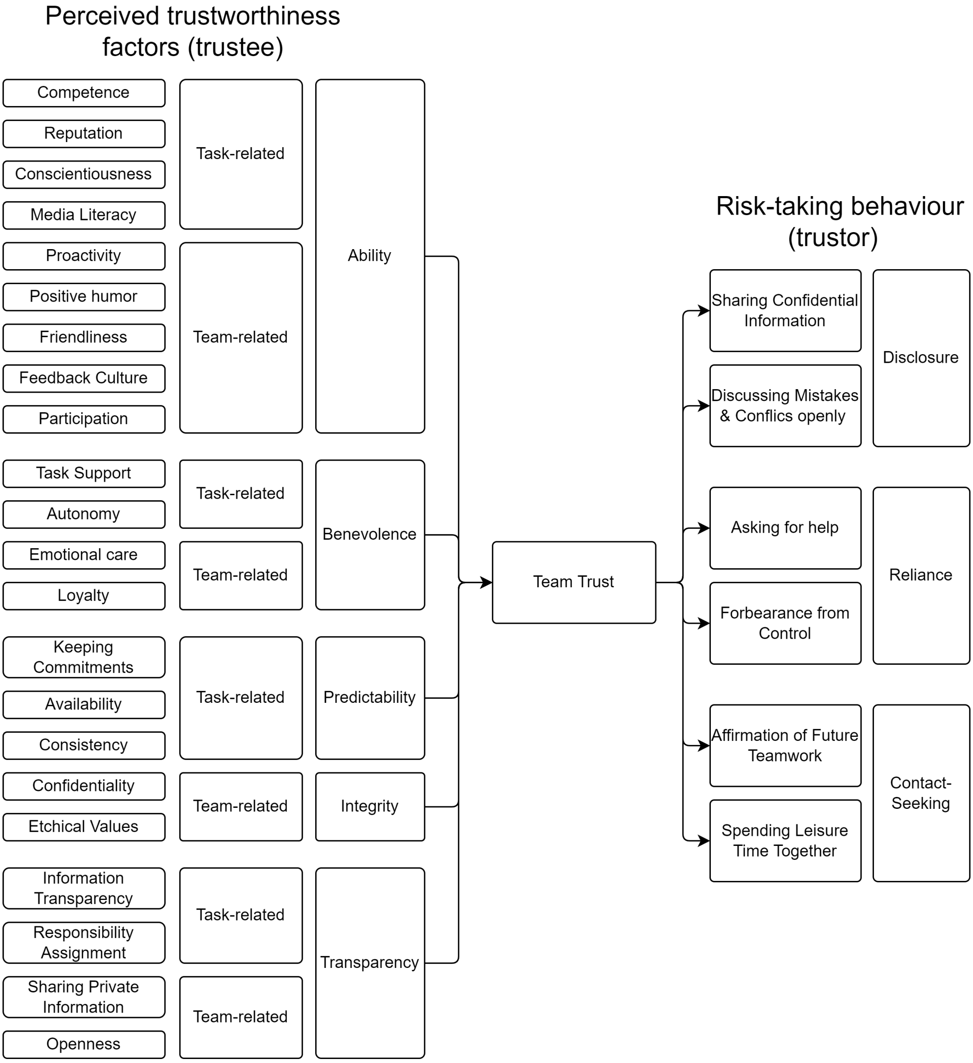

Figure 2.2: Taxonomy of team trust. The perceived trustworthiness factors on the left part of the figure are antecedents for team trust. The risk-taking behaviours on the right part of the figure are consequences of team trust [BHHH20].

is a more complex scenario than investigating a dyadic trust relationship. This is also argued in a review by Schoorman et al. [SMD07]. The study regards these five factors as main categories of perceived trustworthiness factors, which each have its respective subcategories. The resulting taxonomy of team trust is seen in Figure 2.2.

The study used Critical Incident Technique (CIT) to capture stories of incidents where teams either trusted or distrusted each other. Participants indicated that team trust was perceived to emerge when certain trustworthiness factors were present in the team. They are categorised into task-related and team-related factors, which are again categorised into the five main categories of trustworthiness factors. The study found 22 perceived trustworthiness factors that helped trust emergence in the teams. The study showed these factors occurring with varying frequency, which has impacted which factors we included in our study. The trustworthiness factors are as follows. *Competence* refers to the expert knowledge and work experience a peer has in a certain field. Positive *feedback culture* refers to having the ability to give positive and negative feedback in a friendly manner and keep the feedback work-related and not related to personality. Trust was perceived to emerge when there was a willingness to help fellow team members on work-related issues, which is referred to as *task support. Autonomy* pertains to trust emerging in teams where team members have control of their work and the ability to make their own decisions. *Loyalty* is about being faithful to others' decisions and obligations and being able to defend a decision as a team. *Ethical values* indicates that a team member is able to adhere to ethical principles and team values and norms. *Keeping commitments* includes team members keeping their promises, doing their assigned work and staying on agreed deadlines. *Information transparency* refers to team members sharing all relevant information with the team on a general basis. Team members report trust emerging when there are clearly defined roles and responsibilities, which is included in *responsibility assignment.* These factors all contribute to the emergence of trust in teams [BHHH20]. Our study has applied these factors in our interview guide to investigate the five trust dimensions; *ability, benevolence, integrity, transparency and predictability*, in the trust relationships between development teams and security stakeholders.

The study has also categorised the consequences of trust as three main categories of risk-taking behaviour. These are as follows. *Disclosure* is the first main category of risk-taking behaviour and involves the trustor making themselves vulnerable through acknowledging their mistakes or shortcomings, as well as sharing private information. *Reliance* is the second main category of risk-taking behaviour and revolves around the trustor making itself vulnerable by relying on those around them. Reliance includes giving the trustee responsibility and autonomy for work that is important to the trustor. *Asking for help* is a subcategory of reliance, which involves the trustor making itself vulnerable by asking for help, indicating a degree of trust towards the trustee. *Forbearance from control* is the second subcategory of reliance, where the trustor displays trust in the trustee by not feeling the need to monitor them or their work. *Contact-seeking* is the third and final main category of risk-taking behaviour. It involves the trustor initiating contact with the trustee with the intent of spending time together and building a personal relationship. By opening up to the trustee, the trustor makes itself vulnerable.

2.5.7 Trust and communication

The relationship between trust and communication has been subject to research for several years. Some studies suggest that communication leads to trust and vice versa. A study of communication and trust in teams found that the communication between team members influences their task performance by facilitating trust emergence [BFG15]. The researchers found that the communication among the team members led to the emergence of trust between them. Prati et al. also point to open communication between team members as a factor which fosters trust [PDF+03]. A study into the effects of communication and trust on project performance found indications of the relationship being directed the other way [CYL13]. They found trust to affect communication which, as a result, affects the project performance. Team members who trust each other will to a greater extent, be willing to share information, affecting project performance. The overall literature points to trust and communication as important contributors to the success of a project.

2.5.8 Benefits of team trust

The vast majority of research on team trust finds that it has positive outcomes for the team. Some of these outcomes are better team performance, team effectiveness [CRT01], and team creativity [BFG15]. It is also shown that high levels of trust in teams positively influence team satisfaction and relationship commitment [CRT01]. Recent research has found that team trust is positively related to attitudes in teams as well as information processing [BHH16]. Research on trust in organisations has found that organisations with high trust levels are found to rely less on control by management and supervisory rules [CP12]. Team members' attitudes towards the organisation have also shown to be strongly related to the level of team trust [Cos03]. According to other studies, people with high levels of trust in their colleagues or leaders display more selfless behaviours, including sharing information with others. They also showed higher levels of cooperation [JR00; AT03]. When team trust in teams is low, it has been found that task conflicts often are mistaken for relationship conflicts between peers, as well as related to more stress in the teams [Cos03].

2.5.9 Trust in leaders

Leaders have a significant role in establishing and maintaining trust in a team [TSS22]. Different types of leadership styles may influence how trust in the teams develops. Transformational leadership has been reported as a leadership style that promotes team trust, as well as trust in the leader [Sch11]. It is a form of leadership that promotes autonomy in the team and encourages team members to be self-motivated and self-driven, increasing team decision-making and ownership. Transformational leadership has been connected to improving the ability dimension of trust between them [TSS22]. Servant leaders have been found to increase team trust both in regards to ability, and benevolence [Sch11]. This leadership style is concerned with the personal and professional well-being of the team members and has been connected to raised team performance [Sch11]. Some studies [DF02; CSL07] have associated the existence of trust between an employee and its leader with an increase in the employee's job performance, as well as mitigating their intentions of quitting.

2.5.10 Trust in GSE

Significant research has been done on team trust in the context of agile teams in Global Software Engineering (GSE). Trust in GSE is one of the contexts closest to the study's topic area that has seen significant academic attention. The development teams in GSE comprise members who are not co-located, working together across different countries and time zones. Teams in GSE utilise online platforms to conduct their meetings [JGŠ10].

Recommendations for building trust in GSE have been proposed in several studies. Some of these recommendations are collected from the specialisation project preceding the master project [FT21]. The recommendations are supplied with more literature that has become available after the specialisation project.

A recent study identified five different contributing factors in an organisation for building trust in distributed teams [TSS22]. The five categories are working environment, leadership, organisation, personal factors and socio-cultural factors. The categories were further divided into different sub-concepts for guidelines to build trust. Working environment includes concepts such as 1) supporting an open working environment, 2) psychological safe working environment, 3) transparent working environment and 4) engaging working environment. Leadership styles that can promote trust emergence are transformational leadership and servant leadership. The organisation is recommended to use best practices of agile methods and provide coaching in agile for the employees.

At the team level, findings suggest that expectations of teamwork and methods should be addressed in the teams in an early phase, and they should be clear and well-set [JGŠ10]. The project manager should engage the team in trust-building behaviour in the beginning phases of projects. Another finding suggests that teams formed from the same organisation often share some common values, and hence there exists some degree of initial trust. This is relevant for organisations that use consultants from consultancy companies, where teams consist of some people from the hiring organisation and some consultants from the consultancy company. In this case, the project manager is also recommended to engage in trust-building behaviour in the beginning [JGŠ10].

The recommendations for best practices for building trust emphasise good communication patterns that can address the common issues in GSE, along with the important role of the project manager as a facilitator of building trust inside the team. Other studies have also found the project manager role to be an essential stakeholder when building trust [BIS07; NBV06]. Cultural understanding amongst stakeholders from different organisations or nationalities has also proved to be important when building trust [JGŠ10; BIS07; NBV06].

2.5.11 Trust and psychological safety

Agile development embraces rapid and iterative change as one of its core values. For development teams to be successful in an agile methodology, they need to be able to embrace change and hence be resilient to changes quickly. Psychological safety has been addressed as an important factor to be present in a team in order to create team resilience [HR21; LBL11]. Team resilience has been defined as a *"team-level capacity to respond and bounce back from adversity"* [HCJW20]. Edmondson defines psychological safety as *"a shared belief held by members of a team that the team is safe for interpersonal risk-taking"* [Edm99].

Such risks coincide with the risk-taking behaviours when trust is present and can include the risk of being considered ignorant or incompetent if one is to ask simple questions or the risk of being presumed as negative when inquiring about worrying aspects of a project [LBL11]. Psychological safety has been suggested to affect many aspects of team performance, for example, information sharing, self-learning and team learning behaviours, innovation and speaking up for team improvement [HR21; LBL11].

Psychological safety and trust are similar constructs, and Edmondson has addressed the similarities and differences in her work [Edm11; Edm18]. Both constructs relate to interpersonal experience and can affect various behavioural and organisational outcomes. Both describe perceptions of risk-taking or vulnerability and making choices to minimise negative consequences for teams. She describes a difference between psychological safety and trust in the focus on oneself versus others. Trust is often related to giving others the benefit of the doubt, which indicates a focus on others' potential actions or trustworthiness. Psychological safety differs from trust with a focus on others, giving them the benefit of the doubt when they have made a mistake, for instance [Edm11]. However, this definition of trust refers to the earlier adopted definitions of trust [MDS95] and does not encompass the extensions to this dyadic relationship of trust. Extended definitions of trust [BHHH20] explore trust in a group or team and have more in common with the definition of psychological safety developed by Edmondson [Edm99].

Chapter 3 Methodology

Our chosen method for investigating the research question is the qualitative approach. We have conducted a multiple-case study of four cases, and the chosen data collection method is semi-structured interviews combined with the interview technique of war stories. Fourteen participants from the IT industry were interviewed, distributed across four different cases. The interviews were transcribed and coded in NVivo, and open coding was performed in the data analysis. The interviews were conducted in March 2022 and the beginning of April 2022. At the beginning of the project, we also conducted an unstructured literature search into topics related to interpersonal trust, team trust and organisational trust, and challenges and solutions of implementing the agile approach, DevOps and DevSecOps.

Several research methodologies are appropriate for researching trust. The Handbook of Trust Research [LMS11] gives an overview of different approaches used for trust research, both qualitative and quantitative research. The nature of qualitative research allows the researcher to discover the inner thoughts and experiences of the participants. It can be a less rigid and structured way of conducting research than quantitative research methodologies, which reach conclusions through variables and statistics. Qualitative research lets the researchers step into the world of the participants, see perspectives from their point of view, and ultimately contribute to the development of empirical knowledge [CS08]. Our research questions seek to explore the trust relationship between the members of development teams and security stakeholders and how this relates to how they conduct their information security work. To map out the trust relationship in our data material, we use a comprehensive taxonomy of trust factors proposed by Breuer et al. One could argue that a quantitative approach could be fitting for our main research questions since we are looking for how a trust relationship *affect* information security work, and this could be viewed as how one variable affects another variable. However, the topic area of how trust relationships affect information security work in agile methodology has been researched to a limited extent. Therefore we argue that our research methodology is fitting for investigating our research question, as we get in-depth knowledge from interviewing participants in different contexts. To our knowledge, we are the first to map out trust between security stakeholders and development teams, focusing on how this affects information security work.

3.1 Data collection methods

Semi-structured interviews were used as the data collection method. A formal definition of a semi-structured interview is *"an exploratory interview used most often in the social sciences for qualitative research purposes or to gather clinical data"* [MB18]. The interview type allows for follow-up questions and further exploration of the direction in which a participant may lead a conversation. It has characteristics of both structured and unstructured interviews. Structured interviews follow a set of questions that all participants are asked in the same order. It makes a good foundation for comparing data with each other but does not allow much exploring outside the questions. As we want to be able to explore further in the direction that the participants may take the interview, structured interviews are not preferred. Unstructured interviews are characterised by not having the set of questions planned out in advance. Therefore, semi-structured interviews are an appropriate fit for investigating our research questions due to the exploratory nature of the topic and because it allows for some comparison between the interviews.

The interview guide comprises open-ended questions following the semi-structured interviews, as well as collecting "war stories". The guide is built up with lightweight warm-up questions and gradually goes more in-depth, and eventually collects war stories. One interview guide was created for each selection of participants. The reader is referred to Appendix B to view the interview guides. Trust relationships are not trivial to research empirically. It is not a topic that is very much reflected upon or thought about in everyday life. Therefore it can be challenging to answer direct questions about the topic. Participants may have very different perspectives of what trust means and comprise. Therefore, participants are not asked directly about the trust relationship between the team and the security stakeholder. Instead, they are asked about different constructs of trust that may be easier to connect to their everyday life. This way, the interviews' analysis and comparison will be more thorough. The questions related to trust relationships are developed based on several earlier studies [BHHH20; HR21], and are adjusted according to our research questions and context. Questions related to trustworthiness between the team and the security stakeholder are based on the perceived trustworthiness factors that were produced by Breuer et al. [BHHH20]. An important difference in our context compared to the context in [BHHH20] is where the trust relationship is investigated. In [BHHH20], the context of investigation is intra-team; they look at trust within the development team. In our context, we are looking at the trust relationship between the development team and the security stakeholder. However, the study is still a thorough basis for our study with the appropriate adjustments of questions. Table 3.1 shows which perceived trustworthiness factors are used as a basis for questions related to trust in our interview guide. We have only included a few of the 22 identified perceived trustworthiness factors. These factors are chosen based on how often they were mentioned by participants as trust emerging factors. Hence, the most mentioned factor in each main category of perceived trustworthiness factor is included as a basis for trust-related questions in our interview guide. We have applied these factors in our interview guide to be able to investigate the five trust dimensions ability, benevolence, integrity, transparency and predictability.

Table 3.1: The perceived trustworthiness factors are mapped to ability, benevolence, integrity, predictability and transparency. This mapping is developed by Breuer et al. [BHHH20]. These factors are used as a basis for questions related to trust in our interview guide.

Construct of trust	**Perceived trust-worthiness factor**	**Question in interview guide**	
		Selection 1	**Selection 2**
Ability	Competence	2, 14, 15	4, 10, 16
	Feedback culture	7	11
Benevolence	Task support	16	9
	Loyalty	6	
Integrity	Ethical values	8	
Predictability	Keeping commitments		7
Transparency	Information transparency	11	12

In addition to specific questions related to constructs of trust, war stories are collected to further capture trust dynamics in critical situations. These are open-ended questions with the intention of capturing stories that the participant finds interesting to elaborate on. The questions allow capturing rich details on a specific past event that was experienced as challenging to the participant [LS07]. The data collection technique is similar to CIT developed by Flanagan [Fla55], and both techniques capture details of particular events. In a CIT approach, the interviewees are usually asked about specific events that have occurred. However, war stories allow the participants themselves to decide what incidents are critical to report [LS07]. CIT and war stories are ways to capture trust dynamics and relationships in specific settings, such as encounters within teams. It collects rich data on how trust relationships show as behaviours involving creating, strengthening or destroying trust [LMS11]. Using both questions related to constructs of trust and storytelling will give a more detailed description of how trust relationships can affect security work in the development teams.

3.2 Sample description

A total of 14 participants divided into 4 cases were interviewed. Each case consists of representatives from two selections of roles; developers and security stakeholders. Table 3.2 shows an overview of the participants and the different cases to which they belong.

Table 3.2: Overview of the participants with relevant information. Contracted/employed denotes if the participant is contracted out to the company that owns the project or if they are employed at the company that owns the project. Years of experience denotes the years of relevant experience in IT.

Case	ID	Role	Selection	Gender	Contracted(C) /employed(E)	Years of experience
A	1	Senior Developer /Tech lead	1	M	C	12
	2	Senior Developer	1	M	C	7
	3	Security Architect	2	M	C	7
B	4	Junior Developer /Security Champion	1	F	C	1
	5	Junior Developer /Security Champion	1	F	C	1
	6	Information Security Expert	2	M	C	5
C	7	Senior Developer	1	M	C	3
	8	Solutions Architect	2	M	E	21
D	9	Senior Developer	1	F	C	4,5
	10	Senior Developer	1	M	C	6
	11	Senior Developer	1	M	C	7
	12	Senior Developer	1	M	C	14
	13	Senior Developer	1	M	C	20
	14	IT Security Specialist	2	M	E	25

Selection 1 consists of software developers working in teams. This group consist of

10 participants. They are all working with the agile software development approach. The requirements for participating in the study were being a senior software developer and having experience from development projects with security requirements. However, case B contains two junior developers due to not finding developers with the desired characteristics.

Selection 2 consists of security stakeholders that work with the team (either inside or outside), and provide security requirements to the team. This group consists of 4 participants. The title of this role is different depending on the organisation, even though their work tasks generally correspond. When referring to the specific security stakeholder in a specific case, we will use the respective role title used in 3.2. When referring to the general role in selection 2, we will use the name "security stakeholder".

3.2.1 Case A

Case A consists of three participants: two senior developers and one security architect. The three participants are members of the same team, composed of 11 people. One of the senior developers has a role as tech lead as well. The security expert is a part of other projects simultaneously as the project of interest and works part-time in this project. The participants all have university degrees. The participants are employed at a consulting firm and hired in the public sector to create a software application that collects sensitive information about Norwegian children. The consultancy firm is amongst the larger IT consultancy companies in Norway. The project started in January of 2022 and is expected to continue for at least one and a half years. The team is newly formed due to the project having started recently.

3.2.2 Case B

Case B consists of three participants: two junior developers and one security expert responsible for information security. The developers originally were on the same team, but the team was split. They are now members of two different teams. The two teams work closely together, and they have some overlapping members. The teams consist of 10 and 14 members. The two developers have newly taken the role of security champions in their respective teams. The security champion's task, in this case, is to be an intermediary between the team and the information security expert. The participants all have a university degree. Due to the two developers' similar study and work experience, they participated in a joint interview. The three participants are employed by one of Norway's largest IT consultancy firms. The participants are contracted by a Norwegian telecommunication company to work on the development and management of their IT systems. The security expert oversees the information security efforts of six different development teams, with the teams working on various IT systems. The two developers recently joined the project and the team. The information security expert has been working on the project for three years.

3.2.3 Case C

Case C consists of two participants: one senior developer and one solutions architect who sets the premises for the developer team's security work. The team consists of 7 people. The senior developer is employed by a medium-sized IT consultancy company in Norway and is contracted by a firm in the public critical infrastructure sector. The solutions architect is employed by this firm and works as an intermediator between the team and the company's security department. Both have university degrees. Case C is considered more mature than the former two cases because both participants have been on the project for two to three years. The project deals with cloud migration and structuring large amounts of sensitive data.

3.2.4 Case D

This case consists of individual developers and one IT security specialist. Three of the developers are from the same consultancy company as the consultancy company in Case B but are not further associated with each other. Two others are from the same consultancy firm but working on different projects. Therefore, they are regarded as individual developers with different contexts. The information security expert is from the banking industry in Denmark, but the rest are from Norwegian IT consultancy firms.

3.2.5 Recruitment of participants

Participants were recruited through the researchers' and supervisors' personal- and work networks. For cases A and B, contact was first made with one participant, who recruited the rest of the group. We gave the first contact point some guidelines for whom we desired to interview, and then the participant put us in contact with the suitable individuals. This procedure helped us get in touch with potential participants but made finding participants with the correct profile more challenging. In some cases, one of the researchers would have some acquaintance with the participants.

Our selection of participants is, in large part, busy with tight schedules. The participants are interviewed during work hours, resulting in them either taking time off of work or, in the case of the consultants, spending time paid for by their customers. Their tight schedules led to some interviews being cut short. Most of the interviews lasted from 30 to 45 minutes.

3.2.6 Prequestionnaire and consent

The project was approved by the Norwegian Centre for Research Data (NSD). The notification form to NSD can be found in Appendix C. Participation in the study was voluntary, and all participants received a consent form prior to the interviews. All participants gave consent, and in addition, they answered a prequestionnaire with a few questions about their background, work conditions and experience to give a richer context of the participants. The consent form and the prequestionnaire are presented in Appendix A. Selection 1 is a diverse group, so collecting enough background on work conditions is advantageous in the analysis phase.

3.2.7 Gender diversity

The sample should be representative of the populations from which we are recruiting. There is a gender imbalance present in our sample, containing three females in selection 1 and zero females in selection 2. That is a total of three females and eleven males in our participant group. A report from 2018 done by the ODA-network and Kantar TNS [IKT-Norge18] indicate that 15% of software developers were female in the Norwegian IT industry in 2018. The overall proportion of women in the Norwegian IT industry is 28%, including other roles, such as executive roles, sales, operations, and maintenance. Statistics indicate that in Norway, 23% of the people working in the IT sector in Norway is female [CSD]. No available reports give data on the gender representation in the population comprising security stakeholders, from which selection 2 is recruited. However, based on the report from ODA and Kantar TNS, we can assume there is an underrepresentation of females in these roles as well. A gender imbalance is present in the industry, which is reflected in our sample.

3.3 Digital interviews

All interviews were performed digitally via Teams. Interviews performed digitally have some advantages and disadvantages compared to physical face-to-face interviews. The most apparent advantage of digital interviews is accessibility to participants, giving more opportunities to recruit geographically distributed participants. It is more cost-efficient, as travel expenses are unnecessary. Being able to access participants without geographical limitations is the most significant advantage in our case since most of our participants reside in different cities in Norway. Due to the selection of participants generally being a busy group of people, digital interviews can better fit their work schedule as well. Another advantage of digital interviews is that participants can sit in their own comfortable space, either at work or home. This might be less intimidating than participating in a physical interview [GWRC20].

Disadvantages may include technical difficulties that can arise during the interviews. Internet connection issues and poor audio quality did happen for a few interviews. This can lead to inaudible sound and the loss of information. Distractions occurred during some interviews, and one interview was also interrupted to the point that we had to reschedule the interview. Small talk before and after the interview is more natural to occur in a physical setting, and it provides room for clarification, questions, or other thoughts a participant may have. This may be more difficult to capture in a digital interview. Facial expressions and body language may also be difficult to capture in a digital setting, as opposed to physical interviews.

3.4 Data analysis

After every interview, we each wrote a memo of our individual perceptions of the interview, followed by discussing key points that the interviewee addressed. All interviews were recorded and transcribed, and NVivo was used as a tool for the data analysis. The transcripts were coded in two iterations inside each case. First, open coding technique was used to create meaningful segments of the data. Then axial coding approach was used to aggregate the open codes into meaningful categories to create connections between them. The categories are both based on the perceived trustworthiness factors and according to what interesting topics came up. In each case, the categories of codes were the following; *Background and context*, *Attitude towards information security work*, *Competence of information security*, *Organising of security work in the organisation and team* and *Cooperation- and trustworthiness factors.*

3.5 Limitations in methodology

Semi-structured interviews are limited in scope. They capture a person's feelings, behaviour, and thoughts at a specific time. In that way, the findings are derived from a personal point of view. If the interview was conducted at another time on another day, their answers might be different[HL01].

Chapter 4
Results

The following chapter elaborates on findings in cases A, B, C and D, which are structured per case. Findings from cases A, B and C are supplemented with a figure that shows the overview of the team structure, see Figure 4.1, 4.2 and 4.3. The blue areas represent the participants from each case, and the participants also have an ID in the figure. The solid lines between the actors represent the contact of communication, and the dotted lines show when the participants are hired at a consultancy company.

We have divided each case section into cooperation between different stakeholders to separate which relationship is discussed. Findings in the cases are partly related to the perceived trustworthiness factors explained in Table 3.1. Other interesting results are elaborated on in the latter of each case.

4.1 Findings from case A

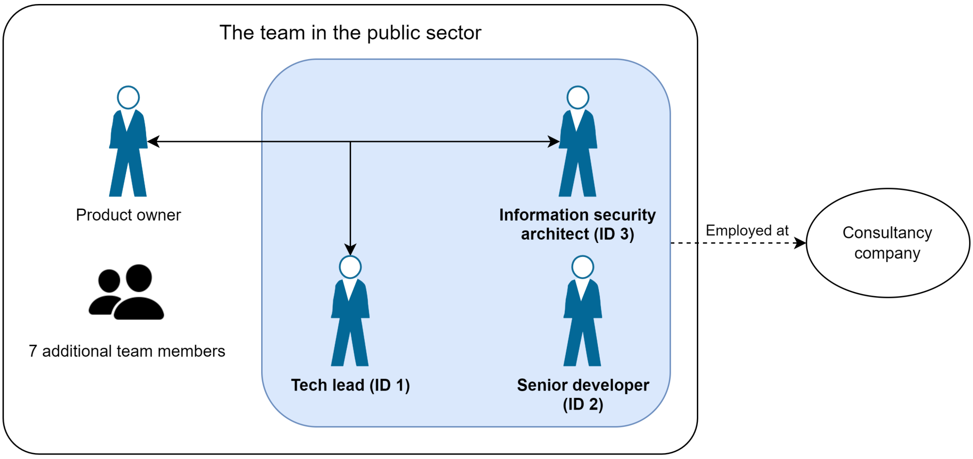
The team in the public sector
Product owner
7 additional team members
Information security architect (ID 3)
Tech lead (ID 1)
Senior developer (ID 2)
Employed at
Consultancy company

Figure 4.1: Overview of security stakeholders and team members in case A.

Figure 4.1 shows an overview of case A and the actors involved. The tech lead was involved in the early communication with the customer. He took part in drafting the proposed solution, as well as being involved in architectural decisions in regards to security. After the work had started on the project, the tech lead transitioned into a developer role. The security architect was less involved during the project's inception but now acts as the security actor in the team. Both the tech lead and the security architect communicate with the project owner.

4.1.1 Cooperation between the security architect and the development team

The following section elaborates on findings regarding cooperation between the security architect and the development team. The findings include experiences related to the project they currently work on and former experiences from previous projects.

Competence

The security architect has an administrative role in the project and keeps the developers' security efforts in check. However, he does not think this will be necessary due to the skills and competency of the developers.

> *"My role is to keep the developers in check, but I mean, it's not really needed because they're really skilled, and some of them work with security all the time, so it's more as a controlling function."*
> (Security Architect, ID 3)

Due to the security architect not having a background as a developer, he will not oversee the team's activities on a coding level but rather advise and control requirements related to privacy and policy.

He also acknowledges that the developers better understand the data flow and how the product functions than those in the controlling roles. In his mind, this knowledge makes the developers' thoughts and ideas carry substantial weight and leads him to consider their opinions when making decisions.

> *"When it comes to the project, the developers' thoughts and ideas carry substantial weight. I mean, they're in the code and have done these things many times before. [...] It's important to air these things, so the more, the better, as far as I'm concerned. So we will all discuss it and reach a solution we all agree is the best."* (Security Architect, ID 3)

The security architect performs frequent risk assessments, and if his assessments do not indicate any prominent issues, the security architect will not interfere with the team's daily tasks. If there were to be issues identified in the risk assessment, the security architect expects the team to resolve the problems. He expects the tech lead, whom he perceives as having considerable security experience, to guide the developers in resolving the issue. The senior developer corroborates this perspective and describes not feeling monitored by the security architect. He characterises the working relationship as more of a cooperation between the two groups rather than the developers being governed by the security actor.

In previous projects, the senior developer has experienced that security practitioners generally lack understanding of developers' work. He has experienced that security teams usually do not operate on a project-to-project basis but across the entire organisation and its projects. As a result, they sometimes lack contextual awareness of the project and knowledge specifically about secure coding practices. Therefore he emphasises that he appreciates working with the tech lead in the current project due to his understanding and knowledge of how the developers work. He also appreciates the tech lead's knowledge of information security.

Feedback culture

Our findings suggest that the feedback culture in the team has positive traits and seems beneficial to the teamwork. The three different views seem to agree on this. The senior developer has no issues with airing concerns related to security in the team or to the security architect. According to the senior developer, this was not the case in his previous project. Here, they did not have somebody in the team with a security role to which they could voice their concerns. The security architect points out that he encourages the developers to ask questions freely and get in touch even outside the regular meetings. To him, it is essential to have open and continuous communication.

The security architect and the tech lead elaborate on how they are conscious when using language to communicate security requirements. They both think adjusting one's speech to be understood when conveying information is essential to avoid misunderstandings. In the current project, the security architect asks follow-up questions when he does not understand something technical and vice-versa when there are things related to laws or regulations that the developers might not understand.

> *"You kind of have to be aware of who you're talking to and not have the expectation that they know exactly the same things as you. Because if they did, they'd have the same role as you, right?"*
>
> (Security Architect, ID 3)

The tech lead has experienced situations in previous projects where there have been misunderstandings between the developers and the customer. In some cases, this has escalated into unprofessional behaviour where team members have yelled at each other about misconceptions. The different vocabularies and languages spoken by the two groups were the cause of the misunderstanding, even though they were technically talking about the same thing.

Information transparency

The three participants agree that information transparency is an essential factor in a good collaboration. The security architect believes that staying up to date and having an open dialogue throughout the project is a success factor for the collaboration with the development teams. He plans to organise a short presentation on privacy and what constitutes personal information for the entire team to be informed on which requirements they need to consider. According to the security architect, when presented with new security requirements from the customer, he will first try to clarify the requirements with the customer. He will then discuss the requirements with the tech lead and get a feel for the possible technical ramifications. The two will then present the requirements to the developers, who may ask them questions.

The tech lead thinks that a success factor for good cooperation is to ensure that documentation is easy to understand and make sure that decisions are well-grounded with the team and controlling actors. If the documentation is well written and simple to understand, the controlling actors will be more capable of knowing what to look for and asking the right questions. He also believes that elaborating on terms and concepts in a way everyone understands within the team, is essential to ensure that all members have a mutual understanding of the concepts related to requirements or technical barriers.

The senior developer thinks that misunderstandings between the developers and security practitioners come from a lack of communication and domain knowledge on behalf of the security practitioners. In past experiences, he has witnessed a misunderstanding that went undetected for a more extended period due to a lack of information transparency.

He also mentions that continuous communication throughout the project is one of the most prominent factors in good collaboration across multidisciplinary roles. The communicating parties should make an effort to be available to answer questions or clarify issues. The possibility to make quick contact is essential.

Task support

The project is still in an early phase, so task support from the security architect is yet to see. At this point in the project, the security architect attends the team's daily stand-up meeting and plans to have weekly meetings where the team can address security-related issues. He tries not to monitor the developers but instead aims for a close working relationship with frequent interaction, where he can answer questions and help the team. He believes that he closely collaborates with the development team in this project.

As a precautionary measure for ensuring that information security is taken care of, the tech lead describes them aiming to have at least two senior developers working on each project. This is also the case for this project. This accommodates task support in that the junior developers can ask questions and get the help they need. In addition, it facilitates self-policing and control of each other's work. According to the tech lead, these measures help make it more difficult for a development team to take shortcuts or not follow best practices.

The senior developer experienced poor task support in his previous project. The team lacked focus on security which led to the team not implementing necessary security measures, leaving the service vulnerable. The senior developer did not know whom to contact when raising the issue. After speaking up, the word spread over time and eventually led to efforts to correct the issue.

> *"After mentioning that there weren't any security measures implemented in the solution, the word eventually spread to someone a bit higher up. They said, 'Okay, maybe we should address this' and eventually, the issue was handled. But at the time, I really wasn't sure who I should contact about those things."* (Senior Developer, ID 2)

Prioritisation of tasks

The security architect has experienced friction with the development team, especially regarding prioritising security tasks weighed up against time and resources implementing new functionality. The developers' focus is on creating a product, while the security architect focuses on creating a secure product, which has been the cause of some friction between the two groups.

> *"[...] then my security requirements can be perceived as extra work for the developers, which can lead to frustration, like 'Do we really have to do this?' "* (Security Architect, ID 3)

On the other hand, the senior developer has not experienced any such issues. However, he has not been in a position of having to prioritise tasks for others. He figures that is why he has not been subject to the issue.

Attitude towards information security

The senior developer sees software security as an exciting field but does not feel the need to devote time to improve his skills in the matter explicitly. The skills he has concerning information security are the accumulated experiences he has gathered throughout his career as a developer. He believes the consensus in the team is that information security is something that needs to be addressed and is not experienced as a burden among the team members. Beyond that, it is not something the team members discuss among themselves.

The consultancy firm has hired people internally who oversees the security efforts in its various projects, independently of the security efforts done in conjunction with the customer. Their purpose is to ensure that each of the teams has all bases covered in terms of security in their projects, and they act as a separate overseeing entity.

4.1.2 Cooperation between the consultants and customer

According to the tech lead, the degree to which the customers want to be involved and take part in decisions varies from customer to customer. Some customers want to be involved throughout the project and have an overview of the progress. Other customers see the developers as experts or advisers, and they almost trust the consultants blindly. Those customers never raise any questions.

In their current project, the customer has one product owner. According to the tech lead, two other employees on the customer side act as the product owner, even though they do not have such a mandate. They cause uncertainty amongst the consultants when they interfere in the project. When the three of them want to prioritise their own tasks, the tech lead feels it leads to the customer seeming less coordinated and united towards those working on the project.

Regarding the development phases of projects, the tech lead has not experienced much annoyance concerning the cooperation between developers and the security personnel on the client-side. This is because the developers are in a position where the customer pays them to do a specific job, regardless of how the developers perceive the job. Even if the developers are to implement something they deem pointless or unnecessary, they still make money each time the customer pays for something.

> *"We often recommend [the customer] to do things efficiently and inexpensive. We make money from it either way. Either we're happy, or they pay us a lot. And sometimes it's both, but it's actually not that often."* (Tech Lead, ID 1)

Keeping commitments

According to the tech lead, ensuring that project deliverables are compliant with the project requirements is the cause of much potential conflict between consultancies and their customers. Suppose the deliverables do not comply with the description. In that case, it could lead to the consultancy firm arguing with the customer about who should cover the cost related to solving the issue. To avoid such a conflict, the tech lead tries to convey the changes to the customer as soon as they occur and to have the requirements altered and approved accordingly. In a previous project, he has experienced presenting the customer with deliverables that were not in line with existing requirements because the consultants believed it was a better solution. Due to the consultancy being late with conveying these changes, the customer became dissatisfied and demanded that the developers fix the issue without additional cost to the customer. This leads to the tech lead emphasising the importance of informing the customer at the earliest opportunity and having alterations signed to avoid disagreements. In cases where issues like this are not resolved, lawyers usually get involved, and the cases end in settlements.

> *"If [the alteration] came as a surprise to them, they might tear up the contract and simply say that 'You are not able to deliver. This is not a satisfactory delivery.' "* (Tech Lead, ID 1)

If the developers notice ambiguous language in the project requirements early, they will try to have the customer sign a revised set of requirements which clears up a potential conflict. The tech lead details some examples of requirements that he would generally want to address before starting a project. Suppose a system is to include a two-factor authentication method where the specific technology is stated in the requirement. Such requirements are especially problematic in the eyes of the tech lead, as they do not affect the solution's level of security but rather lock the requirement to a particular technology. If a more suited technology emerges, the developers can not implement it as the requirement states that another solution is to be used. The tech lead also mentions conflicting requirements as being an issue. The customer might state that the system should display information related to its users while at the same time saying that it should not store personal information.

According to the security architect, it is in the consultancy company's interest to have the customer accept project deliverables and keep its commitments to the customer. Failure to do so leads to consultants working for free or damaging the consultancy's reputation.

Selection and prioritisation of security requirements

Some customers become rather anxious if they perceive that the developers have not started work on a task they deem important. In such cases, according to the tech lead, the developers might try to make it seem like they have started work on the task in question, even though they have not. He also mentions how they actually might start work on the task but at the same time inform the customer that other parts of the project will have to wait. According to the senior developer, in some cases, the customer might choose not to prioritise security if other tasks seem more important.

According to the security architect, agreements between consultancy firms and customers in the public sector often include clauses for daily fines. The consultancy is fined for each day the delivery exceeds the agreed-upon time frame. Due to the possibility of receiving fines, the consultancy firm commonly discusses the prioritisation of tasks with the customer, including security. The security architect believes that if the development team finds itself in a situation where an important security measure is not implemented due to the prioritisation of requirements, they should communicate with the customer and reevaluate its efforts. When prioritising security requirements, the team should choose those requirements that impact on security most and communicate this with the customer. He also stresses that no customer wants to buy an insecure service. It might be in their interest to give the developers more time than receiving software that lacks security.

According to the security architect, those who control the funds on the customer side usually have a tendency toward wanting to spend their money on creating a new service and not necessarily a service that has its bases covered in terms of security and privacy. He has experienced that the customer feels the spending on security is unnecessary.

4.1.3 Cooperation between the development team and the operations team

The tech lead details having had past unpleasant experiences dealing with an operations team. Here, the operations team has acted as the security stakeholder and set the premise for how security is enforced in the project. The tech lead explains that when software is in production, the system can be locked down in a way that makes debugging errors difficult. Being denied access to the system by the operations team and therefore unable to quickly fix such errors is frustrating for the tech lead, especially when they are obligated by the service-level agreement to correct errors within a specific time.

> *"[...] the system is so locked down that you can't access anything. You can't troubleshoot, you can't see any logs, you don't have access to data, and you can't see what's really going wrong. [...] And the developers are just sitting there super frustrated. If the developers know they can safely connect to the live system and retrieve the data they need, they might solve it in 15 minutes, you know? Instead, it might take days and weeks. It is terribly, terribly frustrating."* (Tech Lead, ID 1)

The tech lead has experienced that getting help from the operations team has been such a rigid process that the developers try everything else before contacting them. If the problem persists over an extended period, the two groups can sometimes come to an agreement, where the developers are granted the resources they need. If a more severe incident were to happen, the tech lead believes the incident will be passed on and picked up by upper management relatively quickly. The tech lead thinks the developers will get the necessary access to data if the incident is adequately severe.

4.2 Findings from case B

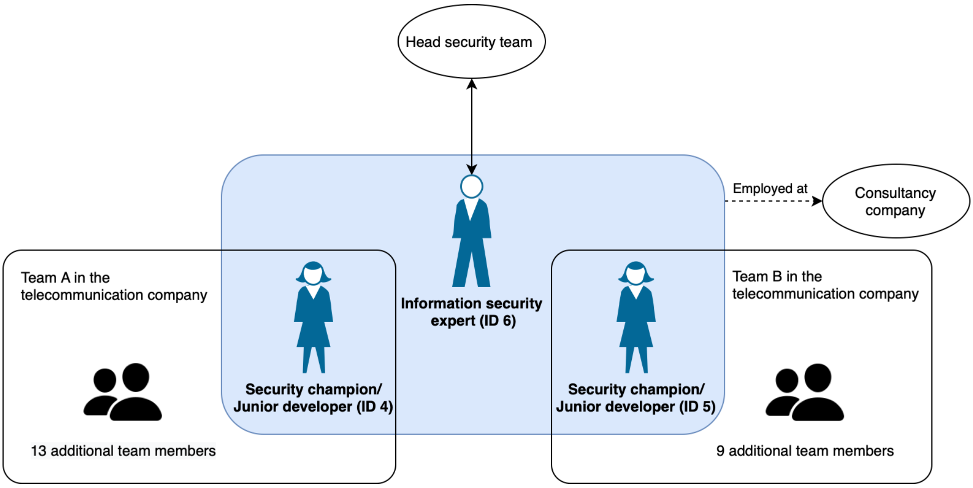
Head security team
Employed at
Consultancy company
Team A in the telecommunication company
Information security expert (ID 6)
Security champion/ Junior developer (ID 4)
Security champion/ Junior developer (ID 5)
Team B in the telecommunication company
13 additional team members
9 additional team members

Figure 4.2: Overview of security stakeholders and team members in case B.

Figure 4.2 shows an overview of case B and the actors involved. The security expert has experience with penetration testing and software development from other projects. He is responsible for ensuring that the IT systems are being developed and maintained in compliance with the customer's rules and guidelines. This includes ensuring the development teams practice secure system development, perform code reviews, use static code analysis tools and remove old functionality. He also makes sure logging is compliant with company policies. He functions as an intermediary between the customer (head security team) and security champions in the development teams. Both teams work on maintaining the old IT system and developing a new system.

4.2.1 Communication and cooperation between the security expert and the development team

Both the security champions and the information security expert describe their communication as sporadic. The security expert describes his job as following up on security issues with the teams and not necessarily being part of the team and working closely with them. This is also the understanding by the two security champions, as they mention that he is not involved in the day-to-day security work and that the teams are free to implement security measures in the way they best see fit. The security expert may advise the team whenever there are bigger security issues, or they have questions. The security expert has monthly meetings with the security department on the customer side. Usually, he contacts the security champions after these meetings to either give feedback or clarify things that have come up in the meeting.

The security expert and the security champions have different opinions on how well their communication works. The security expert describes trying to stay close to the teams and having interpersonal relationships with the team members. He believes this is crucial for a good collaboration between himself and the development teams, and he feels that this is the case for the teams he works with. However, the security champions describe the communication between themselves and the security expert as insufficient. They describe it as being irregular and sometimes random, which causes them to feel that there is no close relationship with the security expert. They explain that they had not been informed adequately in the beginning about the role of the security expert. They describe his role as rather vague. They have also newly taken over the responsibility as security champions in each of the two teams. They admit that they are not entirely confident in their responsibilities and tasks.

> *"We took over the security champion role in February. He then sent us an email to us and the former security champion, so we kind of assumed that the former security champion would reply. But a few days ago, [the security expert] had asked someone else on the team because we had not replied to him. It was a little unclear who had the responsibility. So the communication is not quite there yet."* (Security Champion, ID 4)

They were supposed to be onboarded by a previous security champion, but this fell through. The security champions also explained that this role is relatively new in the organisation. Due to restructuring teams and little experience with the role in the organisation, the security champions did not have a proper onboarding. However, the security expert has a clearer picture of the role. The security expert describes the security champions as his point of contact in each team and someone he communicates with regarding security vulnerabilities and improvements.

There has been a restructuring of the development teams. The security champions used to be a part of the same team, but some teams were split, which caused some changes to their work routines. They acknowledge that this might have affected their start as security champions and how they were informed about the new role. After the original team was divided into two, the security champions felt it had gotten more challenging to know whom to ask about various things. The security champions emphasise the desire for more structured communication with the security expert. Nevertheless, they also admit that they could have been more upfront with their questions concerning their role.

4.2.2 Attitude towards information security

The security expert has experienced developers that have asked why some security requirements are necessary. In his experience, this usually applies to older systems where the system's remaining life span is limited. Then it becomes a trade-off between the risk associated with the vulnerabilities and the cost related to patching. The cost aspect usually weighs heavier in these decisions. However, whenever patching needs to occur, the security champions have experienced certain team members being reluctant to perform security-related tasks. According to them, there is a consensus in the team that security is important and needs to be considered in all the systems. However, not everyone is keen on doing them as they are often seen as one of the less exciting tasks. Still, they emphasise that these tasks are still being executed.

4.2.3 Cooperation between the security expert and the customer

The security expert has frequent contact with the customer's internal security team, communicating with their CISO. He describes the customer's security team as being competent and skilled. Their competence in the security field eliminates his need to alter his language when talking with the security team, which underscores a strong understanding between them. The security expert describes having to alter his language when talking to development teams to give proper and concrete feedback due to them not having the same understanding of information security. He reports having a good understanding of the development team's work day, as he has experience working with both programming and penetration testing. This helps him communicate with those on the customer side and the development teams.

The security expert describes that sometimes the system owners have difficulty in being able to prioritise security tasks versus prioritising the implementation of new functionality. The system owners are ultimately the ones who make the decision, which he and the teams need to follow. Usually, the system owners are the ones who tend to lean towards prioritising new functionality, and sometimes security does have to wait because of this. However, if security is neglected, the security expert would approach the security team on the customer side. They reside higher up in the hierarchy and have more authority to get the security tasks prioritised.

4.3 Findings from case C

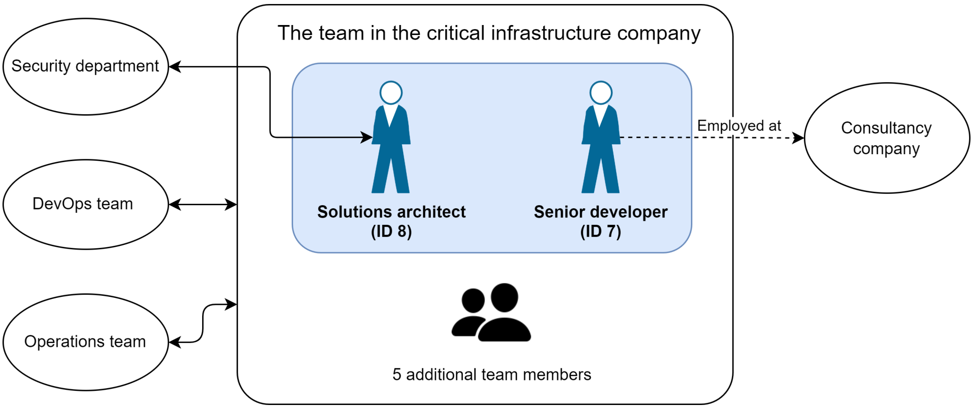
The team in the critical infrastructure company
Security department
DevOps team
Operations team
Solutions architect (ID 8)
Senior developer (ID 7)
Employed at
Consultancy company
5 additional team members

Figure 4.3: Overview of security stakeholders and team members in case C.

Figure 4.3 shows the structure of the team and stakeholders in case C. Findings suggest three main actors who act as relevant stakeholders for giving security requirements for the developing team in this case. The solutions architect sets the project-specific security requirements for the team in conjunction with the project owner and the central security department in the organisation. The solutions architect work as an intermediator between the developing team and the security department. The second stakeholder is the DevOps team that works on centralised deployment scripts in the organisation. The third stakeholder is the operations team, which sets up and maintains firewalls, manages the work computers, and decides the computers' configurations and policies. The project revolves around building an internal system that utilises a cloud data platform provided by a third party. The user must be on the organisation's internal network to access the application.

4.3.1 Security competence in the organisation and development teams

The solutions architect claims there are inadequate security knowledge and resources in the organisation, which he believes affects the security work in the teams. The organisation has a security department, but the resources are scarce, so there are not enough security practitioners who can work team-based with the developers. The solutions architect has background and experience in IT but lacks formal background and knowledge in information security.

The organisation has strategic plans to left-shift the security work in the projects. This initiative has not yet been implemented, and the solutions architect suspects it can take some time before the right resources and structure are present. The organisation has recently introduced security champions and plans to assign them to the teams shortly. It is mentioned that it is essential to give proper training to them, and is it intended to have guidance, courses and frequent meetings with the security department to educate them. This process might take time, but when they have adequate security knowledge, they can help the solutions architect and the rest of the team increase security awareness and competence and share their knowledge.

In this case, privacy concerning the structuring of data is important. The system deals with sensitive data. The team and the solutions architect ensure the creation of secure software by following known and secure patterns for extracting data from different core systems in the organisation. Then the data is migrated onto the data platform. One of the responsibilities of the solutions architect is to identify whenever they stray from the known patterns and then has to evaluate this with the security department.

He emphasises the need to increase the level of knowledge for the employees and the organisation, both because it is needed and because he suspects it will increase the motivation to code in a secure manner.

> *"The second thing is to raise the level of knowledge to the point where it becomes fun, and the developers understand enough. Because that's often the frustration, right? You don't know where errors come from. And no one in the organisation has seen the exact errors as well. It gets tough to work with."* (Solutions Architect, ID 8)

Suppose developers know the reason *why* something fails or why they can't choose a specific path for data extraction. Then, they might find it more motivational to work with it. Another important point is that the developers need the proper knowledge to know when they are moving outside the known patterns and should act cautiously. Sometimes the developers are aware themselves, but other times the solutions architect or the security department are the ones to tell them to act with caution.

The solutions architect believes an essential factor in improving the general attitude towards security work is for the team to acquire enough knowledge about the subject to see the value of creating a robust and secure product. He thinks this will boost the engagement toward secure coding and create occupational pride in coding in a secure manner.

> *"I think we have to work with the attitude. I think that is the most important factor. And you have to get enough knowledge to see the beauty of a thorough and secure product. You have to be able to see whether you have created something good or bad security-wise."*
>
> (Solutions Architect, ID 8)

4.3.2 Attitude towards information security

The solutions architect points out negative attitudes towards security work amongst the team members in the company. He thinks that one of the biggest challenges in the organisation today is that many think of security as a mental obstacle. He thinks many developers believe it is a waste of time and that their job is primarily to produce as much as possible in the shortest time possible. He further explains that this is not the organisation's vision and that most IT organisations want secure systems. He points out that the negative attitude towards secure development could be present due to how security is organised. He puts some of the responsibility onto the organisation because they need to facilitate secure development better. Some of these issues are explained in Subsection 4.3.3. As another example, the security revisions (like risk and vulnerability analysis (RVA-analysis)) occur when projects are nearly finished or when the team have the impression that they are finished. When someone points out that the product is not secure, it will be costly and demotivating for the team.

The senior developer spoke less about the negative attitude towards security in development. Still, a finding from his point of view suggests that when the application resides on an internal network, the security work is not as highly prioritised as for web applications. The attack surface is portrayed as smaller than for a typical web application. Removing deprecated code libraries is not considered urgent, as the risk of incidents is relatively low for internal applications, in his opinion.

> *"It has made our attitude towards security, well, I wouldn't say laid-back, but when something happens now and then, like Node packages getting compromised, we know that unless we screw up pretty bad, we will not be affected by the vulnerabilities that show up."* (Senior Developer, ID 7)

4.3.3 Cooperation between the development team and the operations team

The developer expresses challenges in working and communicating with the operations department. Information transparency is portrayed as low; updates to the computers, removal of chrome extensions and other activities that influence the daily workflow of the teams are not communicated well enough, in the developer's opinion. The updates are usually performed to handle vulnerabilities in existing frameworks or configurations the computers have. This can interfere with the workday, often resulting in lost work time for the developers. They spend long periods trying to debug, only to find out it is not something the developer is in control of. This can cause frustrations:

> *"[...] at the beginning of your workday, you notice that all your chrome extensions have suddenly disappeared from your browser. Because of a security vulnerability, they have deactivated all the extensions. And then you sit there thinking: 'I can't work as a front-end developer without extensions. That simply doesn't work."* (Senior Developer, ID 7)

Situations like these are dealt with in different ways, but seemingly there are frustrations connected with them. The feedback culture is described as almost non-existing in these situations. The developer explains that to get the help he needs, he must go through predetermined channels and receive support tickets. After waiting for answers for a certain amount of time, he might not even meet someone capable of helping. This would then require him to escalate the issues and try contacting them through other channels. He describes his email correspondence with them as "shouting into the void". Another way of getting help is to complain to his manager, who hopefully can take it further to someone willing to help them with the issues. This can take a long time, resulting in a couple of days of lost work in his experience.

According to the developer, part of the reason for the issues with the operations department could be that developers are a new type of user group for this department. Like many companies, the organisation is moving towards becoming an IT organisation. That means technical employees such as developers have unique needs in controlling their computers. They need to know what sort of policies are enforced and when there are changes to these policies. The senior developer points out that some of the challenges regarding collaboration with the traditional operations department could be solved by operating with different kinds of users with other access privileges to lessen its impact on their workflows. The lack of distinction between different types of users reflects in the communication as well:

> *"Regarding communication, there is no distinction between the typical office worker that wants to know as little as possible about their machine and the developers that are dependent on having a certain control and overview of the nitty-gritty parts of their machine. I need to know if my machine is blocking traffic in the firewall because that is essential to whether I can make the data calls I need or test certain things. So that distinction is non-existing, in my opinion."* (Senior Developer, ID 7)

4.3.4 Cooperation between the development team and the solutions architect

The developer draws a sharp line between the operations department giving security requirements and the solutions architect that sets the security requirements in the project. Both the developer and the solutions architect characterise the collaboration between them as working very well. They have been on the same team for almost three years, so they have acquired social bonds and a good work relationship. The developer gives the impression that openness to new ideas and expressing thoughts and worries are welcome in the team.

Task support

Several mechanisms in the team contribute to the task support for security work, and the solutions architect is following them up closely. The solutions architect partakes in the development team's stand-up meetings several times a week to get updates or answer questions. The senior developer characterises his involvement as active towards the team. Part of the task support from the solutions architect includes being proactive, and often it is he who lets the team know when something is not up to code or when they stray off the known patterns. In his opinion, this is often due to a lack of security knowledge in the developing team. Nevertheless, he emphasises that he trusts them:

> *"I really trust a lot of what they do. I know they spend time trying to make it secure, but none of them are security experts. So again, we have to follow the known patterns."* (Solution Architect, ID 8)

RVA-analysis sessions are conducted in the organisation on a quarterly or bi-annual basis. The architect mentions this gives rise to challenges between the developing team and the security department. The RVA-analysis often results in new measures the development teams have to take on, which they often experience as demanding.

> *"Those RVA-analysis sessions never go well, to say it like that. What happens is that someone points out a lot of security requirements that need to be fixed and hands them over to people who already have a lot to do. And it comes a bit like a surprise. So that is not an optimal way to work. [...] You get that huge amount of work that you had not seen coming, and it kind of ruins all your upcoming plans. This creates frustration. Working with secure code could and should be a positive experience, but this way of working does not yield positive experiences."*
> (Solution Architect, ID 8)

He thinks this counteracts the strategy of continuous security because it gives the impression that security is something that can be done four times a year. However, risk analysis should be performed, in his opinion, but rather on a higher level. That way, security implementation can happen continuously in the team. Although he considers the outcomes of the RVA-analysis to create some extra work for the development team, he appreciates the security department's involvement in the correction of the security issues.

4.4 Findings from case D

This section contains findings related to cooperation between developers and security stakeholders from case D. Context and background information about the participants in case D are presented in Table 4.1. For additional participant information, see Table 3.2. The participants stem from different companies and work on separate projects.

Table 4.1: An overview of the participants in case D.

ID	Role	Industry sector	Project context
9	Senior Developer	Contracted out to the banking and insurance sector. The company is amongst the larger ones in Norway's banking and insurance sector. Employed by a mid-sized IT consultancy company in Norway.	The team work on a software application where the bank's customers report insurance claims. She is the team lead and works with front-end development. She has been on the project for a year, and the project has been going on for several years. The team consists of 7 members.
10	Senior Developer	Contracted out to a research institute. Employed by one of the larger IT consultancy companies in Norway. Employed by the same IT consultancy as participants 11 and 12.	The team works on a project where they modernise a research model for hydropower storage reservoirs. The project started a year ago, and he has been on the project since the start. It is an internal application accessed only from the intranet of the company. The team consists of 4 members.
11	Senior Developer	Previously contracted out to the petroleum sector. Currently contracted out to another sector. Employed by the same IT consultancy as participants 10 and 12.	Worked as a developer in the petroleum company in a cross-functional team with designers, QA-personnel and developers. He was there for two years. The team consisted of 20 members.

Continued on next page

Table 4.1 – *Continued from previous page*

ID	Role	Industry sector	Project context
12	Senior Developer	Contracted out to the health department. Employed by the same IT consultancy as participants 10 and 11.	The team work on an application for monitoring the medical use of radiation. His daily work mostly involves data analysis, structuring and advising on data models, and some development. He works as the sole developer in a team that consists of 6 members.
13	Senior Developer	Contracted out to the banking and insurance sector. Employed by a mid-sized IT consultancy company in Norway.	He works with front-end development in a team that works on IT systems regarding consumer loan applications. The team consists of 10 members.
14	IT Security Specialist	Employed in the banking and insurance sector. The company is one of the larger banks in Europe.	He is a part of an intermediating team between the cyber security department in the company and around 100 development teams across different countries in Europe. He works with security tools which company policy requires the developers to use. The cyber security department sets the security requirements for the development teams, and he helps mediate between them. He is situated in Denmark.

4.4.1 Information security competence

The reported strategies to raise security competence in the teams vary from participant to participant. One developer (ID 10) mentioned that there are courses and seminars available from the consultancy company he works at, but he rarely goes to them. He is interested in learning more about information security but mentions that there are many other courses more focused on programming that are worth attending instead.

Another candidate (ID 11) states that most of his knowledge about software security is what he has picked up throughout his career.

Two of the developers (ID 9, 11) mention that it is often expected that developers know software security principles and expected that they practice secure coding. They are, however, divided on whether developers actually know these principles. One of them argues that education has more focus on secure coding nowadays, and therefore most developers know security principles and practice them in their work:

> *"It is not something we talk that much about in the team. But in my experience, most developers today do their job with software security in mind. It is not a separate thing you consider in addition to coding. It becomes a part of the problem-solving."* (Developer, ID 9)

The other argues that is it naive to assume all developers have these skills. A common trait they both mention is that secure coding is rarely talked about, and the outspoken enthusiasm is low.

> *"This is one of my frustrations. There is never any focus on the fact that the developer should know security. I have not once been to an interview and had the question 'How good are you at security?' come up. It is just expected that as a developer, you practice secure code. And that is not true, of course. Not all developers know these principles, but I am not going to claim that I know that much either."* (Developer, ID 11)

This is corroborated by participant 13, who mentions that security is hardly ever spoken about in the team.

The IT security specialist (ID 14) believes that developers are often keen on making secure software solutions, but the competence might not always be there. His job is to make sure that development teams think security first, and by helping them out, he believes their security knowledge would increase over time. He also states that a common demand in the market is that developers can practice secure coding. However, he is more sceptical about the possibility of security practitioners learning more about development:

> *'Seeing from the security perspective, I don't think they are moving into understanding the developers. There is so much going on in the security field that the development field is just out of mind. We don't have time or resources in our minds to take care of how developers actually do stuff."* (IT Security Specialist, ID 14)

4.4.2 Lack of experience of working with security teams or practitioners

Three of the developers (ID 9, 12, 13) in case D expressed that they had little to no experience working with security teams or security people in their previous projects. There are different reasons mentioned for why this is the case. Two of them are primarily working with front-end development. One mentions that:

> *"I don't really consider information security that much. It is not something that I think much about."* (Developer, ID 13)

He also mentions that he does not communicate with any security practitioners but speculates that the developers working on the core system have more communication with them. The other developer is familiar with static code analysis tools but has limited experience cooperating with security practitioners. The third participant currently works on a project where privacy around sensitive health information is a big priority but has not cooperated with the security people or legal team that sets the premises for this data.

4.4.3 The intermediary role between the security department and the development teams

Most of the participants in case D either has worked or are currently working on projects that do not have a mediator role between the security department and the development teams. The only candidate in case D who operates with the usage of a mediator is the information security specialist, who performs the role himself. He describes his role as making sure that developers think security. He tries to pull them toward making software secure by design to avoid them having to go back and fix vulnerabilities at later stages when the applications are out in production. A penetration test towards the end of the development would probably reveal some vulnerabilities, which the developers would need to go back and fix. This happened to another participant in case D (ID 10). He worked on a project where they made a web application, and no one thought of software security in the project. A penetration tester came in towards the end of the project and found major vulnerabilities, which created a great deal of extra work. In retrospect, he believes that if they had had someone in the team that could have an extra focus on security, a lot of this extra work would not have occurred.

The information security specialist explains that by hiring someone in his role, he can improve the understanding across the cyber security department and the development teams. When the development team lacks security abilities inside the team, they depend on someone else to advise them. If the ones advising them are the same as the ones enforcing the policies (the cyber security department), then the developers do not have someone to represent their perspective and try to push the others into a gentle introduction of new policies, for instance. In his opinion, this is why his role is important; he can represent both roles in meeting the other part. It might also be difficult for the development team to know if the policies enforced are actually demanded by law or if the rules only apply to the bank industry. Developers might come from a different area or sector, and when they start in the bank, they might have to work more rigidly because the bank needs to comply with regulations. Whenever there are new policies to be implemented, his role is valuable in communicating this to the developers:

> *"I have a certain say in cyber security, because they also use me the other way around. If I have any input or something that I can see would be a problem when they try to enforce something new. So I also try to mingle a bit so that the developers get a little and the security gets a little, so they can actually meet in the middle, and then perhaps say we introduce the policy in small bits. So the entire policy might be too much to introduce from one day to another. In a year from now, that one will be fully implemented. [...] I try to soften the blow from new requirements, so they are not overburdened by something new."*
>
> (Information Security Specialist, ID 14)

He believes introducing a mediator like himself is part of the solution for good cooperation between security and development.

4.4.4 Security task support

Task support efforts for information security are portrayed differently by the information security specialist and the developers. Candidate 10 explained there was no task support on software security because of the lack of resources available in the project until they had hired a penetration tester late in the project. The report from the penetration test showed vulnerabilities prioritised in a list, including an explanation of how they could be exploited and why it was important to fix them. The experience of cooperating with the penetration tester was positive in his case, and he felt he received good support. However, later, prioritising fixing vulnerabilities versus developing new features became an issue. The project owners focused more on developing new features in the solution than fixing the security vulnerabilities, so only the most critical security measures were prioritised.

The information security specialist tells a different tale of open forums of task support. Developers are encouraged to initiate contact when they have issues or questions regarding either tools, implementation of security requirements or company policies. The facilitating team that the IT security specialist is a member of facilitates around 100 development teams. Therefore, the development teams would have to contact when they need support and have questions, not the other way around. He usually does not do follow-ups with the teams unless he has a personal interest in how someone has solved a security issue. The security team have weekly meetings, and developers are welcome to join the meetings when needed. To help the teams out during the pandemic, they also introduced open forums on different channels on Microsoft Teams, where everyone can post issues or questions regarding security. Often there are many questions about the policies, and multiple people would ask the same question. In his opinion, the information transparency that Teams channels can provide effectively helps the developers. All the members of the channels usually contribute to answering questions as well, not just the people who administrate the channels. His impression is that developers appreciate this way of task support because it is an easy way to make contact if anything were to come up.

4.4.5 Challenges between the development teams and the security stakeholders

Participant 11 from case D explains a situation where the cooperation between the development team and the security stakeholders was non-existing. Here, three security stakeholders provided requirements to the team. The first was a back-end team that managed the Microsoft Azure policies. The second stakeholder was a team that worked with operational security. The third stakeholder was the development team themselves, as they had to make many security decisions and be held accountable for the decisions later. The inter-team cooperation was minimal, and they lacked communication structure and had infrequent interactions and many opposing needs that created frustrations between them. He explained that the operational security team would send out fraudulent scam emails to the employees to find out who would click the links in the email and who would not. If the employees clicked the links and provided requested information, their jobs might be on the line.

The developer describes the back-end team's task support efforts as lacking. There were few means of communication, and the back-end team would often perform changes to the platforms the developers were using without informing them, interfering with their work. There was little understanding from both sides of why the other part had to do things a certain way. Often these frustrations would end up in aggressive emails going back and forth, which would constitute a case of a negative feedback culture. This characterised the working relationship during the first period of the project:

> *"We sent an angry email where we wondered what the **** they were doing, and the tone was pretty aggressive. And obviously they did not appreciate that. So we definitely started off on the wrong foot. It did become clear to us that they were pretty stressed out, the back-end team. When we understood that, we realised that we had made a mistake, and that this was not the way to approach them."* (Developer, ID 11)

One of the consequences of the strict policies and absence of cooperation was that the developers started to take shortcuts. The process of getting access to the necessary data they needed to be able to do their job was very cumbersome. This data was sensitive, and getting access could take a long time. The lengthy process gradually frustrated the developers, who ended up bending the rules of where data should be stored. They disregarded the security policy as they deemed it too cumbersome to work with. However, he claims they would never jeopardise the security of the data on the line. He says:

> *"I am a hundred percent sure that there were other teams who took the same shortcuts as us, because you're ultimately there to do a job. And if you're twiddling your thumbs long enough, you'll get so frustrated that you'll eventually start taking shortcuts."* (Developer, ID 11)

He speculates why the conflict between the back-end team and the development teams happened. The development team expected everything to be up to date when they came in, so they could start coding. The second factor is that Microsoft Azure was relatively newly implemented in the company, and the company had not fully implemented this into their existing IT solutions. The back-end team had their hands full with Azure policies and handling around 50 development teams. There are similar challenges in this situation compared to the challenges explained in Subsection 4.3.3 with regards to the operations team from case C. There is also a lack of information transparency in this situation in case D. The developer has experienced the back-end team deleting a lot of the developers' code without telling them. He speculates that the reason is that their code was not compliant with the Azure policies. He also recognises that the back-end team worked with many teams with lots of vulnerable code, so they had to remove vulnerabilities as swiftly as possible. This came at the expense of information transparency and communication. Physical distance was also an element in the conflict. The only way they communicated was through email, and they never actually met in person. They also did not have a mediator that could help clarify needs and have a foot in each team. He emphasises that if they had had this role in the team, they probably would have avoided the conflict escalating this far.

4.4.6 Attitudes towards information security

One developer is currently working on a project with non-existing security requirements. He is part of a team working on an application for internal use in a company which serves a relatively small group of end-users, and hence are security requirements not explicitly mentioned:

> *"In my team, we are not doing any security activities with regards to the application. It's in part due to the nature of the project. It is an application running on internal servers in the company, and it is not exposed to external network such as the Internet."* (Developer, ID 10)

Chapter 5 Discussion

The discussion is structured after the research questions. First, RQ1.1: *What challenges related to trustworthiness factors exist between the team and the information security stakeholder?* is answered in Section 5.1. Then RQ1.2: *If challenges are present - what are the consequences related to information security work in the team?* is answered in Section 5.2. Then, we answer RQ1: *How do the trust relationship between the software development team and the information security stakeholder affect the information security work in the team?* in Section 5.3. Next, RQ1.3: *What are some recommendations for building trust between the team and the information security stakeholder?* is presented. At last, the limitations of the study are discussed.

5.1 The challenges related to the main trustworthiness factors

This section explores the identified challenges in relation to the five main perceived trustworthiness categories of which trust can be divided. These factors are ability, benevolence, integrity, transparency and predictability and are the common trust factors based on well-known trust literature [BHHH20; MDS95; FG12]. According to the literature, trust emerges in teams where these factors are present.

5.1.1 Ability

Ability refers to the competence and skills of the trustee in a given domain [MDS95]. This main trustworthiness factor comprises different constructs, such as competence, feedback culture, friendliness and participation [BHHH20]. We have identified challenges related to some of these factors.

Security stakeholder as trustor and development team as trustee 👤→👥

Competence (or the lack of it) is the most common challenge we have identified within the ability factor. It is further explained as the trustor's view of how competent the trustee is in a specific domain. In this case, the domain-specific competence is the security actors' competence in security or managerial skills and the development team's competence in both development and software security. In case C, the trustee (the development team) is viewed with less competence in security by the solutions architect. He regards them as having great competence in application development. However, he views the organisation as lacking security competence, which manifests in the team's ability to work with security.

We see positive traits in case A regarding competence. The security architect believes that the relevant information security competence resides inside the team. They have different backgrounds in security. While the security stakeholder has a background in legal security and privacy, he trusts the development team to be competent in implementing security requirements.

Development team as trustor and security stakeholder as trustee 👥→👤

Another perspective is to look at the development team as the trustor and the security stakeholder as the trustee. Generally, we have identified more issues in this direction of analysis, especially regarding the cooperation between the development team and the operations team. We have identified this in case A, C and D.

We have identified common characteristics in the cooperation between the development team and the operations team: negative feedback culture, lack of understanding of the other party's needs, lack of friendliness, and inadequate communication patterns. Issues between the development and operations teams have been pointed out in earlier literature on challenges in DevOps [DMC19].

A lack of security competence among developers has been identified in the literature as one of the most encountered challenges in DevOps or DevSecOps. Developers often lack the necessary security training and education to be able to fulfil the purposes of DevSecOps [RZBS21; DMC19]. It is also stated more often that developers need to acquire more knowledge about security than security people need to acquire knowledge about development [RZBS21]. The IT security specialist in case D echoes this view. He states that the security field is constantly moving forward and that there is not enough time for them to educate themselves in development. A key feature of agile is the cross-functional team, where team members have expertise in one area and are somewhat knowledgeable in the other disciplines. For developers, this means being experts at programming and coding and being knowledgeable of software security principles.

The developers who partook in our study were asked whether they believed the security practitioners had knowledge about the developers' usual work tasks. Most of the developers suggested that security practitioners often lack technical knowledge related to software development and, therefore, lack an understanding of how security requirements are implemented. This contributes to the security stakeholder being viewed as less trustworthy concerning ability. However, in cases A and B, the security stakeholder has earlier experience with software development. The teams reported appreciating the extra understanding they have in their field, which contributes to more trustworthiness with regard to ability.

We asked the participants in selection 2, the security stakeholders if they ever have experienced the development teams asking them if their security requirements were necessary or if the developers ever disagreed with them. The results indicate that some developers might ask if specific requirements are necessary. However, most of them usually understand that they have to fulfil the security requirements, for instance, dictated in policies. Violation of such policies could lead to a project delivery not being fulfilled or might even constitute a breach of law. The challenge here seems to be that they do not understand why some requirements need to be in place, which often leads to a feeling that the requirements lead to unnecessary work and makes them have to make a code "detour" in their development.

Security champions have been highlighted as a recommendation to build trust between the security teams and the development teams [SC20]. We encountered the usage of security champions in both cases B and C, but none of them seemingly have yet managed to utilise the role to its intended potential. In case B, the security expert views the security champions as his contact point in each of the six teams he is responsible for. The purpose of the role, in this case, is to ease the communication and build a structure around communication of security issues, but there are still issues encountered here. These challenges include onboarding and clarifications of expectations of the role. In case C, the organisation is in the process of left shifting security, which in their case involves implementing security champions. However, it requires time and the right resources to give value to the role. He also speculates whether or not the organisation is willing to invest money into raising security knowledge. The discussion of how much money to spend on security measures is also a somewhat typical trait for organisations that gradually want to left-shift security [CLH+].

5.1.2 Benevolence

A trustee's benevolence refers to its selfless inclination towards the trustor. A trustee acting benevolently will look after the trustor and not act solely in its own interest. Benevolence comprises several trustworthiness factors, such as task support and loyalty. Task support refers to the goodwill team members show each other in terms of helping with completing tasks. Our findings have not identified issues related to benevolence but rather positive traits.

Security stakeholder as trustor and development team as trustee 👤→👥

We do not see any challenges related to benevolence in the other direction of analysis. With the development team acting as the trustee, our findings indicate general goodwill from the team to help clarify issues, give status updates and answer questions that the security stakeholder might have. Such is the situation for cases A and C.

We generally see strong loyalty tendencies inside the teams. In case A, this might have to do with the participants coming from the same consultancy company, as they share a work culture from earlier projects. A common trait among participants in selection 2 is reporting members of their teams backing each other when justifying technical decisions, either to an external security stakeholder or a project owner. This is also a characteristic of autonomous teams, typical for both agile and DevOps teams.

Although loyalty bears primarily positive connotations, a possible negative outcome related to team loyalty can be identified from the interviews. In case D, a participant (ID 11) reported having a high degree of loyalty inside the development team. The team experienced conflict with a backend team, which acted as a security stakeholder on the customer side. The conflict is detailed in Subsection 4.4.5. The team's autonomy and the high degree of internal loyalty might have contributed to the escalation of the conflict with the backend team. Through the developer's comments, it seems as though there was a degree of "us versus them" mentality, which may have caused the confrontation of the operations team as harsher than it needed to be.

Development team as trustor and security stakeholder as trustee 👥→👤

In cases A and C, we find positive traits of benevolence with the perspective of the trustor being the development team and the security architect being the trustee. In Case C, the solutions architect tries to help the development team and be proactive regarding security. However, he does not consider himself thoroughly skilled in information security. Suppose the solutions architect lacks competence in information security. In that case, it could be argued that this lack of ability would negatively affect his ability to provide task support and therefore affect the trust relationship. However, task support refers to the trustee's willingness to be of help to the trustor. Even though the solutions architect might lack some ability related to information security, it seems as though he actively tries to assist the team, and this sign of benevolence contributes to the emergence of trust.

5.1.3 Predictability

Team members who show predictability through consistent behaviours and timely fulfilment of work tasks are reported to be more trusted [BHHH20]. If someone repeatedly fails to meet deadlines, this will negatively affect trust relationships. This main category of trustworthiness factors consists of keeping commitments, availability and consistency, and there were challenges identified with all of them.

Security stakeholder as trustor and development team as trustee 👤→👥

The tech lead, in case A, talked at length about how important it is for them to make sure they agree with the customer about deliverables and deadlines. In this case, we perceive the customer as the party who sets the premise for both functional and security requirements. We can consider the customer the trustor and consultancy the trustee. By ensuring there are no ambiguities surrounding the project, the consultancy might be better able to avoid situations where the customer is dissatisfied with the deliverables. The consultancy seems to be very concerned with delivering on time and communicating changes in the project with the customer as soon as they appear. By ensuring they follow the contract and keep their commitments, the consultancy would be able to improve its trustworthiness through its predictable behaviour. According to the tech lead, it seems as though the customer can terminate the contract if there were to be disagreements. Therefore, being thorough in defining deliverables might be a way for the consultancy to protect themselves from this happening.

Development team as trustor and security stakeholder as trustee 👥→👤

Having the habit of being available is considered to be contributing to the trustworthiness of a trustee. Team members who are quick to reply to emails or messages, as well as being socially present, are reported to be more trusted [BHHH20]. We have identified some issues regarding availability when the security stakeholder is a separate entity, such as an operations team. Such issues have appeared in cases A, C and D, where the developers have described having issues working with the operations team. Although their experiences were somewhat varied, there were a few elements that all participants mentioned. They all expressed frustrations related to the communication with their respective operations team. They detail challenges both in regards to availability and information transparency.

One developer described not receiving replies to his emails from the operations team, which indicates issues related to availability. These emails could be regarding questions for clarification or waiting for access to specific data they needed to do their job. When the security stakeholder is a separate entity from the development team, it appears to be a less personal connection between the groups. One developer describes it as follows:

> *"Automatically, it makes things more problematic. The distance between us degrades us to a small email instead of the people behind the screens."*
> (Developer, ID 11)

In the cases where development teams and security stakeholders work closely together, there has not been much to suggest that availability has been an issue in either direction. On the contrary, when the security stakeholder participates in some of the team's daily stand-ups, the development team experience quick feedback. This has been reported in cases A and C and by some interviewees in case D.

We have also identified challenges concerning consistency which is the third trustworthiness factor related to predictability. Being consistent adds to the trustee's predictable behaviour. In addition to one project owner being tied to the project in case A, two additional employees on the customer side also act as product owners. Despite not having the official authority or mandate for such a role, they still get involved and provide the developers with their wishes for the project. Providing the development team with requirements through three different employees, each with their distinct priorities, the customer is perceived as acting inconsistently towards the development team. If a trustee displays consistent behaviour over time, it will lead to the trustor knowing how they will react and build trust [BHHH20]. If the trustee behaves inconsistently towards the trustor, it will create uncertainty and reduce trust.

Another instance of inconsistent behaviour was experienced by a developer (ID 11) in case D. Here, the customer's backend team occasionally sent their employees forged scam emails. Some of these emails prompt recipients to enter their work username and password. They would then let the employees know what had happened. According to the developer, an employee entering its credentials would, in some cases, be grounds for the termination of their work contract. Although phishing simulation might be a suitable security measure in the company, these occasional emails would constitute inconsistent behaviours, reducing the trust the developers have in the backend team. Phishing simulation is a well-known technique for cyber security training in big companies [JGST20]. However, its usual purpose is either by the company's leadership to map out the cyber security knowledge of the company or to encourage the employee to raise their awareness of phishing emails. Using phishing simulation as a measure for the organisation to determine what employers can be trusted and not might counteract the overall goal of raising awareness. Organisations that utilise disciplinary actions as a response to employees repeatedly failing the phishing test can create distrust between the employees and the organisation [KR19].

Establishing standards and expectations for communication channels between the development team and the security actor contributes to predictable behaviour in both analysis directions. We have identified this challenge with communication structure in several cases. Open and robust communication between the members of the development team and the security stakeholder is essential to make the collaboration work, as well as it is an important factor for trust to emerge [MŠ07]. The lack of standards and structure for how the development team and the security stakeholder communicate has made their collaboration complex and unpredictable. Challenges related to this have been found in cases B, C and D. Having clear expectations about where to approach when encountering security issues are recommended [RZBS21] to improve collaboration. This promotes short feedback cycles and more effective communication. Using email as a communication medium is discouraged because it will often be overlooked amongst other emails. We have encountered emails being used as a communication medium in cases C and D (the challenges between the development team and the operations team) and working poorly. Slack channels and Microsoft Teams channels are examples of communication mediums that have been suggested to work better [SC20]. Two participants, in case D, mention they are happy with using slack channels or Teams channels with different channel topics as a communication medium.

5.1.4 Integrity

Integrity refers to how the trustor perceives the trustee's beliefs and principles. This trust dimension includes the factors confidentiality and ethical values [BHHH20]. If these values align with the trustor's perspective, then the trustor is more inclined to trust the trustee [MDS95]. Another way to explain integrity is whether the person does what they say they will do. If there is a high correlation between their word and their actions, a trustor is more likely to trust the trustee [CS09].

Development team as trustor and security stakeholder as trustee 👥 → 👤

We have not identified many challenges related to integrity between the development teams and the security stakeholder. Our findings suggest that challenges regarding integrity might instead happen between the client and consultancy company. It seems to be an essential factor when consultancy companies try to win project offers. The consultancy company needs to come across as trustworthy in the eyes of the client, and they need to make them believe that they have the right resources in the consultancy company to fulfil the client's project. Participant ID 1 elaborated on how a process like this can happen. Challenges related to keeping commitments are also relevant here, as explained in Subsection 4.1.2, but this affects the integrity of the consultancy company. The consultancy company usually has to sign a set of requirements (including security requirements) the client specifies at the beginning of the project to show they can deliver what they need. After winning an offer, they have to manage the client's expectations to avoid getting into conflicts about the consultants not being able to deliver on agreed requirements. In other words, they need to avoid the discussion of "you did not deliver on something you said you would".

5.1.5 Transparency

Transparency refers to the openness of knowledge management and sharing of relevant information within the teams [BHHH20]. Transparency also comprises the willingness to share private information so that team members can confide in each other. This trust dimension also comprises responsibility assignment, which refers to clear task assignments. We have identified challenges regarding responsibility assignment and information transparency.

All of the interviewees have implemented most of the critical features of working and collaborating in DevOps and agile; stand-ups, sprint planning, sprint retro and sprint demos. These features are designed to offer a framework for sharing relevant information with the team and other parts. One of the critical concepts in DevOps is to share relevant information with the team and to counteract working in information silos [Pup21].

Development team as trustor and security stakeholder as trustee 👥→👤

One of the interviewees in case D experienced that the daily stand-ups are not working as they should. The organisation he was contracted to underwent a more extensive reorganisation, resulting in the team being altered. The reorganisation led to those participating in the stand-ups being located across several countries and working on separate parts of an IT system. This caused these meetings to become meaningless for the other people that worked on another part of the system. He describes the stand-up meetings to work better prior to the reorganisation, and he describes the current stand-ups in the following way:

> *"It is pretty boring to go to these stand-up meetings because you have to sit there and listen to the issues they have in other countries. And then you have to tell them about your own issues, but nobody really cares because they have nothing to do with it whatsoever."* (Developer, ID 13)

He even felt so disconnected that he chose to leave the consultancy company and, as a result, the customer at which he worked.

Challenges related to responsibility assignment and information transparency occur in case B regarding the security champions. They both report having a vague understanding of the security stakeholder's role at the beginning of their employment. They also reported having a vague understanding of their role when first assigned due to not receiving any onboarding for the role as security champions. One of the characteristics of this case is that the developers are junior developers and newly graduated from university. They initially had some uncertainties about whom to ask about different issues. This aspect might influence many of their answers and explain why they experienced uncertainties. At the same time, building trust is often one of the most critical factors in the early phases of a team's construction. Furthermore, two factors contribute to challenges; First, the two teams in case B are newly formed after restructuring the old teams. Second, they are junior developers, and this is their first contracted project. The new team structure seems to have affected both the trust's transparency and ability dimensions. Having team members socialise is one of the recommended measures to build trust in a new team [TSS22]. According to the security champions, the consultancy firm they are employed at arranged a project kickoff to let the team members get to know each other. They were happy about these events. The security stakeholder reported being satisfied with the information transparency between himself and the development team. When considering the perspective of the security stakeholder as the trustor and development team at the trustee, there seems to be a trust foundation.

5.1.6 The predominance of challenges related to security stakeholders as trustees

We observe more issues related to trustworthiness factors in scenarios where the development teams act as the trustor and the security stakeholder act as the trustee. A possible reason why this is the case is that more of the study participants are developers than security stakeholders. As a result of more developers than security stakeholders taking part in the study, there will be more experiences detailing both negative and positive aspects related to trustworthiness factors towards the security stakeholder. Another aspect that may impact the disparity is that the security stakeholders' decisions had more impact on the developers' workday than vice versa. By mandating the developers to implement specific security measures, it seems likely that the role is more prone to scrutiny, which would result in more negative trustworthiness factors being detailed by developers.

5.2 The consequences related to information security work

Security work in the team here relates to the collaboration and processes which aims at bettering IT security, as further explained in Chapter 1. This section first presents the consequences of the challenges that affect the process of working with information security in the teams. Second, we discuss how this work process affects the overall information security of the final product created by the teams.

5.2.1 Process related consequences

The trustworthiness factors are human-oriented and influence how work and cooperation processes are performed. The challenges identified in Section 5.1 have had consequences on the information security work in the teams. They vary in different ways, but generally, they negatively impact working with information security in the teams.

Taking shortcuts

Two developers report taking shortcuts by bypassing security measures in their daily work. They explain that the operations teams in their respective organisations create such a cumbersome workday for them, leaving them no choice but to take shortcuts related to security requirements in order to be able to do their job. If they were to follow policies rigidly, they claim they would spend several days performing small tasks. One of them who performs programming on a Windows machine (due to company policies) states the following:

> *"As a developer, I understand why you would have an antivirus running on your Windows machine, but this makes us push the boundaries or find loopholes. I started using Windows Subsystem for Linux as soon as that became available. One of the benefits is that the antivirus program does not know what happens on the virtual machine. So things run so much quicker. You might call it a loophole, but I will never tell anyone about it. I have no incentive to tell anyone that the antivirus program does not operate there."* (Developer, ID 7)

One of the developers stated that they would never compromise the information security in any way, even though they take shortcuts. Nevertheless, disobeying the policies might put the data at risk even though someone strongly believes that the data is not risked. They both have not experienced any security incidents. When asked why they thought that way, they both claim they have enough security competence to know when the data are at risk. Some of the participants have mentioned luck as a factor in why they have not experienced any security incidents.

Tendency to report security issues

Part of a thorough information security incident management is good communication between several stakeholders. When the organisations do not define communication standards between the relevant stakeholders, this can lead to maintaining information silos as a problem [AHR12]. One of the consequences of not having thorough standards and expectations for communicating security issues might be a lower rate of reporting incidents when they happen. We have not seen evidence suggesting security incidents are not being reported (except for the shortcuts some developers take). However, we have seen developers reporting uncertainty about whom they should report incidents. If communications guidelines were in place, this could have reduced frustrations related to the uncertainty and the time spent trying to reach the right stakeholder.

Time consuming work

A common consequence related to the trustworthiness factors on the information security work in the teams is the time involved in collaborating with other stakeholders. As we encounter in our material (Subsections 4.3.3 and 4.4.5), some developers would have to wait for days either trying to fix issues introduced in an update from the operations team without them knowing or would have to wait a long time to be granted access to some data material they require for their jobs. This has been one of the largest contributors to developers becoming frustrated with information security work. However, some companies have the financial ability to have their employees or contracted consultants instead wait the time it takes to follow the company's security procedures rather than risking security incidents. Security resources and training are expensive, and sometimes the companies are unwilling to prioritise security to the extent security professionals deem necessary.

Lack of motivation to engage in security work

A lack of motivation to perform security work in the team is another consequence which can be linked to the challenges of trustworthiness factors. Our findings suggest that lack of competence combined with poor ability, predictability and transparency is part of the reason for this. When team members experience a poor collaboration with security stakeholders, such as an operations team in some instances, they might feel that secure coding practices are cumbersome and demotivating. This collaboration might further lead to developers becoming less motivated. The solutions architect from case C strongly believes that if one can raise a developer's competence in information security, it will spark developers' interest in the field and help motivate them to create secure software. Developers need to feel a sense of pride in creating a secure product. In order to achieve this, the organisation needs to implement more security training, which is a well-known measure from the literature [SMSJ04; AC19].

5.2.2 The security of the final product as a consequence of the process

As a limiting factor, we are not able to correlate the process of working with information security and how secure the resulting products are in cases A, B, C and D. In addition to the scope of the study making this difficult, the majority of projects the participants work on are either not finished or yet in production. However, we can look at the interviewee's earlier experiences of security incidents on services deployed in production.

Overall, our findings suggest that the consequences of challenges related to perceived trustworthiness factors are more apparent in the information security work that a team does rather than in the actual security of the final product. By nature, the trustworthiness factors are closely related to cooperation factors and therefore naturally focus more on human-related factors rather than the security of the resulting product. However, that raises the question of how vital the human process is for the security of the final product.

Thorough work processes regarding information security intend to serve the ultimate goal of making software applications and services more robust and secure and limiting security incidents. DevSecOps is an approach to putting security on the agenda for organisations and development teams. Although none of the teams identified themselves as working with DevSecOps, some participants practised some of its features, including automated deployment pipelines and left-shifting security. Left-shifting security is mentioned in cases B and C as something they strive to accomplish in their respective organisations but have not fully achieved yet. Implementing security champions in the organisations is recommended by BSIMM and is also present in cases B and C.

None of the participants reported having experienced security incidents such as data breaches or cyber-attacks. Four interviewees report having experience with being informed of software vulnerabilities and handling them. Three were categorised as minor vulnerabilities, while one was categorised as major. The most significant vulnerability occurred in case B and was discovered in a logging library, which affected several parts of their IT systems. Leaders further up in the organisation (e.i. the CISO) created a war room to understand the extent of the vulnerability. The security expert got in touch with the security champions in all the teams he managed to map out affected parts of their systems. He described their incident response as quick, effective and with good communication. The vulnerabilities were seemingly patched before the vulnerability was exploited. These types of incidents have the potential of being either a contributor to the emergence of trust between the involved parties or create distrust amongst them if management is performed in a weak manner [BHHH20]. In this case, we encountered positive traits of competence and feedback culture (which suggests the presence of ability), availability (which suggests the presence of predictability) and responsibility assignment and information transparency (which suggests the presence of transparency). This suggests that successful incident management contributed to the emergence of trust in this case.

5.3 Trust relationships' impact on security work

To help answer part of the main research question, we have identified the challenges and evaluated the consequences related to the identified trustworthiness factors, covered in Section 5.1 and 5.2. This section will provide an overall evaluation of the trust relationships discovered in case A, B and C and how these trust relationships have affected the work with information security. The cases are discussed individually. First, we identify risk-taking behaviours inside the case between the development team and security stakeholder. Then we look at what perceived trustworthiness factors we have encountered in the case, to map out the trust relationship between the two actors. Recall, the overview of both risk-taking behaviour and perceived trustworthiness factors is presented in Figure 2.2. Finally, we will discuss how the trust relationship has affected the information security work. Table 5.1 shows an overview of the identified perceived trustworthiness factors as well as identified risk-taking behaviour in case A, B and C.

Table 5.1: Overview of identified perceived trustworthiness factors and risk-taking behaviour in Case A, B, C. The +/- sign refers to whether there are positive or negative traits of the respective factor or behaviour. A cell without a sign refers to no identified occurrence of the factor or behaviour. Only factors and behaviour that have occurred at least once in our material are included.

Perceived trustworthiness factors as antecedents to team trust					
Main category of perceived trustworthiness factors	*Subcategory*	*Case A* *Stakeholders:*	*Case B* *Stakeholders:*	*Case C* *Stakeholders:*	
		Development team and security architect	*Development team and information security expert*	*Development team and solutions architect*	*Development team and operations team*
Ability	Competence	+	+	-	
	Feedback Culture	+	-	+	-
Benevolence	Task Support	+		+	-
	Autonomy		+		-
Predictability	Availability		+		-
	Consistency		-		
Transparency	Information Transparency	+		+	-
	Responsibility Assignment		-		
	Openness			+	
Risk-taking behaviour as consequences of team trust					
Main category of risk-taking behaviour	*Subcategory*	*Case A* *Stakeholders:*	*Case B* *Stakeholders:*	*Case C* *Stakeholders:*	
		Development team and security architect	*Development team and information security expert*	*Development team and solutions architect*	*Development team and operations team*
Disclosure	Sharing confidential information				-
	Discussing mistakes & conflicts openly			+	-
Reliance	Asking for help	+	-	+	
	Forbearance from control	+	+	+	-

5.3.1 Case A

Findings from case A indicate that reliance is one of the risk-taking behaviours that are present between the security architect and the development team. Reliance revolves around the trustor making itself vulnerable by relying on those around them and includes giving the trustee responsibility and autonomy for work that is important to the trustor. Asking for help and forbearance from control are both sub-factors in the category of reliance, and we encounter both of these behaviours in case A. Asking for help includes making oneself vulnerable by asking for assistance or acknowledging one's mistakes. The security architect details how he has no issue asking the developers for their thoughts and insights regarding the implementation of security in the project. He is also open with the team about his shortcomings in terms of his abilities in technical security. The security architect asking the developers for their input and being open about his shortcomings are signs of the security architect trusting the team.

Forbearance from control is characterised as a risk-taking behaviour where the trustor displays trust in the trustee by not feeling the need to monitor them or their work. The security architect displays forbearance from control in his collaboration with the development team. Here, we consider the security architect as the trustor and the development team as the trustee. The security architect states that he wishes not to monitor the team and how they implement the necessary security measures. However, the security architect has made efforts to be of assistance to the development team. By arranging meetings between himself and the developers, he tries to assist whenever questions arise, sustaining strong task support. This coincides with what the senior developer claims to be the case. The senior developer has never experienced the security architect monitoring the development team's work. This behaviour indicates trust being placed in the development team by the security architect. Several potential perceived trustworthiness factors may contribute to the emergence of trust between the groups. His impression of the developer's ability seems to be amongst the most significant contributors. We have seen several indications of him commending their competence. Also, both the security architect and development team regard their feedback culture as good.

From the identified risk-taking behaviour, it can be concluded that there are clear signs of trust between the development team and the security architect. The risk-taking behaviours seemingly affect the work processes that ensure that information security is addressed in the team. Due to the security architect participating in the development team's activities, it seems as though he will be able to keep a focus on security throughout the project. Also, the results indicate a good feedback culture between the two groups. With the security architect making himself available to the development team for questions and security-related concerns, and with a solid feedback culture, it seems reasonable to assume that the team members will be able to air their concerns that may appear along the way. However, keeping the project's short duration thus far in mind, it seems too early to conclude how this level of trust has affected the actual security of the product. Overall, the trust relationship between the two groups seems beneficial to the security work in the project.

5.3.2 Case B

Findings from case B suggest different traits of the risk-taking behaviour reliance. There are identified negative behaviour of asking for help and positive behaviour of forbearance from control. As detailed in Section 4.2, the two security champions have described having some trouble related to asking the team for help. The security champions describe not knowing their team members' field of competence, making it difficult to know whom to ask regarding specific topics. One of them states the following:

> *"So we know this guy is well familiar with the system. Then it should probably be him who should do the task, but suddenly it turns out he has nothing to do with that at all. That was some other guy."*
>
> (Security Champion, ID 4)

They point to the new team composition contributing to this confusion. It appears that the fact that the team is new could be one of the reasons for the trustor to display less risk-taking behaviour. Newly composed teams often struggle from the lack of trust, as trust usually emerges over time with trust-building measures. A study that researched trust in GSE [JGŠ10] emphasises that the initial trust-building phase, where the trustor evaluates the trustworthiness of the trustee, affects the later phases of trust-building. If expectations for communication were not established initially, then there is a risk for the trust to be impaired in later phases due to unclearness in the initial phases. This might be a risk for the later trust development in case B. However, the project manager can mitigate this risk by engaging in trust-building activities.

There are positive traits of forbearance from control. The information security expert delegates tasks to the security champions, who pass them on to their respective teams. The security actor checks in with security champions on their progress on these tasks, but there are no signs of the security champions feeling monitored. The development team reported being free to implement security measures how they see fit, which indicates an aspect of forbearance from control by the security expert. Despite being responsible for several teams, the security does not monitor or excessively document the team's security efforts, which could also be seen as risk-taking behaviour. This also supports the trustworthiness factor autonomy that the security champions feel the team has. They explain that he often recommends them to do something rather than tell them what to do. However, it is beneficial for the security champions to receive additional support from the security expert in the early phases of implementing the roles, such as more frequent communication.

All in all, the results seem to indicate that the development team operates at a high degree of autonomy, completing its tasks without much intervention from the security expert. However, we have not identified significant risk-taking behaviour between the development team and security expert in case B. There are several possible reasons for this. The project has started somewhat recently, and it seems likely to have affected the observed risk-taking behaviours. Not having worked together long enough for the trust relationship to develop might be a causing factor. The interaction between the two groups being too infrequent, or the distance between them being too great to be analysed in a team trust context, might also contribute. The most prominent consequence of the lack of risk-taking behaviour has been the implementation of the security champions, which are characterised by uncertainty of responsibility and lack of on-boarding. Nevertheless, it does not seem to significantly affect the information security work in the team. The focus and resources the organisation devotes to information security seem to have more positive consequences for the information security work than the negative impact of the lack of observed risk-taking behaviour.

5.3.3 Case C

Our findings from case C show that the observed risk-taking behaviour differs depending on the security stakeholder. The senior developer talks about his different experiences working with the solutions architect and the operations team in the company to which he is contracted. These will therefore be evaluated separately.

Development team and solutions architect

There are positive traits of the risk-taking behaviour disclosure. This behaviour includes making oneself vulnerable by disclosing confidential information or being open to talking about own mistakes and weaknesses. Even though the solutions architect's role is to keep the developer's security efforts in check, both groups are active in discussing issues and problems. Both the solutions architect and the senior developer have reported instances where the two groups have let each other know they need to reevaluate their efforts.

Their cooperation also shows signs of reliance. The development team and solutions architect have known each other for a couple of years due to working on the same project for an extended time. The senior developer claims the developers have no issue asking questions or expressing concerns amongst themselves or the solutions architect. This indicates a degree of trust where the developers display risk-taking behaviour by asking for help and support. There are also signs of the solutions architect displaying forbearance from control. He states that he trusts the senior developer, whom he believes is very competent. At the same time, he points out that the developers are not security experts, which is an important factor in their cooperation. This indicates the security architect displaying trust in them within benevolence (the developer team is willing to do the work) but less trust regarding ability (the developer team shows less ability towards security work). There seems to be no indication of the solutions architect monitoring the developers. Despite the solutions architect having pointed to a general lack of security knowledge amongst the developers, he still seems to forbear control by delegating tasks and giving the developers freedom over how they complete their tasks.

We have identified clear positive signs of risk-taking behaviour between the development team and solutions architect. The team has worked together for an extended period and seems to have a foundation of trust among its members. This is indicated by the solutions architect forbearing control of security tasks and the team seeming to be able to discuss issues openly. One of the positive consequences of trust in case C is how they are able to communicate well when they need to be cautious about extracting data. However, we believe that the trustworthiness factors ability and competence are missing in the organisation to achieve better routines of working with information security.

Development team and operations team

Between the development team and the operations team, there are signs of negative traits of disclosure. The senior developer describes having had difficulty dealing with the customer's operations team. None of the participants in this study belongs to the operations team, so the following evaluation of risk-taking behaviours is based on the senior developer's narrative. The senior developer has shown some negative traits of the behaviour of sharing confidential information. As mentioned in Section 5.2, the senior developer detailed a loophole he refrains from informing the operations team about. He reflects on why this is the case:

> *"If the relationship had been better and they had a more pragmatic approach, someone would probably have informed them about [the loophole]."*
>
> (Senior Developer, ID 7)

The senior developer is cautious when sharing information with the operations team, especially since the information may lead to his work becoming more difficult.

Withholding work-related information and being cautious about what to share with others is regarded as negative risk-taking behaviour within the category of disclosure. The senior developer's (trustor) behaviour indicates a lack of trust towards the operations team (trustee).

There are also encountered negative traits of disclosure in the other direction. Here, the operations team would be acting as the trustor and senior developers as the trustee. The senior developer has expressed frustration about the operations team not informing them about crucial changes that affect their work:

> *"I feel they [operations team] are practising black-box security, where the less they inform, and the less people know, the better. Because the more information that's out there, the easier it is to get around certain [security measures]."* (Senior Developer, ID 7)

This has led to instances where developers were surprised to find their tools and software suddenly not working. The developer also feels the operations team controls their policies and computers too much, which can be seen as a negative trait of forbearance from control. The developer states that he wishes they were given more responsibility and freedom over their work, as the policies are overly rigid and strict. We see the consequences of this in their work with information security; it makes the work day cumbersome, and developers take shortcuts without telling anyone. Taking shortcuts and not following security policy due to a lack of forbearance from control could put them at more significant risk of security incidents and make incident management more difficult.

There might be several reasons why the operations team seems hesitant to inform the developers, which would constitute this negative risk-taking behaviour and point to a lack of trust towards the developers. It is, however, difficult to conclude without having had participants in the study from the operations team.

The results indicate mostly negative traits of risk-taking behaviour taking place between the developer team and the operations team. We argue that this has had a negative impact on how the work with information security is performed. We see an impact on efficiency where developers are inconvenienced by the operations team, a lack of open discussion of mistakes and conflicts, as well as the operations team forbearing control. Consequences include taking shortcuts and maintaining a somewhat negative attitude towards doing information security work.

5.3.4 Comparison of the cases

The contexts of cases A, B and C are somehow different, and this might have an impact on how relevant trust relationships are for the work of information security. The ratio between consultants and permanent employees of the hiring organisation might be one of these factors. For cases A and B, all participants are from the consultancy company. People from the same company might have a stronger trust relationship due to sharing some common values and experiences from the company [JGŠ10]. This corroborates our findings of a high level of trust in case A. However, this is not as strong of a factor for case B due to the recent creation of the team. There was one participant from the customer side and one from a consultancy company in case C. They had worked together for a few years, and the high level of trust that was found between them seemed to be influenced by shared experiences over several years.

In cases, A, B, and C, the teams and security stakeholders were structured differently, which affected the trust dimension ability the most. Case A had a security stakeholder who worked closely with the team. The security stakeholder in case B had six different teams he was responsible for, thus making the time spent with the teams more sporadic. In case C, they had one solutions architect that worked with the team. He lacked a formal and technical background in information security (of which he communicated openly to others), and thus security competence was a more scarce resource in that case.

The security resources, in the form of security experts and architects, seem to have the most impact on the trustworthiness category ability. Where there were competent support roles, there was higher reported ability among the team members.

The three different organisations in the cases belong to different sectors. However, we did not find any significant differences in how this affected either trust relationships or information security work. All the organisations claimed that information security was a priority as they all handled either sensitive data or managed critical infrastructure.

5.4 Recommendations to build trust

A summary of recommendations to build trust between the development team and the security stakeholder is presented in Table 5.2.

Table 5.2: Recommendations to build trust between the development team and different security stakeholders.

ID	Recommendation	Trust dimension
1	Common expectations for a communication standard	ability, transparency, predictability
2	Support a close cooperation between the information security stakeholder and the development team	benevolence, transparency, predictability
3	Raise information security awareness and competence	ability
4	Make use of security champions and security mediators in the organisation	ability, benevolence, transparency, integrity

5.4.1 Recommendation 1: Common expectations for a communication standard

Four participants mentioned improved communication as the most significant success factor for good cooperation between the development team and security stakeholders outside the team. It is essential to have a mutual understanding of how one should communicate questions, clarifications or security concerns. Agreeing on expectations for the communication structure before the need for it can reduce time spent on the issue, support quick feedback, and limit frustrations. Having dedicated communication channels with different topics are some of the solutions that the participants mention, and in their experience, they seem to work. Microsoft Teams channels, Slack channels, scrum boards or kanban boards are examples of measures the interviewees report as effective communication tools. Emails are discouraged as a communication medium because the information might get lost amongst all other emails. Having clear expectations for communication is also one of the recommended solutions to adapting DevSecOps in organisations [AHR12].

Communication and trust are closely related. Having a standard for communication understood by all actors involved can support both ability (through structured feedback culture), transparency (through information transparency and openness), and predictability, thus allowing trust to emerge between the team and the security stakeholders. Team communication has a positive effect on team trust [AKH+19], as well as is proven to stimulate both team creativity and support high-performing team efforts [BFG15].

This recommendation applies to security stakeholders who work closely with the development teams, such as security architects or mediators. It also applies to security teams that work alongside them, albeit not as closely, such as operations teams.

5.4.2 Recommendation 2: Support close cooperation between the information security stakeholder and the development team

Five participants mentioned measures that support a close work relationship between the team and the security actor as the most important factor for good cooperation. This includes knowing the team members and their respective areas of expertise and establishing shared goals in the team to support team coherence and autonomy. Leadership that focuses on cooperation within the team is also reported as an essential factor. This recommendation is closely connected with the agile development methodology and its features of collaboration found in DevOps and DevSecOps. Spending time on trust-building measures in the early phases of a project is beneficial for newly formed teams where few members know each other. The project manager has an important role, as it encompasses a facilitating function in building trust [NBV06; BIS07]. These measures include setting clear expectations for role assignment and task responsibility. We also see that when teams consist of members from both a consultancy company and the client organisation, it is important to create a coherent team. The typical separation between the security stakeholder and the development team should be reduced so that both parties feel there is less of a threshold to reach out when needed. The separation can be reduced by having the security stakeholder participate in update meetings, establishing communication standards or through security champions. These measures can support benevolence, transparency and predictability, and thus trust can emerge between the security stakeholder and the development team.

This recommendation applies to security stakeholders who work closely with the development team, such as security architects, mediators, and project managers.

5.4.3 Recommendation 3: Raise information security awareness and competence

Four of the participants mentioned raising security awareness, competence and attitude towards security as important factors in establishing good cooperation between the team and the security actors. Increasing security competence is one of the most stated recommendations in many studies investigating challenges and solutions in DevOps adoption within organisations [RZBS21; SC20]. Raising security awareness also affects the trust dimension of ability, as a trustor is more inclined to trust the trustee if they believe the trustee has knowledge in a certain domain. Based on the analysis of the interviewees, there seems to be a focus on raising security competence in the IT organisations, which supports the goal of emerging trust between the development team and the security stakeholders. Left-shifting security is one of the measures that can help raise security competence in the organisation. However, such organisational changes are complex, requiring time, resources and money. If the organisation becomes more aware of information security, this change can propagate to the development teams and security stakeholders. Our data material also suggests that raising security awareness gives other positive outcomes, such as a positive attitude towards doing information security work in the teams.

This recommendation applies to security stakeholders, project leaders, and the leadership of IT organisations.

5.4.4 Recommendation 4: Make use of security champions and security mediators in the organisation

Two participants mentioned that including a mediator role in the teams is a success factor for collaboration between the development team and the security actor. The role should be someone who can convey and clarify information and help cooperation between an organisation's development team and security team. This can be realised in teams through security champions or between teams by using a mediator. Two of the cases we have studied already use security champions, and they encountered challenges regarding implementing the role in both cases. Therefore, it is essential to devote both resources and time to training the security champions. The role requires someone with a specific skill set, and the lack of which will lead to the role not fulfilling its potential. The usage of security champions is also a recommended measure in BSIMM [BSIMMb] to foster collaboration. The role has been argued to bridge the gap between IT security (which responsibility often resides further up in the organisation) and software security (which responsibility often resides in the development teams) [TJC20]. The IT security specialist in case D works currently as an information security mediator between development teams and a cyber security department in his organisation. He tries to provide value to the development teams and the security department in the organisation. Including a mediating role contributes to benevolence (through task support), ability (through feedback culture), transparency (through disseminating information) and integrity (through mediating interest conflicts). Thus, it can assist in the emergence of trust between the development team and the security stakeholders.

5.5 Limitations

This section discusses the limitations that might have been present throughout the stages of the study.

5.5.1 The overall fit between the study and the framework

We have used the framework of Breuer et al. [BHHH20] as a guideline to investigate trust relations, and it has both advantages and disadvantages. The framework provides a taxonomy of perceived trustworthiness factors as antecedents of team trust and risk-taking behaviours as consequences of team trust. This is advantageous when mapping out behaviours we see and how they relate to the complex concept of trust. However, this taxonomy has been developed for intra-team trust, which differs slightly from our context.

Thus, some of the factors in the framework might suit this context. An example of this might be physical and "mental" distance between the security stakeholder and the development team. Through the interviews, this has shown to be of interest. A prerequisite of the framework is that the team already works within a short distance and thus does not address distance. We have also seen that the participants more frequently mention some trustworthiness factors as more important than others. This reflects in the answers to the research questions concerning challenges, consequences and recommendations in Chapter 5. Ability, transparency, predictability, and benevolence are factors that seem to be more important than integrity. Overall the taxonomy provided by Breuer et al. seems like a good fit but may have lacked some constructs that could have been relevant to explore in our context.

We have also experienced that the perceived trustworthiness factors within the five main categories may sometimes overlap. This has introduced challenges in our data analysis and how we map certain behaviours to certain factors. We have followed the definitions in Brauer et al.'s taxonomy to avoid ambiguity instead of other sources. An example of this challenge is the two factors *keeping commitments* and *integrity*. Some authors [CS09; MDS95] argue that *integrity* actually comprise the factor of *keeping commitments*. According to these sources, integrity both comprises to which extent the trustor finds the trustee's ethical values acceptable and how congruent the trustee's words and actions are. However, *keeping commitments* are categorised by [BHHH20] as a sub factor within *predictability*. This can create challenges in the analysis phase and show that the factors are closely related.

5.5.2 Social desirability bias

Social desirability bias may be present in the study. This refers to when participants *"give socially desirable responses instead of choosing responses that are reflective of* *their true feelings"* [Gri10]. A common trait of all the participants is elaborating on contexts in work environments. In this way, they might feel like they are not only representing themselves but also the company they are employed at. In addition, some also represent the firm to which they are contracted. This might affect their ability or willingness to speak freely about their work conditions and relations with other coworkers. Most of the interviewees from cases A, B and C were also aware of the participation of the other interviewees in the same case. This could also make them more reluctant to share critical thoughts and reflections.

5.5.3 Triangulation

Triangulation refers to using multiple data collection methods or several data sources to increase the credibility of a study [Sal10]. The research questions are studied from two perspectives, the team perspective and the perspective of the information security stakeholder. This allows us to compare the trust relationship findings from both perspectives. Relevant literature from Chapter 2 is also used as a data source, improving credibility further. Case A, B and parts of C have participants from both the development side and the security stakeholder's side. Case D contain individuals from different case contexts, with none being from the same case. This impacts the triangulation. The material they represent is their own experience, and we have not introduced the perspective of the other side of the conflicts that are presented. However, the stories told in this case are still relevant to understanding trust relationships and how they affect information security work from their perspectives.

Another limitation is the lack of documentation supporting what the participants communicate. Examples of this can be organisation policies that include security requirements that could be compared to the security requirements the participants say they follow. Another data source could be security evaluations from penetration testers to see how secure a solution is. However, this would require a study with a longer duration to capture how a trust relationship over a more extended period might affect the security of the final solution.

5.5.4 Recruitment

All of the participants did not fulfil the requirements we set for the participants in the study. We wanted to recruit senior developers, as their experience might bring valuable insights into the topic area. We also wanted them to have experience with a project where security requirements were an essential factor in the project. However, due to the limited reach of this study and busy potential participants, we included two junior developers (case B) and some senior developers who have not worked on projects with any security requirements.

5.5.5 Evaluation of validity and reliability

Some limitations regarding the study's reliability and validity are relevant to consider when assessing the study design. The following section evaluates these two aspects of semi-structured interviews and case studies and is included from the specialisations project [FT21]. Although semi-structured interviews with the war story technique appear to be an appropriate methodology, the method has some weaknesses and limitations. Runeson and Höst propose guidelines [RH09] that address different types of validity and reliability when conducting case studies or interviews in software engineering. First and foremost, validity denotes the study's trustworthiness and should be considered from the beginning of the study and throughout all research stages. The following types of validity can threaten the overall validity, and our study has taken countermeasures.

Construct validity refers to if the researched topic in the researcher's mind reflects in the research questions and interviews. If the interview objects interpret the questions differently from the researchers, construct validity is a threat. To increase the validity of the findings from the interviews, the transcripts or parts of them can be sent back to the participant to verify outcomes [MB18]. Case descriptions presented in 3.2 have been sent back to the participants for verification and approval.

Internal validity refers to the validity of causal relations. There might be unknown factors affecting another factor that the researcher is unaware of, which may cause threats to internal validity. There might be a valid threat to the internal validity of this study. We have tried to manifest whether there is a "cause and effect" scenario with trust relationships and how information security work is performed in the teams. Our findings suggest that there are effects related to the work process of the information security work, with the cause being either high-trust relationships or low-trust relationships in the teams. However, many other factors can influence how information security work is performed; tools, organisational structure, education [RZBS21] to mention a few.

External validity concerns the extent to which the results can be generalised and hence relevant for other research groups, defining a theory or other cases. In our study, the findings and discussion are applied to the context we have researched. We have investigated trust relationships between security stakeholders and development teams in different contexts in organisations and provided recommendations for these groups specifically. The purpose is that these findings can be relevant to case contexts with similar characteristics as the investigated cases. The reality of trust relationships might be different in other organisational contexts and are out of this research scope.

Reliability is the final aspect and refers to how the conducted study is dependent on the researcher(s). Ideally, the study should have the same results independently of the researcher(s), but interviews are prone to bias and errors and can cause a threat to reliability. This might occur at different stages in the process, like preparation of questions, conducting the interview or in the stage of interpretation analysis [HL01]. Bias and errors come from the personal nature of the interview that allows for more in-depth exploration based on spontaneous follow-up questions that are not prepared in advance. To reduce bias and increase validity [RH09], two researchers are conducting the study. We have cross-checked coding in several iterations for a clear understanding of the coding process. Parts of categorisation and coding have been done together as well.

Chapter 6 Conclusion and future work

We have investigated how trust relationships between software development teams and security stakeholders affect the information security work in the teams. Challenges related to perceived trustworthiness factors and their consequences on information security work have been identified between the two parties. The topic has been investigated within the greater perspective of making agile development more focused on information security and, ultimately, contributing toward creating more secure and robust IT solutions.

The challenges we have identified related to trustworthiness factors are mostly found when analysing the relationship between the development team acting as the trustor and the security stakeholder acting as the trustee. Challenges were identified in regards to ability, transparency, predictability and benevolence. Analysing the relationship in the other direction, where the security stakeholder is the trustor and the development team the trustee, we find ability as the only trustworthiness factor of which we identified challenges. We consider ability as the most important factor affecting information security work. Integrity is found to be the least important factor in its effect on how information security work is performed. The consequences encountered relate to the process of working with information security in the team. They include developers taking shortcuts and trying to avoid cumbersome security policies. There are also uncertainties about where to report security issues and a lack of incentive to report security concerns. Time-consuming work and lost work time are also consequences and considerable contributors to the development team's frustrations. Lastly, we see a lack of motivation to engage in information security work due to the identified challenges.

We have formulated four recommendations for building trust between the development team and the security stakeholder based on existing literature, suggestions from the participants, and the analysation of the interviews. These include 1) establishing common expectations for communication between the development teams and the security stakeholders, 2) supporting close cooperation between the responsible security actor and the development team, 3) raising information security awareness and competence in the development team and 4) making use of security champions or security mediators in the organisation to facilitate cooperation between security departments and development teams. These four recommendations aim to improve the five trust dimensions. We recommend implementing these in the early phases of a project.

We have investigated three different cases in the IT industry and the fourth case with supplementary material. Case A showed positive traits of risk-taking behaviour, which corroborate the high degree of trust we encountered among the participants. The high degree of trust seemingly had a positive effect on the process of working with information security. The participants in case B showed fewer traits of risk-taking behaviour, which seemingly came as a result of the project having started relatively recently. However, it did not impact the overall information security work in the team. In case C, there seemed to be two security stakeholders of interest; the solutions architect and the IT operations team. There seemed to be apparent risk-taking behaviour between the architect and the development team, which positively influenced the process of working with information security. There was a lack of risk-taking behaviour between the development team and the operations team, which negatively impacted the work process. The findings from this study indicate that trust relationships impact the process of working with information security in the contexts we have investigated.

The contributions of this master thesis include mapping out trust relationships in a new context to investigate how it affects specific work in the information security field. The recommendations for building trust aim to foster a cooperative relationship between a software development team and a security stakeholder to improve the work processes with information security. With a focus on trust relations, we contribute to a more comprehensive picture of how interdisciplinary cooperation emerges.

Recommendations for future work

There is a need for more research on trust relationships to solve the challenges related to integrating information security work in agile development. There is a need to validate our findings throughout a long-term project period. We have investigated three cases and collected experiences and thoughts from a fourth case. We had one encounter with each participant, which only provides a snapshot of the situation and context at the time of the interview. Trust generally evolve over time, and it would be interesting to research a development team and the security stakeholders over an entire project. Such a project could facilitate the testing and validation of our recommendations for building trust. It could also make it possible to objectively evaluate the overall security of the final product through either penetration testing or other security testing approaches. Evaluating the final product after a long-term project could help answer whether or not trust relationships affect the solution's security.

We also see the possibility of investigating trust relationships in other team structures and compositions. Cases A, B and C all had information security mediators, which in the study seems beneficial for communication and trust. Based on the findings from case D, the industry does not seem to have adopted the mediator role to a large extent, which points to the possibility of investigating trust relationships directly between development teams and security teams.

Another interesting aspect that can be explored further is the trust relationship between an organisation and the consultancy company it contracts for a project. Several participants mentioned economic factors, reputation and ownership of responsibility for potential security incidents as issues they have experienced between the two parties. Investigating trust relationships in this context could contribute to solving some of these cooperation challenges.

References

[AC19] S. Abraham and I. Chengalur-Smith, «Evaluating the effectiveness of learner controlled information security training», *Computers & Security*, vol. 87, p. 101 586, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818308873.

[Agile manifesto] *Manifesto for agile software development.* [Online]. Available: https://agilemanifesto.org (last visited: May 28, 2022).

[AHR12] A. Ahmad, J. Hadgkiss, and A. Ruighaver, «Incident response teams – challenges in supporting the organisational security function», *Computers & Security*, vol. 31, no. 5, pp. 643–652, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404812000624.

[AKH+19] S. Akhtar, K. U. Khan, *et al.*, «Antecedents of task performance: An examination of transformation leadership, team communication, team creativity, and team trust», *J. Public Affairs*, vol. 19, no. 2, e1927, May 2019.

[ASMA22] M. A. Akbar, K. Smolander, *et al.*, «Toward successful devsecops in software development organizations: A decision-making framework», *Information and Software Technology*, vol. 147, p. 106 894, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584922000568.

[AT03] S. Albrecht and A. Travaglione, «Trust in public-sector senior management», *International Journal of Human Resource Management*, vol. 14, Feb. 2003.

[Atl21a] Atlassian, *DevSecOps: Injecting Security into CD Pipelines | Atlassian*, Oct. 2021. [Online]. Available: https://www.atlassian.com/continuous-delivery/principles/devsecops (last visited: Oct. 19, 2021).

[Atl21b] ——, *What is devops?*, Oct. 2021. [Online]. Available: https://www.atlassian.com/devops (last visited: Oct. 19, 2021).

[Bad21] R. Badhwar, *The CISO's Transformation.* Cham, Switzerland: Springer, 2021.

[BFG15] K. Boies, J. Fiset, and H. Gill, «Communication and trust are key: Unlocking the relationship between leadership and team performance and creativity», *The Leadership Quarterly*, vol. 26, no. 6, pp. 1080–1094, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1048984315000934.

[BHH16] C. Breuer, J. Hüffmeier, and G. Hertel, «Does trust matter more in virtual teams? A meta-analysis of trust and team effectiveness considering virtuality and documentation as moderators», *J. Appl. Psychol.*, vol. 101, no. 8, pp. 1151–1177, Aug. 2016.

[BHHH20] C. Breuer, J. Hüffmeier, *et al.*, «Trust in teams: A taxonomy of perceived trustworthiness factors and risk-taking behaviors in face- to-face and virtual teams», *Human relations*, vol. 1, no. 1, pp. 3–34, 2020. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0018726718818721.

[BIS07] E. Berki, H. Isomäki, and A. Salminen, «Quality and trust relationships in software development», *Software quality in the knowledge society-Software quality management XV*, pp. 47–65, 2007.

[BMC Blogs] *Types of IT Teams.* [Online]. Available: https://www.bmc.com/blogs/it-teams (last visited: May 31, 2022).

[BSIMMa] Building Security In Maturity Model or BSIMM from Synopsys. [Online]. Available: https://www.bsimm.com/about.html (last visited: Oct. 19, 2021).

[BSIMMb] Software Security Metrics and Strategy | BSIMM. [Online]. Available: https://www.bsimm.com/framework/governance/software-security-metrics-strategy.html (last visited: Oct. 19, 2021).

[BSIMMc] What Is the BSIMM and How Does It Work? | Synopsys. [Online]. Available: https://www.synopsys.com/glossary/what-is-bsimm.html (last visited: Oct. 19, 2021).

[CH01] A. Cockburn and J. Highsmith, «Agile software development, the people factor», *Computer*, vol. 34, no. 11, pp. 131–133, 2001.

[CLH+] M. Casagni, M. Lead, *et al.*, «Federal devops summit»,

[Cos03] A. C. Costa, «Work team trust and effectiveness», *Personnel Review*, vol. 32, no. 5, pp. 605–622, Oct. 2003.

[CP12] Y. J. Cho and T. Poister, «Human resource management practices and trust in public organizations», *Public Management Review - PUBLIC MANAG REV*, vol. 15, pp. 1–23, Jan. 2012.

[CRT01] A. C. Costa, R. A. Roe, and T. Taillieu, «Trust within teams: The relation with performance effectiveness», *European Journal of Work and Organizational Psychology*, vol. 10, no. 3, pp. 225–244, 2001. [Online]. Available: https://doi.org/10.1080/13594320143000654.

[CS08] J. Corbin and A. Strauss, *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory.* Thousand Oaks: SAGE Publications, Inc., 2008.

[CS09] J. A. Colquitt and S. C. Salam, «Foster trust through ability, benevolence, and integrity», *Handbook of principles of organizational behavior: Indispensable knowledge for evidence-based management*, pp. 389–404, 2009.

[CSD] H. G. Corneliussen, G. Seddighi, and C. A. Dralega, «18. women&#x2019;s experience of role models in it: Landmark women, substitutes and supporters», in *Modeller*, pp. 375–395. [Online]. Available: https://www.idunn.no/doi/abs/10.18261/9788215034393-2019-18.

[CSL07] J. A. Colquitt, B. A. Scott, and J. A. LePine, «Trust, trustworthiness, and trust propensity: A meta-analytic test of their unique relationships with risk taking and job performance», *Journal of Applied Psychology*, vol. 92, no. 4, pp. 909–927, 2007. [Online]. Available: https://doi.org/10.1037/0021-9010.92.4.909.

[Cyber Degrees] *Penetration tester career overview.* [Online]. Available: https://www.cyberdegrees.org/jobs/penetration-tester (last visited: Jun. 10, 2022).

[CYL13] S. O. Cheung, T. W. Yiu, and M. C. Lam, «Interweaving trust and communication with project performance», *Journal of Construction Engineering and Management*, vol. 139, no. 8, pp. 941–950, 2013.

[DF02] K. T. Dirks and D. L. Ferrin, «Trust in leadership: Meta-analytic findings and implications for research and practice», *Journal of Applied Psychology*, vol. 87, no. 4, pp. 611–628, 2002. [Online]. Available: https://doi.org/10.1037/0021-9010.87.4.611.

[DMC19] E. Diel, S. Marczak, and D. S. Cruzes, *Communication challenges and strategies in distributed devops*, Feb. 2019. [Online]. Available: https://repositorio.pucrs.br/dspace/bitstream/10923/14121/2/Communication_Challenges_and_Strategies_in_Distributed_DevOps.pdf (last visited: Oct. 25, 2021).

[Edm11] A. Edmondson, «Psychological safety, trust, and learning in organizations: A group-level lens», *Trust and Distrust in Organizations: Dilemmas and Approaches*, Oct. 2011.

[Edm18] A. C. Edmondson, *The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth*, eng. Newark: John Wiley & Sons, Incorporated, 2018.

[Edm99] A. Edmondson, «Psychological safety and learning behavior in work teams», *Administrative Science Quarterly*, vol. 44, no. 2, pp. 350–383, 1999. [Online]. Available: https://doi.org/10.2307/2666999.

[EGHS16] C. Ebert, G. Gallardo, *et al.*, «Devops», *Ieee Software*, vol. 33, no. 3, pp. 94–100, 2016.

[FG12] A. C. Fulmer and M. J. Gelfand, «At what level (and in whom) we trust: Trust across multiple organizational levels», *Journal of Management*, vol. 38, no. 4, pp. 1167–1230, 2012. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0149206312439327.

[Fla55] J. C. Flanagan, «The critical incident technique.», *Psychol. Bull.*, vol. 51, no. 4, p. 327, 1955.

[FT21] K. Formo Buene and H. Tenold Fridtun, «Investigating trust relations in devsecops», Department of Information Security, Communication Technology, NTNU – Norwegian University of Science, and Technology, Project report in TTM4502, Dec. 2021.

[Full Scale] *What Is A Senior Developer: Qualities To Look For | Full Scale.* [Online]. Available: https://fullscale.io/blog/senior-developer (last visited: May 31, 2022).

[Gri10] P. Grimm, «Social Desirability Bias», in *Wiley International Encyclopedia of Marketing*, Chichester, England, UK: John Wiley & Sons, Ltd, Sep. 2010.

[GWRC20] L. M. Gray, G. Wong-Wylie, *et al.*, «Expanding Qualitative Research Interviewing Strategies: Zoom Video Communications», *NSUWorks*, vol. 25, no. 5, pp. 1292–1301, 2020.

[HCJW20] A. Hartwig, S. Clarke, *et al.*, «Workplace team resilience: A systematic review and conceptual development», *Organizational Psychology Review*, vol. 10, no. 3-4, pp. 169–200, Apr. 2020.

[HL01] S. Harvey-Jordan and S. Long, «The process and the pitfalls of semi-structured interviews: The journal of the health visitors' association», English, *Community Practitioner*, vol. 74, no. 6, p. 219, Jun. 2001, Copyright - Copyright TG Scott & Son Ltd. Jun 2001; Last updated - 2016-04-30. [Online]. Available: https://www.proquest.com/scholarly-journals/process-pitfalls-semi-structured-interviews/docview/213313284/se-2?accountid=12870.

[HR21] P. Hennel and C. Rosenkranz, «Investigating the "socio" in socio-technical development: The case for psychological safety in agile information systems development», *Project Management Journal*, vol. 52, no. 1, pp. 11–30, 2021. [Online]. Available: https://doi.org/10.1177/8756972820933057.

[IKT-Norge18] *Andelen kvinner i norsk it-bransje*, 2018. [Online]. Available: https://www.ikt-norge.no/wp-content/uploads/2019/10/kvinner-i-itbransjenoda2018-kantartns.pdf (last visited: Mar. 11, 2022).

[JGŠ10] S. Jalali, C. Gencel, and D. Šmite, «Trust dynamics in global software engineering», in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '10, Bolzano-Bozen, Italy: Association for Computing Machinery, 2010. [Online]. Available: https://doi.org/10.1145/1852786.1852817.

[JGST20] D. Jampen, G. Gür, *et al.*, «Don't click: towards an effective anti-phishing training. A comparative literature review», *Hum. Cent. Comput. Inf. Sci.*, vol. 10, no. 1, pp. 1–41, Dec. 2020.

[JR00] F. L. Jeffries and R. Reed, «Trust and adaptation in relational contracting», *The Academy of Management Review*, vol. 25, no. 4, pp. 873–882, 2000. [Online]. Available: http://www.jstor.org/stable/259212.

[Kenzie Academy] *Front End vs. Back End: What's the Difference? - Kenzie Academy.* [Online]. Available: https://kenzie.snhu.edu/blog/front-end-vs-back-end-whats-the-difference (last visited: May 31, 2022).

[KR19] B. Krebs, *Krebs on securty: Should failing phish tests be a fireable offense?* [Online]. Available: https://krebsonsecurity.com/2019/05/should-failing-phish-tests-be-a-fireable-offense/ (last visited: May 20, 2021).

[LBL11] C. A. Lengnick-Hall, T. E. Beck, and M. L. Lengnick-Hall, «Developing a capacity for organizational resilience through strategic human resource management», *Human Resource Management Review*, vol. 21, no. 3, pp. 243–255, 2011, International Human Resource Management: Theoretical and Strategic Advances. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053482210000355.

[LMS11] F. Lyon, G. Mollering, and M. N. K. Saunders, eng. Edward Elgar Publishing Limited, 2011.

[LS07] W. G. Lutters and C. B. Seaman, «Revealing actual documentation usage in software maintenance through war stories», *Information and Software Technology*, vol. 49, no. 6, pp. 576–587, 2007, Qualitative Software Engineering Research. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584907000158.

[MB18] D. Magaldi and M. Berler, «Semi-structured interviews», in *Encyclopedia of Personality and Individual Differences.* Springer International Publishing, 2018, pp. 1–60. [Online]. Available: https://doi.org/10.1007/978-3-319-28099-8_857-1.

[MC01] D. H. McKnight and N. L. Chervany, *Trust and Distrust Definitions: One Bite at a Time.* Jan. 2001, vol. 2246.

[MC17] H. Myrbakken and R. Colomo-Palacios, «Devsecops: A multivocal literature review», in *Software Process Improvement and Capability Determination*, A. Mas, A. Mesquida, *et al.*, Eds., Cham: Springer International Publishing, 2017, pp. 17–29.

[McG04] G. McGraw, «Software security», *IEEE Security & Privacy*, vol. 2, no. 2, pp. 80–83, 2004.

[McG18] G. McGraw, «Silver Bullet Talks with Tanya Janca», *IEEE Secur. Priv.*, vol. 16, no. 5, pp. 7–11, Oct. 2018.

[MDS95] R. C. Mayer, J. H. Davis, and F. D. Schoorman, «An integrative model of organizational trust», *The Academy of Management Review*, vol. 20, no. 3, pp. 709–734, 1995. [Online]. Available: http://www.jstor.org/stable/258792.

[Monday] *4 types of Agile ceremonies and how to manage them.* [Online]. Available: https://monday.com/blog/rnd/agile-ceremonies (last visited: May 31, 2022).

[MŠ07] N. B. Moe and D. Šmite, «Understanding Lacking Trust in Global Software Teams: A Multi-case Study», in *Product-Focused Software Process Improvement*, Berlin, Germany: Springer, 2007, pp. 20–34.

[MT11] B. McEvily and M. Tortoriello, «Measuring trust in organisational research: Review and recommendations», *Journal of Trust Research*, vol. 1, no. 1, pp. 23–63, 2011. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/21515581.2011.552424.

[NBV06] P. T. Nguyen, M. A. Babar, and J. M. Verner, «Critical factors in establishing and maintaining trust in software outsourcing relationships», in *Proceedings of the 28th International Conference on Software Engineering*, ser. ICSE '06, Shanghai, China: Association for Computing Machinery, 2006, pp. 624–627. [Online]. Available: https://doi.org/10.1145/1134285.1134377.

[NIST01] N. I. of Standards and Technology, *Information security*, 2001. [Online]. Available: https://csrc.nist.gov/glossary/term/information_security (last visited: Jun. 1, 2022).

[NN19] N. S. (NSM) and N. C. (NCSC), «Helhetlig digitalt risikobilde 2019», Nasjonal Sikkerhetsmyndighet, Rødskiferveien 20, Kolsås, Tech. Rep. DOE-SLC-6903-1, 2019.

[PDF+03] L. M. Prati, C. Douglas, *et al.*, «Emotional intelligence, leadership effectiveness, and team outcomes», *International Journal of Organizational Analysis*, vol. 11, pp. 21–40, Dec. 2003.

[Pup21] Puppet, «State of DevOps Report 2021 | Puppet», *Puppet Report*, Nov. 2021.

[RH09] P. Runeson and M. Höst, «Guidelines for conducting and reporting case study research in software engineering», *Empir. Software Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009.

[RSBC98] D. M. Rousseau, S. B. Sitkin, *et al.*, «Not So Different After All: A Cross-discipline View of Trust», *Acad. Manage. Rev.*, vol. 23, no. 3, Jul. 1998.

[RZBS21] R. N. Rajapakse, M. Zahedi, *et al.*, «Challenges and solutions when adopting DevSecOps: A systematic review», *arXiv*, Mar. 2021. [Online]. Available: https://arxiv.org/abs/2103.08266v2.

[Sal10] N. J. Salkind, «Triangulation», *Encyclopedia of research design*, vol. 1-0, 2010. [Online]. Available: https://methods.sagepub.com/reference/encyc-of-research-design/n469.xml.

[SC20] M. Sánchez-Gordón and R. Colomo-Palacios, «Security as culture: A systematic literature review of devsecops», in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 266–269. [Online]. Available: https://doi.org/10.1145/3387940.3392233.

[Sch06] C. D. Schultz, «A trust framework model for situational contexts», in *PST '06: Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, New York, NY, USA: Association for Computing Machinery, Oct. 2006, pp. 1–7.

[Sch11] J. Schaubroeck, «Cognition-based and affect-based trust as mediators of leader behavior influences on team performance.», *J. Appl. Psychol.*, vol. 96, no. 4, p. 863, 2011.

[SH21] G. Spafford and J. Herschmann, «Hype Cycle for Agile and DevOps, 2020», *Gartner*, Nov. 2021.

[SKF21] S. Sharma, D. Kumar, and M. Fayad, «An impact assessment of agile ceremonies on sprint velocity under agile software development», in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2021, pp. 1–5.

[SMD07] F. D. Schoorman, R. C. Mayer, and J. H. Davis, «An integrative model of organizational trust: Past, present, and future», *Academy of Management Review*, vol. 32, no. 2, pp. 344–354, 2007. [Online]. Available: https://doi.org/10.5465/amr.2007.24348410.

[SMH18] V. Stray, N. B. Moe, and R. Hoda, «Autonomous agile teams: challenges and future directions for research», in *XP '18: Proceedings of the 19th International Conference on Agile Software Development: Companion*, New York, NY, USA: Association for Computing Machinery, May 2018, pp. 1–5.

[SMSJ04] J. Stanton, P. Mastrangelo, *et al.*, «Behavioral information security: Two end user survey studies of motivation and security practices.», *Proceedings of the 10th Americas Conference on Information Systems*, p. 175, Jan. 2004.

[SvYVR21] R. Smit, J. van Yperen Hagedoorn, *et al.*, «The Soft Skills Business Demands of the Chief Information Security Officer», *CSUSB ScholarWorks*, vol. 30, no. 4, pp. 41–59, 2021. [Online]. Available: https://scholarworks.lib.csusb.edu/jitim/vol30/iss4/3.

[TJC20] I. A. Tondel, M. G. Jaatun, and D. S. Cruzes, «IT Security Is From Mars, Software Security Is From Venus», *IEEE Secur. Priv.*, vol. 18, no. 4, pp. 48–54, Jul. 2020.

[TLH19] N. Tomas, J. Li, and H. Huang, «An empirical study on culture, automation, measurement, and sharing of devsecops», in *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 2019, pp. 1–8.

[TSS22] S. Tyagi, R. Sibal, and B. Suri, «Empirically developed framework for building trust in distributed agile teams», *Information and Software Technology*, p. 106 828, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584922000064.

[vYSVR21] J. M. J. van Yperen Hagedoorn, R. Smit, *et al.*, *Soft Skills of The Chief Information Security Officer*, 2021. [Online]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1015&context=bled2021 (last visited: Jan. 31, 2021).

Appendix A
Consent form and prequestionnaire for selection 1 and 2

Sjekk universell utforming i skjemaet

Samtykkeskjema - Studie om sikkerhetsarbeid i utviklingsteam utvalg 1

Obligatoriske felter er merket med stjerne *

Prosjektinformasjon og formål

Dette er et samtykkeskjema for deltakelse i en masteroppgave som handler om hvordan informasjonssikkerhet i utviklingsteam kan ivaretas. Prosjektet vil se på samarbeidet mellom utviklingsteam og aktører utenfor teamet som setter premisser til sikkerhet, samt hvordan denne relasjonen påvirker arbeidet med sikkerhet i utviklingsprosjekter.

I dette skrivet får du informasjon om bakgrunn og formål for prosjektet, samt hva deltakelse i prosjektet innebærer.

Målgruppen er utviklere som jobber teambasert på prosjektarbeid.

Hvem er ansvarlig for prosjektet?

NTNU er ansvarlig for prosjektet.

Knut Formo Buene (knutfb@stud.ntnu.no) og Helga Tenold Fridtun(helgatf@stud.ntnu.no) utfører arbeidet ved masteroppgaven.

Ansvarlig hovedveileder er Maria Bartnes ved NTNU og Sintef, maria.bartnes@sintef.no.

Medveileder er Roy Myhre ved Sopra Steria, roy.myhre@soprasteria.com.

Hva innebærer det for deg å delta?

Deltakelse i prosjektet innebærer å delta i et intervju der det blir tatt lydopptak ved fysisk møte, eller videoopptak ved digitalt intervju. Vi ber deg også om å fylle ut et par spørsmål om din arbeidsbakgrunn litt lengre ned i dette skjemaet.

Vi kommer ikke til å bruke navnet ditt eller bedriftsnavnet i masteroppgaven, og det vil heller ikke være mulig å identifisere deg ut fra informasjonen vi inkluderer i masteroppgaven.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern - hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Alle lyd- og videoopptak slettes etter at forskningsprosjektet er avsluttet i juni 2022.
All annen data vil være fullt anonymisert.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra NTNU har Norsk senter for forskningsdata (NSD) vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket (prosjektID: 384152).

Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

NTNU ved

- Knut Formo buene, knutfb@stud.ntnu.no
- Helga Tenold Fridtun, helgatf@stud.ntnu.no
- Maria Bartnes, maria.bartnes@sintef.no

Vårt personvernombud v/Thomas Helgesen, thomas.helgesen@ntnu.no

Jeg har mottatt og forstått informasjon om deltagelse i masteroppgaven om sikkerhetsarbeid i utviklingsteam, og har fått anledning til å stille spørsmål. Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet *

Jeg samtykker til:

○ å delta på et fysisk intervju med lydopptak samt å svare å på noen spørsmål om min arbeidsbakgrunn i dette skjemaet.

○ å delta på et digitalt intervju via Teams med videoopptak samt å svare på noen spørsmål om min arbeidsbakgrunn i dette skjemaet.

Hva heter du (fornavn og etternavn)? *

Hvilket kjønn identifiserer du deg med? *

○ Mann

○ Kvinne

○ Ikke-binær

○ Ønsker ikke å oppgi

Hva slags studiebakgrunn har du? *

Hvor lang arbeidserfaring har du? *

Indiker antall år.

Hvilken bedrift jobber du i? *

Inkluder gjerne hvilken bedrift du er ansatt i og hvilken bedrift du eventuelt er utleid til.

Hva er din rolle i teamet? *

Hvor mange består teamet ditt av? *

Se nylige endringer i Nettskj

Sjekk universell utforming i skjemaet

Samtykkeskjema - Studie om sikkerhetsarbeid i utviklingsteam utvalg 2

Obligatoriske felter er merket med stjerne *

Prosjektinformasjon og formål

Dette er et samtykkeskjema for deltakelse i en masteroppgave som handler om hvordan informasjonssikkerhet i utviklingsteam kan ivaretas. Prosjektet vil se på samarbeidet mellom utviklingsteam og aktører utenfor teamet som setter premisser til sikkerhet, samt hvordan denne relasjonen påvirker arbeidet med sikkerhet i utviklingsprosjekter.

I dette skrivet får du informasjon om bakgrunn og formål for prosjektet, samt hva deltakelse i prosjektet innebærer.

Målgruppen er personer som har en kontakt med teamet og kan stille funksjonelle krav og/eller legge føringer for sikkerhetsarbeidet i utviklingsprosessen i teamet.

Hvem er ansvarlig for prosjektet?

NTNU er ansvarlig for prosjektet.

Knut Formo Buene (knutfb@stud.ntnu.no) og Helga Tenold Fridtun(helgatf@stud.ntnu.no) utfører arbeidet ved masteroppgaven.

Ansvarlig hovedveileder er Maria Bartnes ved NTNU og Sintef, maria.bartnes@sintef.no.

Medveileder er Roy Myhre ved Sopra Steria, roy.myhre@soprasteria.com.

Hva innebærer det for deg å delta?

Deltakelse i prosjektet innebærer å delta i et intervju der det blir tatt lydopptak ved fysisk møte, eller videoopptak ved digitalt intervju. Vi ber deg også om å fylle ut et par spørsmål om din arbeidsbakgrunn litt lengre ned i dette skjemaet.

Vi kommer ikke til å bruke navnet ditt eller bedriftsnavnet i masteroppgaven, og det vil heller ikke være mulig å identifisere deg ut fra informasjonen vi inkluderer i masteroppgaven.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern - hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Alle lyd- og videoopptak slettes etter at forskningsprosjektet er avsluttet i juni 2022. All annen data vil være fullt anonymisert.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra NTNU har Norsk senter for forskningsdata (NSD) vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket (prosjektID: 384152).

Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

NTNU ved

- Knut Formo Buene, knutfb@stud.ntnu.no
- Helga Tenold Fridtun, helgatf@stud.ntnu.no
- Maria Bartnes, maria.bartnes@sintef.no

Vårt personvernombud v/Thomas Helgesen, thomas.helgesen@ntnu.no

Jeg har mottatt og forstått informasjon om deltagelse i masteroppgaven om sikkerhetsarbeid i utviklingsteam, og har fått anledning til å stille spørsmål. Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet *

Jeg samtykker til:

○ å delta på et fysisk intervju med lydopptak samt å svare å på noen spørsmål om min arbeidsbakgrunn i dette skjemaet.

○ å delta på et digitalt intervju via Teams med videoopptak samt å svare på noen spørsmål om min arbeidsbakgrunn i dette skjemaet.

Hva heter du (fornavn og etternavn)? *

Hvilket kjønn identifiserer du deg med? *

○ Mann

○ Kvinne

○ Ikke-binær

○ Ønsker ikke å oppgi

Hva er din stillingstittel og/eller rolle i forhold til utviklingsteam(ene)? *

Hva slags studiebakgrunn har du? *

Hvilken bedrift jobber du i? *

Inkluder gjerne hvilken bedrift du er ansatt i og hvilken bedrift du eventuelt er utleid til.

Hvor lang arbeidserfaring har du? *

Indiker antall år.

Se nylige endringer i Nettskj

Appendix B
Interview guide for selection 1 and 2

Utvalg 1 - Intervjuguide (Utviklere)

Innledende samtale / Bakgrunnsinformasjon

1. Fortell litt om deg selv og hva slags arbeidsbakgrunn du har?
    a. I korte trekk, hva slags prosjekt jobber du på nå? Hvor lenge har prosjektet pågått? Hvor lenge har du vært med?
2. Hva er din erfaring med, eller forhold til informasjonssikkerhet?
3. Hvilke sikkerhetsaktiviteter utføres av teamets medlemmer?

Hoveddel

4. Fortell meg litt om hvordan teamet ditt er bygget opp. Hvem gjør hva i teamet?
5. Hvem er det som setter premissene for sikkerhet i prosjektet? (Kunde, internt, dedikert rolle)
6. Kan du fortelle meg litt om samarbeidet mellom teamet ditt og de som setter sikkerhetskravene til prosjektet?
    a. Hvordan er kommunikasjonen? (hvor ofte, digitalt eller fysisk?)
    b. Hvem på teamet er inkludert i den kommunikasjonen? (tilbakeholden, åpent, lukket)
7. Hvordan vil du beskrive tilbakemeldingskulturen i teamet/ fra premissgiver? Gir folk positive og negative tilbakemeldinger til hverandre? (Hva er fordelingen? Oppleves noe av det som kritikk fra premissgiver?)
8. Er det åpenhet for å ytre bekymringer eller sikkerhetsrelaterte ideer til premissgiver? Hvorfor / hvorfor ikke?
9. Hvordan forholder teamet seg til sikkerhetskrav fra premissgivere? (Overholdes frister og krav, balansen mellom tid og krav? er det realistiske deadlines)
10. Hva er teamets holdning til å utføre sikkerhetsarbeid? (Interesse? Viktig/Uviktig? Engasjerende eller noe en må gjøre, hvordan påvirker det sikkerhetsarbeidet?)
11. Har du opplevd at sikkerhetskrav er uklare eller unødvendige? (Hvorfor? Hvordan løser du det? Hvem får du hjelp av? Hvis uklare; er kravene rettferdiggjort? hva kjennetegner gode krav, får du en prioritert liste m/krav?)
12. Kan du fortelle meg om en gang du har opplevd gnisninger mellom utviklerteamet og sikkerhetsansvarlige? (Utdyp og følg opp!)
    a. Hvordan? / Hvorfor ikke?

    b. Har dette hatt noen konsekvenser for sikkerhetsarbeidet?
    c. Var det aspekter av samarbeidet som kunne ha vært annerledes?

13. Kan du fortelle om en gang det har vært et tilfelle av mangelfullt sikkerhetsarbeid relatert til et prosjekt du har kjennskap til? (Var det noe snakk om hvem sin feil det var?)
    a. Hva tror du utløste det?

    b. Hvordan ble premissgiver involvert? eller ikke?

    c. Påvirket det samarbeidet til sikkerhetsansvarlige? Hvordan?

    d. Hvordan håndterte teamet/premissgiver det sammen?

    e. (Ble det gjort tiltak i etterkant for å sikre at det ikke skjer igjen?)

    f. Som du kjenner til, har mangelfullt sikkerhetsarbeid resultert i et sikkerhetsbrudd?

14. I hvilken grad føler du sikkerhetsansvarlig har forståelse for teamets arbeidsoppgaver/arbeidshverdag?
    Hvordan?
    Hvilke aspekter?
    Hva har de mye/lite forståelse for?

15. Hvordan opplever du premissgivers kompetanse innen sikkerhetsarbeidet? (Policy eller programvaresikkerhet

16. Hvordan vil du beskrive sikkerhetsansvarlig sin oppfølging av oppgaver dere gjør? (Hvordan? monitorering / kontroll)

    a. Hvordan påvirker monitoreringen (el. mangel) sikkerhetsarbeidet i teamet?

Avsluttende del

17. Hva tenker du er den største suksessfaktoren for et godt samarbeid mellom teamet og premissgiver for sikkerhet?

18. Hva kjennetegner en god sikkerhetsansvarlig?

Utvalg 2 - Intervjuguide (Premissgiver)

Introduksjon

1. Fortell litt om deg selv og hva slags arbeidsbakgrunn du har?
2. Kan du fortelle litt om hva din rolle er i det prosjektet du jobber med nå?
3. Hvilke team forholder du deg til?
    a. Hvor lenge har du jobbet med teamet?
4. Hva er ditt ansvarsområde for teamene? (hva slags type sikkerhet?)
5. Hvor lenge har du jobbet i din nåværende rolle?

Hoveddel

6. Setter du premisser for sikkerhet til teamet? Hvordan? Hvilke type krav? Hvem andre setter premisser?
7. Hva er teamenes ansvar når det kommer til sikkerhet i løsningene de utvikler? (sikkerhetsaktiviteter, programvaresikkerhet?)
8. Kan du fortelle meg litt om samarbeidet mellom deg og teamene dine?
    a. Hvordan er kommunikasjonen? (hvor ofte, digitalt eller fysisk, når?)
    b. Hvem på teamet er inkludert i den kommunikasjonen? (tilbakeholden, åpent, lukket)
9. Fortell litt om hvordan du følger opp sikkerhetsarbeidet i teamene? Hvordan fungerer det? Hvordan påvirker det sikkerhetsarbeidet deres? Hvordan tilbakemeldinger gir du?
10. Når det kommer til å forankre sikkerhetskrav til teamene, hvordan gjør du det, og oppleves det som nødvendig?
11. Hva slags type tilbakemeldinger får du på sikkerhetskravene du stiller fra teamene? Er de forståelige?
12. I hvilken grad føler du at du har innblikk i teamets arbeidsoppgaver/arbeidshverdag?
    a. Påvirker det hvordan du forholder deg til teamet?
    Hvordan?
    Hvilke aspekter?
    Hva har de mye/lite forståelse for?
13. Kan du fortelle meg om en gang du har opplevd gnisninger mellom utviklerteamet og deg som sikkerhetsansvarlige? (Utdyp og følg opp!)
    a. Hvordan? / Hvordan har dere klart å unngå det?
    b. Hva tror du er grunnen til gnisninger?

    c. Har dette hatt noen konsekvenser for sikkerhetsarbeidet?
    d. Var det aspekter av samarbeidet som kunne ha vært annerledes?

14. Kan du fortelle om en gang det har vært et tilfelle av mangelfullt sikkerhetsarbeid eller sikkerhetsbrudd relatert til et prosjekt? (Var det noe snakk om hvem sin feil det var?)
    a. Hva tror du utløste det? (hvor glapp det?)

    b. Hvordan påvirket det relasjonen til teamet?

    c. Ble det gjort tiltak i etterkant for å sikre at det ikke skjer igjen?

15. Hvordan opplever du teamenes holdning til å gjøre sikkerhetsarbeid?

16. Hvordan opplever du teamenes generelle kompentanse i deres felt?

Avslutning

17. Hva tenker du er den største suksessfaktoren for et godt samarbeid mellom deg og utviklingsteamet du setter premisser for?

Appendix C
Notification form to Norwegian centre for research data

NSD NORSK SENTER FOR FORSKNINGSDATA

Meldeskjema

Referansenummer

384152

Hvilke personopplysninger skal du behandle?

- Navn (også ved signatur/samtykke)
- E-postadresse, IP-adresse eller annen nettidentifikator
- Bilder eller videoopptak av personer
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Beskriv hvilke bakgrunnsopplysninger du skal behandle

Jobbtittel, arbeidsgiver, studiebakgrunn, tidligere jobberfaring

Prosjektinformasjon

Prosjekttittel

Masteroppgave om tillitsrelasjoner i utviklingsteam

Prosjektbeskrivelse

Formålet med prosjektet er å undersøke hvordan tillitsrelasjoner i team påvirker sikkerhetsarbeid i utviklingsteam. Dette er et prosjekt i forbindelse med gjennomføring av masteroppgave ved Kommunikasjonsteknologi og Digital sikkerhet ved NTNU.

Begrunn behovet for å behandle personopplysningene

Personopplysninger samles inn for å skaffe nok relevant kontekst om deltagerne i jobbsammenheng. Det vil bli gjennomført personlige intervjuer, og for å kunne behandle datainnsamlingen i ettertid, er det ønskelig med videoopptak/lydopptak.

Ekstern finansiering

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Helga Tenold Fridtun, helgatf@stud.ntnu.no, tlf: 90577655

Behandlingsansvar

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for informasjonsteknologi og elektroteknikk (IE) / Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Maria Bartnes, maria.bartnes@sintef.no, tlf: 45218102

Skal behandlingsansvaret deles med andre institusjoner (felles behandlingsansvarlige)?

Nei

Utvalg 1

Beskriv utvalget

Utviklere / teammedlemmer

Rekruttering eller trekking av utvalget

Utvalget rekrutteres gjennom eget nettverk og gjennom veileders arbeidsplass innad i bedriften. Førstegangskontakten går via mail/slack, og deretter vil de inviteres til deltagelse via et elektronisk samtykkeskjema.

Alder

25 - 65

Inngår det voksne (18 år +) i utvalget som ikke kan samtykke selv?

Nei

Personopplysninger for utvalg 1

- Navn (også ved signatur/samtykke)
- E-postadresse, IP-adresse eller annen nettidentifikator
- Bilder eller videoopptak av personer
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Hvordan samler du inn data fra utvalg 1?

Personlig intervju

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Samtykke (art. 6 nr. 1 bokstav a)

Informasjon for utvalg 1

Informerer du utvalget om behandlingen av opplysningene?

Ja

Hvordan?

Skriftlig informasjon (papir eller elektronisk)

Utvalg 2

Beskriv utvalget

Ekstern stakeholder utenfor teamet

Rekruttering eller trekking av utvalget

Utvalget rekrutteres gjennom eget nettverk og gjennom veileders arbeidsplass innad i bedriften. Førstegangskontakten går via mail/slack, og deretter vil de inviteres til deltagelse via et samtykkeskjema.

Alder

25 - 65

Inngår det voksne (18 år +) i utvalget som ikke kan samtykke selv?

Nei

Personopplysninger for utvalg 2

- Navn (også ved signatur/samtykke)
- E-postadresse, IP-adresse eller annen nettidentifikator
- Bilder eller videoopptak av personer
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Hvordan samler du inn data fra utvalg 2?

Personlig intervju

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Samtykke (art. 6 nr. 1 bokstav a)

Informasjon for utvalg 2

Informerer du utvalget om behandlingen av opplysningene?

Ja

Hvordan?

Skriftlig informasjon (papir eller elektronisk)

Tredjepersoner

Skal du behandle personopplysninger om tredjepersoner?

Nei

Dokumentasjon

Hvordan dokumenteres samtykkene?

- Elektronisk (e-post, e-skjema, digital signatur)

Hvordan kan samtykket trekkes tilbake?

En registrert deltaker kan trekke samtykke ved å sende en mail til ansvarlige personer for prosjektet som er oppgitt i samtykkeskjema.

Hvordan kan de registrerte få innsyn, rettet eller slettet opplysninger om seg selv?

En registrert deltaker kan få innsyn, rettet eller slettet opplysninger om seg selv ved å sende en mail til ansvarlige personer for prosjektet som er oppgitt i samtykkeskjema.

Totalt antall registrerte i prosjektet

1-99

Tillatelser

Skal du innhente følgende godkjenninger eller tillatelser for prosjektet?

Behandling

Hvor behandles opplysningene?

- Mobile enheter tilhørende behandlingsansvarlig institusjon
- Maskinvare tilhørende behandlingsansvarlig institusjon

Hvem behandler/har tilgang til opplysningene?

- Student (studentprosjekt)
- Prosjektansvarlig

Tilgjengeliggjøres opplysningene utenfor EU/EØS til en tredjestat eller internasjonal organisasjon?

Nei

Sikkerhet

Oppbevares personopplysningene atskilt fra øvrige data (koblingsnøkkel)?

Ja

Hvilke tekniske og fysiske tiltak sikrer personopplysningene?

- Opplysningene anonymiseres fortløpende
- Adgangsbegrensning
- Endringslogg

Varighet

Prosjektperiode

10.01.2022 - 13.06.2022

Skal data med personopplysninger oppbevares utover prosjektperioden?

Nei, alle data slettes innen prosjektslutt

Vil de registrerte kunne identifiseres (direkte eller indirekte) i oppgave/avhandling/øvrige publikasjoner fra prosjektet?

Nei

Tilleggsopplysninger

Knut Formo Buene & Helga Tenold Fridtun

Investigating trust relationships between software development teams and information security stakeholders

NTNU
Kunnskap for en bedre verden