UPPSALA
UNIVERSITET

UPTEC STS 22011
Examensarbete 30 hp
Juni 2022

Integrating security into agile software development

A case study on the role of inertia

Andersson, Rasmus
Edström, Carl

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA
UNIVERSITET

Integrating security into agile software development

Andersson, Rasmus
Edström, Carl

Abstract

The security directives at Ericsson Group IT have recently been re-worked to apply to modern security requirements. For Ericsson's software development teams developing internal applications, security tools have been implemented into the daily workflow to follow these new directives. Before, security mainly was considered during the reviews and scheduled assessments of the software projects. The goal of these new tools is to add security to every part of the software development process. Security thus adds to the scope of work of the developers at Ericsson Group IT, which has, in the past, evolved from being solely a developer to being responsible for development and operations to development, security and operations.

However, adding methods and tools to the developer's workflow can create *inertia* and friction in daily work. We intend to apply the concept of inertia to agile work practices to examine how small-scale projects are affected when new security tools and methods are introduced and implemented in the agile workflow. Research suggests that linked processes and methods should be put in place to achieve desirable results from the implemented tools and be integrated into the team's agile methodologies. The thesis aims to identify the factors that affect inertia by investigating and analysing the developers' use of methods and tools.

As for data collection, a pilot study and a case study were applied to a team at Ericsson Group IT. The data was collected through qualitative surveys conducted on twelve proven factors regarding successfulness in work implementations. The data was then analysed through the Gioia methodology by compiling the collected data into first-order concepts and linking them to familiar second-order themes. These themes were then translated into aggregate dimensions synthesised from the study's theoretical framework.

The results showed that several factors affected the change process: personnel training and education, appropriate communication, and adaptability to the change process. These are all factors attributing inertia to the change process, and awareness of these can help mitigate and facilitate a successful change process. Streamlining successful change processes is vital when integrating security as a requirement into an agile software development team.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: David Neess Ämnesgranskare: Anders Arweström Jansson

Examinator: Elísabet Andrésdóttir

Populärvetenskaplig sammanfattning

Säkerhetsdirektiven på Ericsson Group IT har nyligen omarbetats för att gälla moderna säkerhetskrav. För mjukvaruutvecklingsteamen som utvecklar interna applikationer har säkerhetsverktyg implementerats i det dagliga arbetsflödet för att följa dessa nya direktiv. Detta, till skillnad från tidigare, då säkerheten främst togs i beaktning vid granskningar och schemalagda utvärderingar av programvaruprojekten. Målet med denna implementation av nya verktyg är att lägga till säkerhet till varje del av mjukvaruutvecklingsprocessen, genom att införa till automatiska processer som testar mjukvaran för eventuella sårbarheter. Detta för att säkerställa säker och kvalitativt god mjukvaruutveckling. Säkerhet utökar därmed arbetsomfånget för utvecklarna på Group IT, som tidigare har växt från att enbart vara utvecklare till att ansvara för utveckling och drift till utveckling, säkerhet och drift.

Att lägga till verktyg i utvecklarens arbetsflöde kan dock skapa *tröghet* och friktion i det dagliga arbetet. Vi avser att applicera begreppet tröghet till en agil arbetsprocess för att undersöka utvecklarens relation och användning av dessa verktyg och de metoder som verktygen verkar inom. Forskning tyder på att kopplade processer och metoder bör införas för att uppnå önskvärda resultat från de implementerade verktygen och integreras i teamets agila metoder. Målet med studien är att identifiera vilka faktorer som påverkar tröghet genom att undersöka och analysera utvecklares användning av metoder och verktyg.

När det gäller datainsamling innehåller studien en pilotstudie och en fallstudie som har tillämpats på ett team på Ericsson Group IT i Borås. Data samlades in genom enkätundersökningar som baserades på tolv beprövade faktorer för framgång i arbetsimplementeringar. Data analyserades därefter genom att sammanställa den insamlade datan till nyckelord och dessa kopplades följaktligen till relevanta teman. Dessa teman översattes sedan till aggregerade dimensioner som växt fram ur studiens teoretiska ramverk. Detta arbetssätt skapar en tydlig översikt och struktur av datainsamlingen vilket gynnar sammanställningen och analysen av studiens resultat.

Resultaten visade att flera faktorer påverkade förändringsprocessen: personalutbildning, situationsanpassad kommunikation och slutligen anpassningsförmåga till förändringsprocessen. Dessa är faktorer när man tillskriver förändringsprocessen tröghet och medvetenhet om dessa kan hjälpa till att mildra och underlätta en framgångsrik förändringsprocess. Tillhandahålla framgångsrika förändringsprocesser är avgörande när man integrerar säkerhet som ett krav i ett agilt mjukvaruutvecklingsteam.

Förord

Detta examensarbete markerar avslutningen på vår civilingenjörsutbildning i system i teknik och samhälle med inriktningen informationsteknik vid Uppsala universitet. Examensarbetet omfattar 30 högskolepoäng och har utförts hos Ericsson AB, på avdelningen Group IT i Borås, av Rasmus Andersson och Carl Edström under vårterminen 2022.

Vi vill börja med att rikta ett stort tack till vår ämnesgranskare Anders Arweström Jansson vid Uppsala universitet, professor vid institutionen informationsteknologi, visuell information och interaktion. Din roll som bollplank har hjälpt arbetet framåt och till det bättre.

Vi vill tacka Ericsson AB i Borås för att vi fått äran att utföra vårt examensarbete hos er. Vi vill även tacka vårt team, UDB, och alla respondenter för att vi fått låna er tid och att vi fått möjligheten att ta del av er expertis inom området.

Slutligen vill vi tacka vår handledare David Neess för ditt stora engagemang och för att du tagit dig tid för oss genom hela arbetet. Vi vill också passa på att tacka vår mentor Glenn Wadstedt. Tack för alla kontinuerliga samtal som hjälpt oss framåt och möjliggjort att ro arbetet i hamn.

Rasmus Andersson & Carl Edström
Uppsala, juni 2022

Glossary

API Application Programming Interface (API) is a software intermediary that allows two applications to talk to each other. 9

Azure Azure is a cloud computing service operated by Microsoft for application management via Microsoft-managed data centres. 6, 7, 25

CI/CD CI/CD is the combined practices of continuous integration (CI) and continuous delivery (CD). 9

CIA triad CIA stands for Confidentiality, Integrity, and Availability (CIA) and are often referred to as the CIA triad, ensuring that information is not compromised when critical issues arise. 24

DAST Dynamic Application Security Testing (DAST) is the process of analysing a web application through the front-end to find vulnerabilities through simulated attacks. 5

DevOps DevOps is a set of practice that combines software development (Dev) and IT operations (Ops). 4–7

DevSecOps DevOps is a set of practices that combines software development (Dev), security practices (Sec) and IT operations (Ops). 4, 5

Fortify Fortify is a static and dynamic application testing service offered by Micro Focus. 1, 5, 18

FOSS Free and Open-Source Software (FOSS) is freely licensed to use, copy, study, and modify the software. 24

IDE An Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. 9

Pipeline Pipeline is a set of automated processes and tools within the DevOps that allows developers to collaborate on building and deploying code to a production environment. 5, 9, 22–24

Pull Request Pull Request is an event in software development when a developer is ready to begin merging new code changes to the codebase. 5, 25

SAST Static Application Security Testing (SAST) is used to secure software by reviewing the source code of the software to identify sources of vulnerabilities. 5

SCA Software Composition Analysis (SCA) is a methodology to provide users better visibility into the open-source inventory of their applications. 5

Scrum Scrum is a framework for developing, delivering, and sustaining complex products through an agile way of work. 5, 7, 8

SME A Security Subject Matter Expert (SME) is a person with extensive knowledge, expertise, and experience within security. 1

Sprint Sprint cycle is an agile method which is a timeboxed period when a team delivers a set amount of work, often spanning over two to four weeks. 8, 13, 18

SSL Secure Sockets Layer (SSL) certificate is a digital certificate that authenticates a website's identity and enables an encrypted connection. 24, 30, 32

UI User Interface (UI) is the point of human-computer interaction and communication in a device. 24

UX User Experience (UX) is how a user interacts with and experiences a product, system or service. 5

Acronyms

API Application Programming Interface. 9, 23, *Glossary:* API

DAST Dynamic Application Security Testing. 5, 9, *Glossary:* DAST

FOSS Free and Open-Source Software. 24, *Glossary:* FOSS

IDE Integrated Development Environment. 9, 22, *Glossary:* IDE

SAST Static Application Security Testing. 5, 9, *Glossary:* SAST

SCA Software Composition Analysis. 5, 9, *Glossary:* SCA

SME Security Subject Matter Expert. 1, *Glossary:* SME

UI User Interface. 24, *Glossary:* UI

UX User Experience. 5, *Glossary:* UX

Contents

1 Introduction **1**
- 1.1 Purpose, scope and research question . . . 2
- 1.2 Delimitations . . . 2

2 Background **4**
- 2.1 Organisational background . . . 4
- 2.2 Processes and tools . . . 7

3 Theory **10**
- 3.1 Organisational change management . . . 10
- 3.2 Transforming sociotechnical systems . . . 13

4 Method **17**
- 4.1 Method choice . . . 17
- 4.2 Studies . . . 18
- 4.3 Study implementation . . . 18
- 4.4 Data analysis . . . 20
- 4.5 Method criticism . . . 20

5 Results **22**
- 5.1 Pilot study . . . 22
- 5.2 Case study . . . 25
- 5.3 Summary . . . 28

6 Analysis **29**
- 6.1 Pilot study . . . 29
- 6.2 Case study . . . 32

7 Discussion **35**
- 7.1 Work management . . . 35

7.2 Requirements and needs 37
7.3 Synthesising the concept of inertia 39

8 Conclusion **41**

Bibliography **44**

A Appendix **48**
A.1 Pilot study 48
A.2 Case study 50

Introduction

The development of internal applications at Ericsson Group IT adheres to different types of security assessments. The applications must conform to current directives before being used as internal applications at Ericsson, and an Ericsson Security Subject Matter Expert (SME) must review the application.

Group IT's current directives include security in the development workflow and not just in the reviews and scheduled assessments. There are software solutions to achieve this, such as Fortify, which scans the code base automatically and thus enables instant and continuous security assessments. In this way, security as a new practice will be a part of the workflow and the daily routines (Microsoft Corporation, 2016). The developers' scope of work has evolved from the developers being responsible for the development and operations of the infrastructure, which has grown from purely being accountable for the development. In other words, a developer has gone from being solely a developer to being responsible for both operations and security. Therefore, the developer is constantly exposed to new methods and tools to keep up with development evolution, demanding that the developer handle new and continually changing requirements (Neess, 2022).

Previous research regarding the implementation of routines and tools has shown that awareness of inertia in sociotechnical change processes can enable and facilitate a successful transformation (Lind, 2017). A *sociotechnical system* is defined as the interaction between technology and people in a social context. Awareness of what Lind (2017) describes as *inertia* in a sociotechnical system, such as an IT-related change process, can potentially increase the chances of successfully achieving the goal of the said change process. More concretely, Lind (2017, p. 45) describes *inertia* as "the impact of those characteristics of a sociotechnical system that affect the effectiveness and ef-

ficiency of a specified change process in the system". Further, Lind (2017) argues that inertia can provide a sociotechnical perspective often lacking in today's IT-related change processes (Lind, 2017).

1.1 Purpose, scope and research question

Given the lack of research regarding inertia in agile change processes, we intend to apply the concept of inertia to agile work practices to examine how small-scale projects are affected when new security routines and tools are introduced and implemented in the agile workflow.

- The concept of inertia in agile change processes has little research. Therefore is this relevant to examine further.
- As a development team has evolved from solely focusing on development to development and operations, and now development, security and operations, we believe introducing the concept of inertia can help understand how a development team is affected when new security tools and methods are added to the agile workflow.
- Highlighting this can reveal what obstacles can arise and hopefully how to mitigate or avoid them.

Additionally, the study aspires to illustrate the dominant experience regarding safety routines for those involved in the individual development team.We strive to fulfil the study's purpose by performing a case study on a software development team at Ericsson Group IT in Borås. The team consists of six team members.

The study's research question is:

> How can the concept of inertia provide a theoretical lens through which the change process can be understood better, i.e. which factors describe the effects on an agile software development team when new secuirty tools and methods are included in the project requirements?

1.2 Delimitations

Ericsson is a global company with a vast quantity of departments and teams. By focusing on a small development team, our idea is to gain knowledge from and develop an approach from the developer's perspective. A multiple case

study based on more than one team or a more extensive project team could improve the generalizability of the result. However, we focused on a small group because of time constraints and the manageability of the study. Given the time frame and resources, a larger group would have been too great for this study.

Background

This chapter is divided into two parts. The first section, organisational background*, offers a description of the studied organisation and its goals and how they wish to fulfil them. The second section,* processes and tools*, presents the necessary knowledge about the technical tools which were to be implemented and the working processes which were to enable the implementation itself.*

2.1 Organisational background

Organisations must deliver and operate software quickly and reliably to meet the accelerating demand in an ever-changing industry. The faster the team can change the software, and the sooner one can deliver value to the customer, run experiments, and receive valuable feedback. Five metrics capture operational capabilities: deployment frequency, lead time for changes, time to restore service, change failure rate and reliability. These five metrics, with cluster analysis, reveal four distinct performance profiles, namely elite, high, medium, and low (Alphabet Inc, 2021).

The evolution of the development teams at Ericsson Group IT is that they are becoming more self-sufficient and given an extended responsibility for their work. From only working with development to also working with operations, that is, operation of the infrastructure of the applications, DevOps The evolution has continued together with the modern requirements and now also includes the responsibility of security. The developers are now also given the security responsibility and thus are responsible for development, security and operations. That is, DevSecOps (Neess, 2022).

Ericsson has formulated an innovation plan, *imagine possible*, to meet these modern demands for their DevSecOps teams. The plan's core is to upgrade all development teams within Ericsson to elite performance (Lambert, 2022).

Elite performance is desirable due to the significant difference in performance compared to a low-profile DevOps team. The number of code deployments increases, lead time from committing to deploy becomes faster. The time to recover from code incidents excels, and the change failure rate decreases (Alphabet Inc, 2021).

The five metrics regarding operational capabilities have also been included and applied to *imagine possible*. The implementation has resulted in three keywords: throughput, stability, and behaviour, linked to the five metrics. Throughput measures lead time and deployment period. Stability controls the change fail rate and time to restore. Behaviour focuses on cooperation and collaboration, empathy and humanness, execution speed, and a speak-up environment (Lambert, 2022).

The IT department at the site in Borås has worked towards the goals of *imagine possible* by upgrading twenty software development teams to "elite DevOps teams" by 2024. Further, they see the need to shorten the feedback loops to their developers regarding the security testing. Today, security testing is not a self-service; it is done by requests in the reviews and scheduled assessments, resulting in long lead times. The long wait time is partly because of the dependency of external parties, as the development team does not conduct security testing themselves. Therefore, they aim to lower or remove this dependency and work towards more autonomous development teams regarding security testing, the "Sec" in DevSecOps.

Ericsson Group IT intends to implement security tools and methods into the daily workflow to achieve this goal. For example, Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) are implemented into the developers' pipelines. Such tools enable security feedback to developers within minutes after a pull request has been made. A shorter feedback loop will allow developers to iterate their solution and mitigate any security issues/warnings with higher speed and quality. Ericsson's first step is testing and experimenting with Fortify to be concreated into their pipelines. Moreover, they also want to examine how implementing these tools affects their working methods within Scrum (Neess, 2022).

One of the twenty teams in Borås that work towards becoming *elite* is UDB, a software development team. UDB is a team in the IT section at the Borås site where these new security directives are applied and have thus implemented these tools into their workflow. The team consists of six team members: a tester of the User Experience (UX), four developers, and a product owner. An external security team handles the tools and executes the im-

plementation through a handover phase. In this phase, the security team introduced the tools into the development team's workflow through a process with transitional ownership of the different domains of the tools. When this phase is completed, the aim is to enter the self-service phase. This phase includes more ownership of the security tools by the development team, and therefore more responsibility is moved to the developers. The handover- and self-service phases are illustrated in two figures in an ideal scenario. The handover- and self-service phases show how a successful change process is conducted in an ideal scenario.

The handover phase is illustrated in Figure 1, which shows the ownership areas during the handover phase. Grey is the project team, UDB, blue is the team responsible for the Azure DevOps environment, and yellow is the security team that are responsible for the implementation of the security tools. During the handover phase, the security team owns the responsibility of the tool, the scanning results and policy changes. Which means that potential problems with the security tool will be handled by the security team. The security team owns the responsibility of the security tool itself and its version updates.

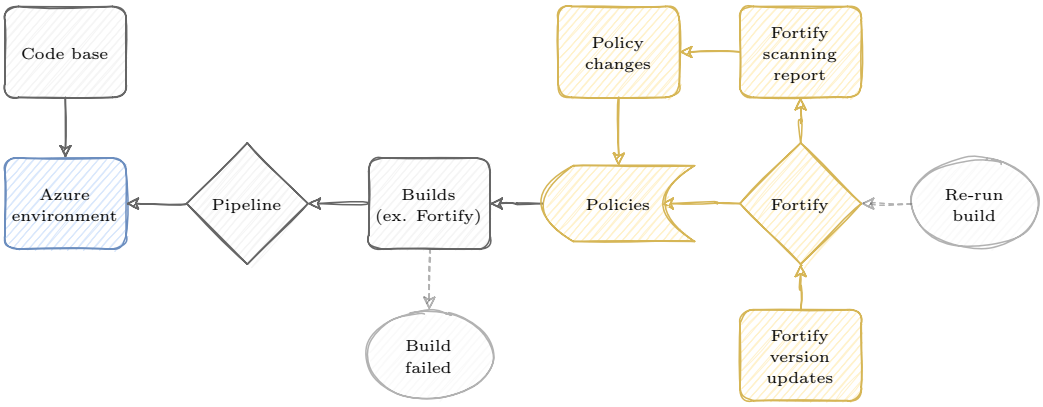

Figure 1: Illustrates the change project and the ownership areas during the handover phase. Grey is the project team, blue is the Azure DevOps team, and yellow is the security team.

The handover phase was to be moved idealy over to a self-service phase in which the development team has more control over the tools and its processes. This so they can be more independent with less dependency on external teams. The self-service phase is illustrated in Figure 2, where some of the ownership domains have been transferred to the development team. The

development team owns the scanning results and any policy changes, whereas the security team still owns the responsibility of the security tool itself and its version updates.

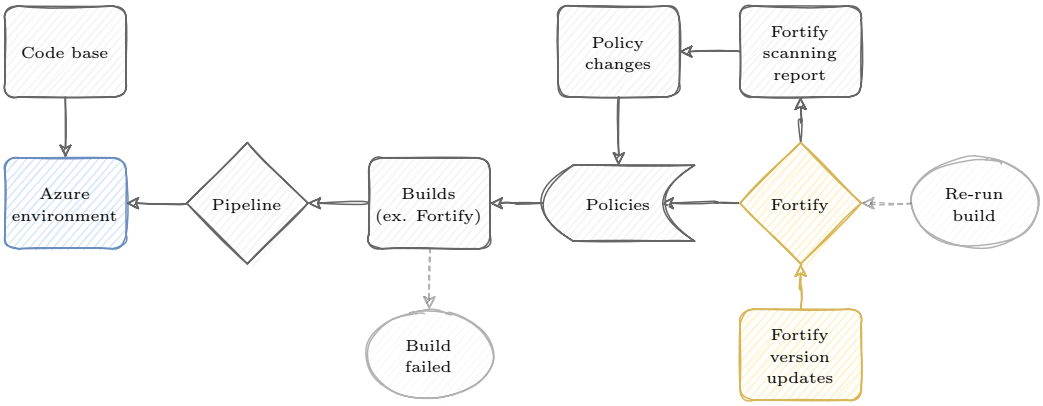

Figure 2: Illustrates the change project and the ownership areas during the self-service phase. Grey is the project team, blue is the Azure DevOps team, and yellow is the security team.

2.2 Processes and tools

The transition to more secure and sustainable code for the software teams at Ericsson Borås has been made possible through several processes and tools, such as an agile way of working in which the specific security tools are implemented. The tools are introduced by working with the Scrum framework, which enables a planned and well-executed integration of the tools into the developers' toolbox.

2.2.1 Agile way of work

Scrum is a framework for developing, delivering, and sustaining complex products, such as software development projects, according to the authors Schwaber and Sutherland (2017). The authors developed the system during the early '90s. They claim that the framework enables the delivery of products such as software projects with the highest possible value whilst retaining productivity and creativity (Schwaber and Sutherland, 2017). The framework provides an agile way of work which Ericsson has adopted to their software teams at the site in Borås. The framework includes several tools, such as the *backlog*, enabling structured and well-executed planning.

Schwaber and Sutherland (2017) define the backlog as "an ordered list of everything known to be needed in the product. It is the single source of requirements for any change to be made to the product". The backlog is, in other words, filled with tasks that should go into the product or the software project, and the developers then complete these in a settled order. The order is decided between sprints, which the Scrum framework defines as fixed periods (Schwaber and Sutherland, 2017).

The sprint backlog is a subset of items in the product backlog selected to be worked on for a fixed period. In the Scrum framework, the product owner is responsible for the content and order of the product backlog. This responsibility can be taken on by the team as well. The items in the backlog change continuously, and new items are added, dropped or changed. Schwaber and Sutherland (2017), Scrum creators, explain that a product backlog is never complete. The earliest development of the product backlog lays out the initially known and best-understood requirements. It evolves as the product and the environment it will be used in develops. The product backlog is dynamic and constantly changing to identify what the product needs to be appropriate, competitive and valuable (Schwaber and Sutherland, 2017).

The Scrum framework can also be applied when introducing and implementing new tools into the developers' workflow. The implementation can be split up into tasks that go into the backlog and, lastly, into either a specific sprint or spanning several sprints. Developers are then enabled to introduce the new tools step-by-step in an agile manner and slowly and steadily work towards integrating the tools completely into the workflow.

2.2.2 Security tools

Traditionally, security within software development was only regarded during the end of the development lifecycle, that is, before putting the software out in production. Today, security has gotten more extensive attention and must now be a part of every phase of the software development lifecycle. Security tools must be implemented and included in the developer's toolbox to allow for the new security requirements. Furthermore, while development teams move away from traditional waterfall methods to an agile way of working, such as Scrum, security must follow this shift. The agile process speeds up the development and production, which has also increased the usage of open-source code, which allows for faster development and better cost-effectiveness. More rapid development and more extensive dependence on open-source deepens the importance of security during the whole development lifecycle (Abasi, 2021; IBM Cloud Education, 2020; Microsoft

Corporation, 2016).

There are several types of security tools which can be included in the developers' toolbox. Such tools can, for example, be tools that scan for vulnerabilities during code writing, after a coding session, or tools that scan the codebase as a whole. One example of a method that scans during code writing is tools that use SAST, which provides developers with real-time feedback for their code regarding security issues within the codebase. Such tools can be added to the Integrated Development Environment (IDE) and thus integrate into the developer's workflow. Adding tools to the IDE enables ongoing feedback regarding security issues in the code that the developer can resolve without waiting for scans, providing a faster workflow (Micro Focus, 2022a).

Moreover, SAST can also be implemented into the CI/CD pipeline, allowing for additional security through scans of the codebase. Such implementation can be done together or independent of the SAST plugin in the IDE. One benefit of implementation in the pipeline is that the developers are not required to install anything onto their machines (Micro Focus, 2022c).

Other tools can be added to the developers' toolbox, for example, DAST, which scans for vulnerabilities outside the codebase, such as vulnerabilities found in third-party Application Programming Interface (API)s. DAST applies simulated attacks to find susceptibilities in the application (Micro Focus, 2022a). Another example of a tool is SCA, which scans open-source software. Open source is commonly used in modern codebases and is often a large part of the code, for example, plugins. The open-source code can have vulnerabilities, and an SCA can be used to find these vulnerabilities. The SCA tool scans the code base, detects the open-source code, and scans it for current known vulnerabilities (Micro Focus, 2022b).

Theory

This chapter is divided into two parts. The first section, organisational change management*, introduces a general view of change management. The second section,* transforming sociotechnical systems*, presents a background to the concept of sociotechnical systems and how to transform them. Further, the theory introduces and explains relevant subjects related to the case study.*

3.1 Organisational change management

Changes affect people, and people are by default pessimistic about change. A common goal can assist a change process, but it can be hard to understand a goal set up by an "outsider". A change process would be easy to implement if everyone acted in the same way. However, humans have different backgrounds, experiences, and viewpoints that interpret situations differently (Källberg, 2013). Above all, creating trust, promoting work morale, and providing good communication should be prioritised before internal processes and tools (Measey, 2015).

Change is also something that has gotten more extensive attention in today's work environment, where agile practices are standard. Adaptive change management is vital for the agile manifesto, which states, "we value responding to change over following a plan" (Beck et al., 2001, p. 1). Change management covers a wide range of approaches to transforming an organisation. These approaches are rooted outside software development. Jurgen Appelo states that change management is an approach for transforming an organisation by transitioning individuals, teams, and even the whole business in a specific direction. Perhaps transformational management would be a more accurate term (Appelo, 2011).

3.1.1 Change management processes

There are a lot of different change management processes. *Leading change* is a paper written by Kotter (2012) that provides an eight-step process for the transformation effort. The method starts with (1) establishing a sense of urgency, (2) creating a guiding coalition, (3) developing a vision and strategy, (4) communicating the change vision, (5) empowering employees for broad-based action, (6) generating short-term wins, (7) sustaining acceleration by consolidation gains and producing more change and (8) anchoring new approaches in the culture (Kotter, 2012).

ADKAR is a similar model, written by Hiatt (2006). ADKAR is an acronym for (A) awareness of the need for change, (D) desire to support and participate in the transformation, (K) knowledge of how to change, (A) the ability to implement requires skills and behaviours, and (R) reinforcement to sustain the change (Hiatt, 2006).

Whereas Kotter (2012) and (Hiatt, 2006) suggest a start and end of the transformation process with intermittent periods of stability, the PDCA cycle focuses more on continuous improvement. PDCA tries to maintain changes throughout the introduction of standards. Plan-Do-Check-Act-cycle, (PDCA), often called *The Deming cycle* after W. Edwards Deming (Koiesar, 1994), begins with planning a test or change to improve (P). Then that change is completed on a small scale (D), and the result is studied, what went well and what was learned (C). Lastly, the change adopts relinquishes (A), and the process repeats (Moen and Norman, 2006).

3.1.2 Adoption of change

Appelo (2011, 2012) explain the adoption of change to adapt to change. Like other innovations, change starts with initiators and innovators, followed by early adopters and the early majority, followed by the late majority and laggards (Rogers, 1995). However, change processes can fail, mainly at the so-called "chasm", the gap between early adopters and the early majority. Agile practices aim to deliver value for the customer fast and continuously; changing requirements in the backlog are essential to that. Changing customer demands requires that the process built to fulfil these demands also fluctuates. A consequence is that the organisation needs to change fundamentally. Therefore change management is a vital part of agile practices; it allows organisations themselves to adapt (Appelo, 2012).

3.1.3 Complex change and the adjacent possible

Managing change is a process and give the impression that successful change is a matter of execution and avoiding pitfalls. On the other hand, some changes cannot occur in a single step or at all, as Snowden (2015, 2016) argues. He asserts that three stages are necessary to achieve change within an organisation. First, mapping the current dispositional state of the system, identifying the attractors at play and their stability where attractors appear like clusters or behaviour patterns. These attractors forms from the interaction between the entities in the system (Snowden, 2016). Secondly, having mapped the system's state and identifying desirable patterns of behaviour adjacent to the current, Snowden draws on Kauffman's concept of the adjacent possibility (Snowden, 2016). The adjacent possible is the reachable adjacent state of the system concerning its current actual position (Kauffman, 2003). Identifying the "adjacent possible" is desirable because more significant shifts are more demanding or impossible or bring with them unintended consequences. Therefore, the present brings an evolutionary potential rather than endless possibilities that can be reached through incremental change (Snowden, 2015). Thirdly, Snowden concedes that some systems might not have an adjacent possible or that the energy required to escape the current attractor is too great. If this is the case, the recommended course of action is to disrupt the attractor to allow the natural emergence of a new adjacent possible. Figure 3 illustrates a system with four visible patterns of attractors. A possible transition to the centrally located adjacent can occur instead of transitioning from the present (actual) states to the goal state.

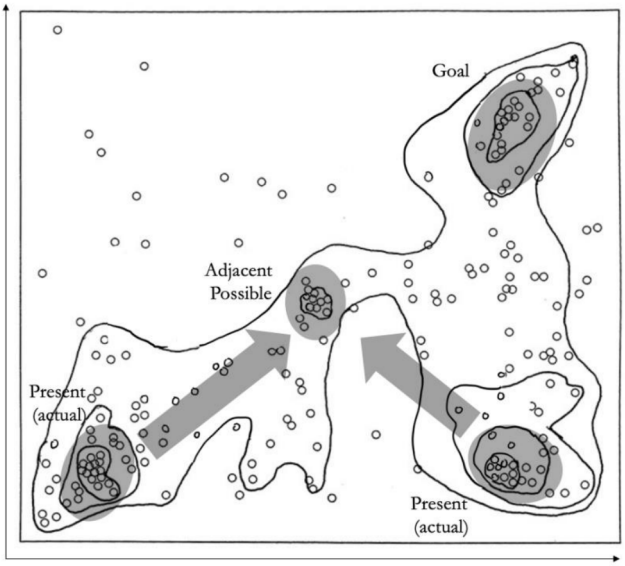

Figure 3: Two-dimensional illustration of the adjacent possible between the current actual state and the desired goal. The axes represent the relevant dimensions for change. The system is spread out, and parts of the system are already in the desired state. Adapted from Snowden (2016).

3.1.4 Continuous planning

In change management, continuous planning manifests the value of an agile work environment that welcomes change. As Inayat et al. (2015) explain, continuous planning is a routine task for agile teams (Jun et al., 2010). A team never sticks to a fixed plan, and they adapt to the upcoming changes from customers as the project progress. Possessing this flexibility facilitates changing requirements in later stages of projects (Inayat et al., 2015). Continuous planning is made possible through small increments of work rather than big releases. The priorities are revisited at regular intervals, such as at the end of a sprint. Continuous planning enables continuous reevaluation and helps scope the work to what is valuable to the customer. It makes it possible to revisit a problem later if the need for improvement arises (Inayat et al., 2015).

3.2 Transforming sociotechnical systems

A *sociotechnical system* is a system that can be seen as a combination of two interdependent subsystems - one social and one technical (L. Klein, 2014).

Society can consist of several interconnected sociotechnical systems of varying size and complexity. These include large societal functions, such as energy distribution or public transportation systems, to small firms, businesses, and even individual technology users. As a research tradition, sometimes referred to as "sociotechnical system design", it dates to the '40s and '50s and studies of coal mines in the United Kingdom (Griffith and Dougherty, 2001; Pasmore et al., 1982; van Eijnatten, 1993).

The impression that people and technology in an organisation are interdependent and that, because of this, the introduction of new technology needs to be performed with the consideration of an organisation's current structures and processes in mind is still being advocated (L. Klein, 2014):

3.2.1 Organisational inertia and the stability of societal functions

Besson and Rowe (2012) refer to inertia in organisations as an essence of the act of organising. To become organised entails the entrenchment of routines and patterns, and it is the entrenchment that becomes the source of inertia for the change process in which existing routines and patterns need to be changed (Besson and Rowe, 2012). The level of entrenchment is essentially a measure of permanence of, for example, formal and informal hierarchies, routines and procedures, and technical infrastructure. However, it also includes influences from less tangible factors such as people's norms and values, fears and beliefs, agendas, and vested interests (Besson and Rowe, 2012).

Similar to the entrenchment of organisational routines and patterns, but on a societal scale, is the description by Geels (2005) of the stability of the societal functions as resulting from networks of dependencies within them, observable as, for example, legal contracts, financial investments, social rituals, institutional arrangements and regulations. Geels (2005) explains that when reliance upon a specific technology grows, this contributes to its dominance and increases the stability of the sociotechnical systems of which the technology is a part. Once the technology has started to gain dominance, it may benefit from increasing returns and become increasingly embedded and relied upon in its societal or organisational context. An example of becoming embedded by dependencies is when the number of integrations between an organisation's information technologies increases. From a technological standpoint, creating and maintaining the integrations provides stability as the necessary expenditure of resources, e.g., time and programmers. This can grow exponentially with each new integration of a technological system.

From a social and organisational standpoint, the reliance of work processes on the increased efficiency is provided by the eventual loss of knowledge and experience related to how work was done before the integrations.

In addition to the definition of inertia by Besson and Rowe (2012), Lind (2017) defines *inertia* as "the impact of those characteristics of a sociotechnical system that affect the effectiveness and efficiency of a specified change process in the system". A definition that is general enough to accommodate all sociotechnical systems. Lind (2017) states that inertia is intimately tied to a specified change process. Therefore, it is necessary to establish the change process before analysing which system characteristics affect a particular process (Lind, 2017).

3.2.2 Twelve critical success factors for change management in information projects

There are several definitions of critical success factors (CSF); Leidecker and Bruno (1984) define CSFs as characteristics, conditions and variables. These should be adequately sustained, maintained, or managed to affect the success factors of an organisation competing in a specific industry (Leidecker and Bruno, 1984). Bullen and Rockart (1981) give a second definition and define the CSFs as the restricted number of fields in which positive outcomes will result in "successful competitive performance" for employees, organisational units, and an organisation as a whole. Ramaprasad and Williams (1998) state that CSFs should be used in three crucial areas: project management, information systems implementation, and requirements. Despite CSFs' existence, they have not been much explored in the literature regarding change management in information system projects.

Ziemba and Obłąk (2015) state twelve factors for this specific purpose. Top management support (1) means that you have active and visible support from a management team. Senior management's involvement and commitment are also crucial, in combination with the direct participation of the strategic decision-makers in the information system project. Recognising the change (2) means that the need for change has to be established and promote a positive approach to change. A shared vision (3) should be strongly advocated across the organisation. Planning a project as a change (4) indicates that one should evaluate the gap between where the organisation is now and where it would like to be. Manage an entire change process as a project, prepare a change management plan and promote change in the organisation. Managerial activity (5) involves managers directly associated

with the change process. Effective communication (6) means that one should communicate the change message on all levels throughout the organisation. Organisation readiness to deal with changes (7) implies that the employee needs to feel that the organisation is ready to deal with change to achieve an impression of safety. Employee training (8) means a clear demonstration of using the information system. Employee involvement (9) indicates that the employee needs to believe that the change is meaningful and impacts the organisation's success. Satisfaction is important, too; employee satisfaction (10) is connected to the final product and its acceptability by the employee. Information flow (11) is to have readily available and current data gathered in one place and available to all interested. Lastly, performance measurement (12) is the measure of change performance and value it to employees to demonstrate success (Aladwani, 2001; Chrusciel and Field, 2006; Cocks, 2014; Davenport et al., 2004; Graetz, 2000; Guimaraes et al., 1992; Hotek and White, 1999; Sutanto et al., 2008; P. S. Weber and J. E. Weber, 2001).

These twelve definitions are boiled down to the following twelve factors, according to Ziemba and Obłąk (2015):

1. The support of top management in both words and actions.
2. A recognised and well-defined need for change.
3. Clear objectives for what to change and a shared vision for how.
4. Project planning activities that clarify necessary tasks and required resources.
5. Managerial commitment and involvement at the line level.
6. Effective communication, regarding the particulars of the planned and performed changes as well as for sustaining engagement and motivation.
7. Organisational readiness to deal with the resulting changes.
8. Employee training to facilitate the transition to a new way of working.
9. Employee involvement in the change process.
10. Employee satisfaction with both planned and actual change outcomes.
11. Continuous information flow regarding the state of the change project.
12. Continuous performance measurements, evaluating the progress of the project against set goals and objectives.

Method

This chapter presents the methods used to conduct the study by motivating the methodology and the interview method. The data collection through surveys aims to create a clearer picture of how security work applies to internal small-scale projects at Ericsson Group IT in Borås.

4.1 Method choice

To carry out the study and determine how developers experience the transition to security work. Therefore, a qualitative method was chosen, rather than quantitative, providing broader data. Since this study analyses conversion to security work in the daily workflow and its work management, the study was carried out through a hermeneutic approach of an abductive nature. That is, empirical data were analysed against existing theories on the subject to be able to draw the most probable conclusion about the current phenomenon (Douven, 2017).

Further, there were six respondents in total, and the respondents were questioned in a pilot and a case study. The pilot study was executed to build a basic understanding of the working method and the relationship to safety-related issues. Moreover, to gain basic knowledge of what ideas and thoughts the team had on the oncoming change to compare the views before and after the start of the change process. We also wanted to use the pilot study results to help form relevant questions for the case study. We then executed the case study to examine and help understand how the team uses the tools and how they are experienced and affect the daily work.

4.2 Studies

4.2.1 Pilot study

We interviewed the team through surveys to build a basic understanding of the working method and the relationship to safety-related issues. We wanted to gain basic knowledge of what ideas and thoughts the team had on the oncoming change to compare the views before and after the start of the change process. Further, we examined the security directives and interviewed the IT security group to determine what the security work looks like and distinguish and concretise possible changes and proposals for possible additions.

We aimed to study who was part of the team and their responsibilities. We sought to answer who was responsible for security issues, how they work today, and how they wish to work in the future. Further, we asked about the key directives and how the team viewed them.

4.2.2 Case study

We examined the application of automated security tool, Fortify, to understand how the team uses the tools and how they are experienced and affect the daily work. Moreover, how they are linked to requirements management. We asked if there is room to enter more automatic code tests, in which contexts are they practical, and how are the results used. We also investigated how developers experienced these tools and the processes involved. We performed our case study over a so-called sprint spanning three weeks.

We aimed to study the introduction of new security tools and how mature the working method was with these tools. Further, we studied whether the developers were involved as stakeholders or not. Additionally, we questioned how they viewed the tools and whether they enjoyed them. Lastly, we asked what they thought worked well and less well during the implementation.

4.3 Study implementation

Surveys were conducted with people who experienced implementing new security methods in the daily workflow. Structured surveys were chosen for an in-depth investigation. The questions in structured surveys are usually short and easy to understand (Silverman, 2017).

4.3.1 Survey questions

The literature study is the basis for the surveys, where questions are built from the drawn theoretical framework. The survey questions can be found in the form of a survey guide in the appendix. The survey guide for the pilot study can be found in appendix A.1 and the guide for the case study in appendix A.2. The framework of the questions is presented in Table 1.

Questions	Concepts	Significance
1-6	Change management	Investigate how the management worked towards the transition and how the team viewed the management's involvement.
7-9	Adoption of change	Examine how the team has adopted the change process and how they regard the shared vision.
10-11	Organisational inertia	Look into how the transition has affected the team through sociotechnical means and examine what processes have been implemented to deal with such challenges.

Table 1: Framework for the survey questions and the studied concepts.

4.3.2 Survey respondents

The six respondents in the survey study were all parties in the development team and are listed in Table 2.

Respondent	Role	Pilot survey	Case survey
A	Software engineer	21/2-2022	30/3-2022
B	ICT engineer	23/2-2022	30/3-2022
C	Software engineer	22/2-2022	30/3-2022
D	Senior software engineer	24/2-2022	30/3-2022
E	Software engineer	24/2-2022	30/3-2022
F	ICT consultant	23/2-2022	4/4-2022

Table 2: Respondents in the survey and their date of response.

4.3.3 Survey conduct

The surveys were conducted via Microsoft Forms and sent out via email. The surveys were split up into sections not to be perceived as too heavy and extensive for the respondents. The respondents were informed that the pilot study survey should take approximately 20–30 minutes and the case study 30–40 minutes. The deadline was three days after the respondents were handed the survey.

4.4 Data analysis

Primary data were collected from formally structured surveys, and the data were analysed using the Gioia method. This method structures data from survey studies by grouping the responses from the respondents using first-order concepts and linking them to second-order themes, and then translating them into an aggregate dimension (Gioia, Corley, and Hamilton, 2012).

The Gioia method is an approach for structuring material collected from qualitative research. The methodology was used by structuring the survey data and developing first-order concepts from the survey material, which were then linked to second-order themes. A simplified model in Figure 4 illustrates the method. The figure exemplifies some keywords sorted under second-order themes related to the aggregate dimensions. The aggregate dimensions are synthesised from the theoretical framework presented in chapter 3.

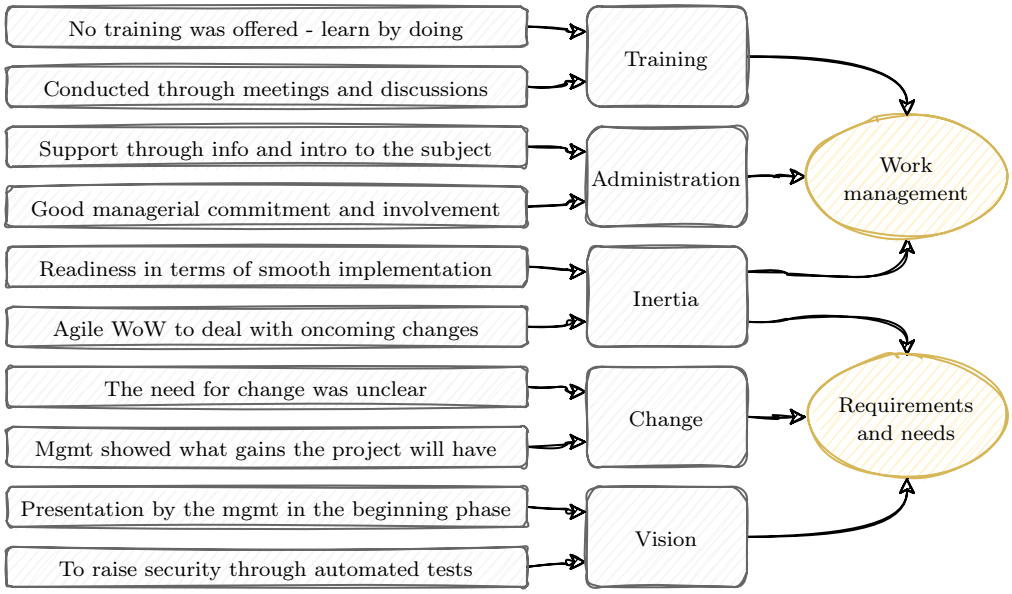

Figure 4: Illustrates the study's Gioia methodology for data analysis, demonstrating the first-order concepts, second-order themes and its aggregate dimensions.

4.5 Method criticism

A common criticism of qualitative data collection is that the collector's perception risks staining the data. The data collector can influence the data by, among other things, forming the survey questions based on expected answers

(Bryman and Bell, 2011). With this in mind, the survey questions have been designed to give the survey respondents great freedom in their responses.

Data collection through surveys is characterised by the respondent's subjective perception, which is reinforced in qualitative surveys because the number of respondents is relatively tiny (Allwright and Bailey, 1991). In this study, data collection was based on six respondents. The benefit of the method choice is that it allows for more in-depth and developed answers, which may be set against the image not having the same breadth as quantitative studies.

Results

This chapter is divided into two parts. The first part, pilot study*, describes the results of the pilot study. The second part,* case study*, describes the results of the case study.*

5.1 Pilot study

5.1.1 Change

Regarding improvements, one respondent thought that the tools should visualise the data in a dashboard with easy-to-understand metrics. Other than this, the respondent also wanted to point out the importance of automation. Moreover, another respondent thought that security tools should provide clear reports to work effectively. Reports that deliver warnings about vulnerabilities from regular scans on the codebase. Such reports can be achieved by implementing security testing and modelling threats and risks in each software project. A security test plan should be included in the design phase of a new feature in the project to work with this. Lastly, one respondent emphasised that the dependency on external parties should lessen.

Most respondents agreed that security is essential and critical in software development, particularly during development and maintenance. Moreover, most respondents were also optimistic about implementing new security measures and tools. Furthermore, a majority of the respondents encouraged the implementation into the pipeline. One respondent was positive towards implementation into the IDE, and one was negative. Lastly, one respondent was negative about implementing any security tool and did not want to implement more friction to the workflow.

5.1.2 Administration

According to the respondents, security has previously been handled by external parties. Topics regarding security have been administered by a connection to the internal identity manager at Ericsson, who had access to perform different security actions. Management has previously tried to implement new tools to give more developers access to security measures. One such tool, Sonarqube, an open-source platform, was tested locally and connected to the pipeline but was not implemented further. This security approach requires many setups to ensure that the reports give relevant and productive results.

In addition, one respondent thought that clear communication is key to successful working with security in a sustainable manner. The superiors should provide directions or information on where to find instructions on implementing and working with the security tools. One respondent answered similarly and conveyed the importance of working with the agile workflow to implement the new methods sustainably. Another respondent highlighted the importance of a shared understanding of why a new process is introduced in the workflow. Moreover, good motivation and a clear scope of how and when the method should be implemented are essential components of a successful change process.

5.1.3 Vision

The overall opinion of the development team regarding the relation to security within software development is that there should exist good code writing, well-implemented authentication, and authorisation tools from the start. Testing and deploying the software should also be in place to identify intended users, stakeholders, designs, and processes. Controlling who can be responsible for owning the security product is a good idea. As well as using the proper setup for APIs to ensure secured access and conduct daily reviews for the failure report. Trust is also a factor lifted by one responder regarding working with security measures.

Altogether, the respondents said the oncoming routines are to implement scanning code in the pipeline and performing daily code checks that result in immediate alerts. They have received instructions from management that they should use static application testing and mitigation of vulnerabilities before each commit. Moreover, management aims to assign reliable resources to the development team, provide customised training, and ensure that everyone has a clear goal and obvious procedure to follow. An iterative evaluation is an excellent idea to add, which can also enable an frictionless implementation

of scanning of Free and Open-Source Software (FOSS) into their pipelines.

Concerning how security work is integrated today, one respondent said that security is accomplished by doing tests based on the User Interface (UI). The tests are run at night, and the results are discussed in the morning standup. In addition, another respondent said that more work is needed and that the performance of the build cycle is essential and should not be overly affected by security tools.

5.1.4 Training

One respondent thought that the biggest challenge is to specify the security standards; there are many standards. One cannot follow all but should decide based on the product at hand and the business specification for the product. Then comes the proper setup of the team to include the right competence or maybe include a designated security responsible resource. Another lifted the decision making of the right tool to use in supporting the fulfilment of security measures in the way of working because it needs to be effective and time-efficient. A third said that automation and knowledge to interpret different measurements, metrics, and results. Another respondent lifted the issue of communication and understanding across the whole team. One respondent highlighted the importance of the experience of the team members. Lastly, one respondent expressed that education should be included to utilise the new method(s) effectively.

The perception of the existing security directives spans from reasonable to a lot. There is room for improvement and knowledge, for example, the CIA triad would benefit the whole team, one respondent wrote. Overall, keeping security measures on top all the time is key to success.

5.1.5 Inertia

Regarding previous challenges with security, all of the respondents expressed that there had not been any significant challenges. The reason is due to a more ad hoc approach in the past. Although, confusion about who owns the security certificates has led to them expiring without any warnings being upheld, for example SSL certificates.

Furthermore, one respondent thought that information and data from security tools should be easily accessible to the users. Lastly, one respondent deemed that security should be implemented in the early stages of a software development project but is often implemented in a late stage when security

breaches have already become a problem.

The overall perception is that today the security work spans a longer time, which has created a gap within the development cycle. There are quarterly external scans and yearly reviews of the prominent security aspects, but daily code hygiene is executed, and servers' health checks up are reviewed. No code goes into the main branch before a pull request review session. So, there are checked more frequently than others. In addition, one respondent said that the security work in a day-to-day activity included authentication, authorisation, data validation, encryption, protecting sensitive data, analysing the impact of external component integration, and logging in Azure.

Concerning the way of work and the daily workflow, one respondent said that the implementation phase could inhibit the daily workflow. However, once the new security tools are fully implemented, they will work well. Some respondents were also worried that the scanning results of the tools can be hard to read and understand and may thus inhibit the workflow. Therefore, there should be clear instructions on what warnings should be mitigated and what warnings could be put on hold to work against these adverse effects. One respondent emphasised that security work should be well implemented in the agile work process to work effectively. Another respondent noted that security work would lower the workflow somewhat, but the gains in more secure code will overweigh potential loss of workflow. One respondent also stated that development could be inhibited if security directives limit their use of vulnerable internal data.

5.2 Case study

5.2.1 Change

The respondents were not entirely sure about the need for change and thus were not aware of the common goal. However, some respondents had views of the common goal and expressed this in several ways. One respondent stated that the message was that security measures are becoming highly important, and action to implement is needed from all Ericsson software development teams. Another respondent continued on this line and added that the goal was to raise security through automated tests. Further, one respondent wrote that "[the] need of change was presented by showing the gains that we as a project will have and what we will gain at the organisation level". Lastly, one respondent added that the object was a natural way of achieving an evolved form of working with security.

5.2.2 Administration

The respondents were satisfied with the management and its support through the transition. One respondent stated that the team was supported through information and introduction to the security subject. Moreover, another respondent expressed that management supported through verbal support, in terms of encouragement in security tools. One respondent noted that there were learning sessions in place, and another respondent worded that management offered a high-level introduction to the security tools. Lastly, a respondent expressed that management supported the team by giving them freedom in the implementation phase.

Regarding management commitment and involvement at the line level, one respondent thought it was satisfying, while another respondent only deemed it enough. Moreover, one responded that this commitment allowed the team to raise questions and ask for support when needed. However, the respondents' answers were not conformed, and one respondent expressed that all commitments from higher-level were communicated through one team member.

5.2.3 Vision

One respondent expressed that the change vision was unclear. However, most respondents did not indicate this, and instead, one of the respondents noted that the management presented the change project in the beginning phase. According to one respondent, the change project was automated security tests of the code base, which was a mandatory change. Further, another respondent continued and reported that it was highlighted that these implementations were to be required in the future and that a trial period was the first step. Lastly, one respondent formulated that management presented a comparison and choice over other competitive tools. The specific tool was then implemented with assistance from an external team.

Regarding participation in the change project, one respondent wrote, "I was briefed during the implementation phase but not part of the planning activities". Whilst another respondent noted that one team member was responsible for the implementation. Moreover, one respondent expressed that external support was always near when a problem occurred. Another respondent answered similar to this and stated that there were scheduled meetings with support teams for questions and answers.

5.2.4 Training

According to the team members, no training was offered; however, one respondent wrote that the progress was instead "learn by doing". However, one respondent expressed that training has been conducted through meetings and discussions with the team. Moreover, according to one respondent, there was no organisational commitment at the line level to ensuring proper recourses. Another respondent expressed that one team member handled all external communication, regarding education.

5.2.5 Inertia

Communication concerning the progress of the change project was regarded as suitable by several team members. The communication was mainly done via discussions within the group, which kept motivation up. Moreover, one respondent documented that the communication regarding sustaining engagement and motivation has mainly been done via the security product team.

Inertia was regarded as suitable by one respondent in terms of smooth implementation. Another respondent also expressed smoothness of the implementation, and reported that this was due to the agile ways of work to deal with oncoming changes. However, most respondents expressed that there was no inertia for the resulting changes from the implementation. Lastly, one respondent wrote that "dependence on an external team for a period and through internal discussions on dealing with changes", which means that there was an dependence on an external team during the handover phase.

5.3 Summary

Concept	Pilot study	Case study
Change	Clear and optimistic view of a change process regarding security improvement added tools	Expressed uncertainty about the purpose of the change process
Administration	Expectations that the management provides clear communications and distinct directions regarding change processes, together with a well-motivated scope	Satisfaction with managerial support throughout the transition of the change process
Vision	Awareness of the oncoming change regarding implementing security tools and ideas on how the change process could be mitigated	Different levels of perception of the change vision
Training	Highlighting the importance of the right competence for the respective tasks	Contradicting opinions of training and education
Inertia	Less knowledge and awareness of inertia during previous change processes, although signs of inertia is mentioned and describe	The term inertia was regarded as suitable and not an attribute to any friction during the change process

Table 3: Summary of the results according to the theoretical concepts.

Analysis

This chapter is divided into two parts. The first part, pilot study*, presents an analysis of the pilot study. The second part,* case study*, presents an analysis of the case study.*

6.1 Pilot study

6.1.1 Work management

Administration

Various things were expressed about how the administration at Ericsson handled and affected the view of change and how the developers received it. As Källberg (2013) mentions, the view of change depends on humans with different backgrounds, experiences, and points of view. Measey (2015) states that creating trust, work morale, and good communication is, above all, to be prioritised.

From the pilot study, one can see that the respondents had ideas and views of how things should be handled by the administration when a change process is initiated. One respondent listed communication as the key to success, another wanted clear instructions, and a third highlighted the importance of agile workflow and shared understanding of why the change is necessary. Moreover, good motivation and a clear goal on how and when was something that could be distinguished as very important among the survey's respondents.

One can, when reading this, see that the statements by both Källberg (2013) and Measey (2015) can be found in the respondent's answers, unintentionally perhaps; nonetheless, the statements are there. A conclusion that can be

drawn is that the respondents, in this study, are experienced. They know what to expect from the administration during a change process and how a change is implemented in the best way. In other words, the condition for an excellent accomplished change process seems to be in place.

Training

According to the pilot study, precise instructions and education would facilitate the implementation of new security tools. The respondents requested the right competence for the implementation through education and a designated security responsible resource. The respondents highlighted the importance of choosing the right security tool and indicated that they wanted to participate in the decision-making procedure. Knowledge of the team was lifted along with integrated education to utilise new methods effectively.

From the respondents, one can see that parts of Kotter (2012) eight-steps model and ADKAR (Hiatt, 2006) was indirectly referred to, and this strengthens the case that the group of respondents are experienced and that they are well adjusted to a changing process.

Inertia

Besson and Rowe (2012) refer to inertia in organisations as an essence of the act of organising. The authors argue that the fortification of routines and patterns is essential to becoming organised. The overall perception regarding the organisational inertia, the respondents' security work perspective, is that it spans a long time, leading to a sense of a time gap within the development cycle, which can be a result of a more ad hoc approach which has led to some confusion and has resulted in not knowing who is responsible for the security certificates, such as SSL certificates. The respondents continue to describe the problems with a late security approach by saying that security has already become a problem when security tools have been implemented in a security breach. The lack of fortified routines and patterns leads to organisational inertia, which coheres with Besson and Rowe (2012).

"The impact of those characteristics of a sociotechnical system that affect the effectiveness and efficiency of a specified change process in the system" is the definition of inertia, according to Lind (2017). A *sociotechnical system* is a system that can be seen as a combination of two interdependent subsystems, one social and one technical (H. K. Klein and Myers, 1999). These two statements from Lind (2017) and H. K. Klein and Myers (1999) correspond with Besson and Rowe (2012) and fortify the organizational inertia at Ericsson.

To curb the organisational inertia, administration and management need to have, as Besson and Rowe (2012) say, focus on routines and pattern, which according to the respondents, has been overlooked.

6.1.2 Requirements and needs

Change

Besson and Rowe (2012) argues that when introducing changes to the employees' work, they need to believe that the change is meaningful and impacts the organisation's success. The respondents seem to look at the incoming change positively and deem that the shift is a necessary step forward. Further, the employees emphasise that the change process; implementation of new tools; should be done with a clear and consistent scope. The employees should understand the change process and its components, the new implements, and how to use them properly. This is, in other words, a clear demonstration of using the information system, which Besson and Rowe (2012) points out to be an essential factor in a well-executed change process.

Vision

The development team had a clear and consistent vision which boiled down to an urgent need to develop secure code by implementing new security routines. According to Kotter (2012), establishing a sense of urgency is the foundation of an organisational transformation. Further, Kotter (2012) explains that creating a guiding coalition and developing a vision and strategy are also essential elements in the change process. Furthermore, communicating the change vision is vital for the transformation, according to Kotter (2012). Ziemba and Obłąk (2015) also theorise along these lines and argue that recognising the change means that the need for change has to be established and promotes a positive approach to change. As the team have been informed of the change process and as they were primarily optimistic about change and have had communication, one could argue that the criteria provided by Kotter (2012) and Ziemba and Obłąk (2015) were fulfilled.

Moreover, a shared vision should be strongly advocated across the organisation, Ziemba and Obłąk (2015) explain. The team shared a clear scope of what methods and procedures should and would be put into place, which indicates that the process was moving in order with the eight-step plan provided by Kotter (2012) and the twelve factors provided by Ziemba and Obłąk (2015). Ziemba and Obłąk (2015) explains that employee involvement indicates that the employee needs to believe that the change is essential and

impacts the organisation's success.

Inertia

Kotter (2012) eight-step model mentions a guiding coalition, communication and empowering the employees as requirements and needs for a successful change process. These steps, from Kotter (2012), go in line with Besson and Rowe (2012), which refers to inertia in organisations as an essence of the act of organising. To become organised, the fortification of routines and patterns is essential (Besson and Rowe, 2012). This means, in this context, that fortifications of Kotter (2012) requirements are crucial to success with a change process. Moreover, through this, to managed to curb the organisational inertia.

To connect with the response from the pilot study, there was confusion regarding ownership of the SSL certificates, which can be referred to as a lack of guiding coalition. The current security work could create a time gap, leading to lesser communication within the development cycle. The future security implementation could inhibit the workflow, leading to the employees feeling less empowered. That can lead to a less organised organisation and therefore increase organisational inertia. From this, one can see how the essence of the act of organisation Besson and Rowe (2012) intertwined with Kotter (2012) model and therefore tells us what can be done to meet the requirements and needs for a successful change process and at the same time become organised and curb the organisational inertia.

6.2 Case study

6.2.1 Work management

Administration

The general opinion regarding the administration's role in implementing new security tools was positive. The respondents were satisfied with the support from management, communication, encouragement and freedom under their responsibility. Questions could be raised, and support was always given. A good combination of factors coheres with Källberg (2013) and Measey (2015), who state that the view of change depends on humans with different experiences and that creating trust, work morale and good communication is above all to be prioritised. One can also find evidence that some parts of the eight-step process by Kotter (2012) were fulfilled during the implementation

phase. Kotter (2012) states that establishing relevance (1), guiding coalition (2), vision strategy (3), having good communication (4), and empowering employees (5) are vital parts of success in a changing process (Kotter, 2012).

In addition to Kotter (2012), one can see traces of the ADKAR-model (Hiatt, 2006), which go hand in hand with the similarities with the models. Awareness (A), desired (D), and so forth are easy to detect and tell us that Ericsson's administration has been doing a good job when implementing this new security tool (Hiatt, 2006).

Training

The respondents declared that relevant training was not offered; learning by doing was the approach instead. An approach that contradicts both Kotter (2012) and Hiatt (2006) argues in terms of not giving the employees a guiding coalition (Kotter, 2012), empowering employees for broad-based action (Kotter, 2012) or knowledge of how to change (Hiatt, 2006). A respondent lifted that some educational content had been conducted through meetings and discussions, classified as knowledge of how to change (Hiatt, 2006).

Kotter (2012) seventh- and eighth step tells us that sustaining acceleration by consolidating gains and producing more change, and anchoring new approaches in the culture. Things that require that the organisational commitment, at the line level, ensures proper recourses be in hand. According to the respondents, this has not been the case.

Inertia

The respondents expressed that there has not been any inertia for the resulting changes from the implementation. Communication concerning the progress has been suitable and kept the motivation up. One respondent mentioned the smoothness of the changing process and highlighted the agile ways of work as the main reason. As Appelo (2012) writes, "change management is a vital part of agile practices; it allows organisations themselves to adapt" one can say that agile practices can be a vital part of change management.

6.2.2 Requirements and needs

Change

The team mostly agreed upon the common end goal: to raise security through more robust and secure code through tools with automated tests. However, the response regarding the specific need for change and how the end goal

would be achieved was not wholly consistent. Ziemba and Obłąk (2015) argues that the need for change has to be established and, through this, promotes a positive approach to change. Further, the authors contend that a shared vision should be strongly advocated across the organisation (Ziemba and Obłąk, 2015).

Vision

The pilot study shows that the development team had a clear and consistent vision, which boiled down to an urgent need to develop secure code by implementing new security routines. When asked in the case study, different opinions regarding the vision during the implementation phase were presented. One respondent expressed unclarity about the change process vision. However, most respondents disagreed with this statement and instead expressed that the management did present the change project at the beginning of the process. In addition, one of the respondents noted that management presented a comparison and choice over other competitive tools and explained why this specific security tool had been chosen for this changing process.

As the changing process could move in order with the eight-step plan by Kotter (2012) and the twelve factors by Ziemba and Obłąk (2015), according to the pilot study, one can argue that during the case study, the change process did not move in this direction. It can be argued that an inconsistent perception of the vision existed. Kotter (2012), Ziemba and Obłąk (2015) and Measey (2015) all point out the essence of good communication is necessary for a successful change process. Therefore, less good communication can answer why the inconsistent perception regarding the vision existed.

Inertia

In contrast with the pilot study, the response from the case study stated that there was no inertia for the resulting changes from the security tool implementation. There was, on the contrary, a smoothness due to the agile ways of work. The respondents indicated that the communication concerning progress was suitable, which kept motivation up, which indicates that the requirements for a successful change process were met during the implementation phase. Many parts of the eight-step model by Kotter (2012) is fulfilled, such as a guiding coalition, communication and empowering of the employees, one can say that it has been a successful change process. In addition, it also fulfils what Besson and Rowe (2012) and Lind (2017) argued about organisational inertia. Therefore, one can say that the organisational inertia has been or is in balance for the UDB team at Ericsson Group IT.

Discussion

This thesis's purpose is to apply the concept of inertia to agile work practices to examine how small-scale projects are affected when new security methods and tools are introduced and implemented in the agile workflow. The thesis research question is how the concept of inertia provide a theoretical framework and which factors describes the effects on an agile software development team, when new secuirty methods and tools are included in the project requirements. The results from the study conclude that communication is a vital part of a successful change process. That adaptability is a crucial mindset to have if one is to succeed in a change process. Further, this chapter introduces the reader to the discussion, which is divided into two parts. A review of the two dimensions of work management *and* requirements and needs *is presented and discussed.*

7.1 Work management

7.1.1 Administration and training

Managing change is a process and gives the impression that a successful transformation is a matter of execution and avoiding pitfalls (Snowden, 2015, 2016). One can deduce that both the respondent and the management shared the same perception regarding how a change process should be accomplished. In the pilot study, it was evident that the respondents had ideas of what to expect from management during a change process. The case study shows that the views from the pilot study became actions and that the communication regarding the change process had worked well. The conclusion is that the input from the administration was satisfactory, and the respondents were satisfied.

When asked about communicating training and education, the opinion changed

abruptly. Respondents conveyed they did not get the training required but learned by doing instead. Henceforth, the respondents expressed that organisational commitment and proper recourses did not exist. Reading this gives us the impression that there is an ambiguity with the perceived change process regarding communication. On the one hand, the respondents claim good awareness and communication from the administration.

On the other hand, when asked if management offered education, the respondents upheld a less positive attitude. This ambiguity is not what previous research recommends for a successful change process. Previous research states that creating trust (Measey, 2015), creating a guiding coalition (Kotter, 2012), and a desire to support and participate in the transformation (Hiatt, 2006) are essential for a successful change process.

These ambiguous opinions about communication are interesting in terms of the perception of information. An explanation is that the case study survey did not map the correct perception. The questions may have been formulated too vaguely, thus giving the respondents too much room for various interpretations. However, this explanation seems unlikely since the questions have been used successfully in previous research by, for example, (Ziemba and Obłąk, 2015). They are known to map the perception about communication regarding performance and education.

"We value responding to change over following a plan" is key to adaptive change management, according to Beck et al. (2001), and can be the answer to why the perception of the case study's communication conveys such an ambiguous result. Snowden (2015, 2016) states that a change process is not only a matter of execution and avoiding pitfalls, but more a path to find the adjacent possible to the desired goal (Kauffman, 2003). Stating this is relevant when trying to understand the evolution of communication. The respondents had a clear view of what to expect from the administration. The administration knew what was expected and made a plan, but on the way, the plan became more important compared to the response to the change.

Therefore, having this in mind when orchestrating communication and how it interprets by the recipient, it is essential to remember that where one starts might not be where one is heading. During the process, a lot can change. Do not reduce the importance of education and training; this increases participation and makes the group more receptive to change. Further, providing appropiate communication customs is one of the success factors for change processes.

7.1.2 Inertia

The respondents expressed that there has not been any inertia for the resulting changes from the implementation. The respondents express gratitude and state that motivation has been kept up and smooth throughout the change process – a tremendous result in creating the impression that the change process has run smoothly and without hazards. One can question this when going through the rest of the case study results. A wide range of statements coheres with less inertia, but many statements contradict.

A general opinion regarding the implementation was positive: the support from management was satisfactory, communication was good, and the respondents had freedom under responsibility. Nevertheless, education was not offered, a not wholly consistent end goal was presented, and an unclear vision of the change process was given. With this said, one can declare that there are many contradictions regarding the implementation. One on hand understandable that the respondents experience less or no inertia. At the same time, not being given any education or a transparent change process vision, one can argue for the possibility that there has been some inertia. Therefore, the analyses regarding inertia may confuse.

One explanation for this is that the word inertia is too direct to use in one analysis model, as inertia is a term that has been applied in the present study. A possible explanation is a favourable opinion when asking general questions regarding the implementation work. However, it may be easier to point out the negative aspects of direct questions about how the education has been and the vision.

The results in the present indicate that Ericsson Group IT has used elements of both the eight step-model from Kotter (2012) and the ADKAR-model by Hiatt (2006). We believe that using the PDCA model (Koiesar, 1994) can be more applicable in this case. PDCA is made for small scale projects, and there is a reevaluation phase included in the model. Using PDCA can be a way to avoid organisational inertia.

7.2 Requirements and needs

7.2.1 Change and vision

A change affects people, and people are by default pessimistic about change (Källberg, 2013). Kotter (2012) argues that creating a guiding coalition and developing a visual strategy is essential for success with change. These two

factors, from Källberg (2013) Kotter (2012), will mitigate the change process and make people more accessible to the change process.

According to the respondents, there was a shared end goal for the change process. Nevertheless, the team's answers were not unanimous and did not agree with common management principles on how this goal should be achieved. The team's responses were more consistent in the pilot study than in the case study, which could be because the implementation process changed somewhat from idea to practice. We conclude this because the pilot study indicates one clear change path, but the nature of this path was not the same in the case study. Another explanation for this could be the team's expectations that were not met in terms of the change process approach. Unattended expectations can depend on either management's lack of information before the change process or unrealistic expectations from the team. Moreover, when examining the respondents' answers regarding the change process, the answers were consistent regarding having a shared vision of the change process during the pilot study.

Furthermore, the answers altered somewhat in the case study and were no longer entirely aligned with management's shared vision. The case study analysis indicated an inconsistent perception, perhaps due to inadequate communication between management and the development team. It could also be due to the difference between theory and practice. There could be several reasons for this difference, again due to an unrealistic vision from the management or unrealistic expectations from the team members. Nonetheless, communication was deemed adequate or satisfactory in the respondents' answers regarding communication in other study questions. So perhaps the lack of the management's communication can be disregarded. Looking for other reasons for the inconsistency between the two studies, one can ask if the study questions were designed to be too general, resulting in sprawling answers. However, as previously remarked, we judge this as less likely due to the questions established by previous research by Ziemba and Obłąk (2015).

With this said, communication was deemed adequate, and the case study questions were established by previous research. One can only conclude that communication was the weaker party in the matter. We say this with reference to an unclear change process vision. One can consider that having an unclear change process vision indicates that appropiate communication customs has not been provided. Moreover, as stated before, appropiate communication customs is one of the success factors for a change process.

7.2.2 Inertia

The combined analyses of the pilot study and the case study indicate that assigning the term *inertia* to both dimensions seems problematic. Analysing the theme in both dimensions was fruitful, but discussing the relevance in a differentiated way was not. One possible reason may be that inertia can be easy to detect but hard to distinguish, which causes ambivalence.

Keeping what Beck et al. (2001) is stating in mind, "we value responding to change over following a plan" can help us understand the ambiguous answers regarding inertia. Nearly all theoretical material regarding a successful change process states that creating a plan and upholding good communication is critical (Hiatt, 2006; Kotter, 2012; Measey, 2015; Snowden, 2015). However, as Jun et al. (2010) states, "a team never sticks to a fixed plan, and they adapt to the upcoming changes as the project progresses". Combining what the authors are conveying concludes, according to us, with flexibility and awareness. It is crucial to have a clear goal, follow some established change process methods, be humble toward the possibility that a plan can be overlooked and always be prepared to adapt (Hiatt, 2006; Kotter, 2012). One can deduce that being adaptable to the change process is essential.

7.3 Synthesising the concept of inertia

The result shows that inertia has not been considered or handled, which may have caused some obstacles in the previous and current change process, according to the pilot study, see Table 3. Despite this, inertia could be distinguished from the case study and therefore establish the relevance to highlight this. Furthermore, from the case study, it could be deduced that the respondents considered inertia as an appropriate way to describe certain parts but that no inertia had arisen, see Table 3.

However, the non-existence of inertia can be contradicted, according to us, due to the ambiguous answers given regarding education, training, communication, vision and the change process itself. There was some confusion regarding whether education and training have been offered, what had been said and how the change process procedure looked like.

Further, the results showed that the knowledge about the concept of inertia does not seem to be widely accepted. This led to the proposals we put forward to Ericsson Group IT, which introduced a specific change methodology (PDCA) to deal with adaptability to the change process, appropriate communication customs, and offered adequate education and training to everyone

involved in the change process. These are three vital factors for successful change processes that have been synteshised from the themes in the discussion.

Conclusion

This thesis examines how a development team is affected when new security aspects are mixed into the project requirements. We can conclude from the discussion three general factors: the importance of education and training, appropriate communication customs for the situation, and adaptability to the change process.

Training and education are essential factors in facilitating a successful change process. If these factors are not met, the team can experience a sense of uncertainty about why the change is necessary in the first place. Without the right tools to execute the new workflow, the team will feel like they do not know what they are doing. If they do not understand why they are doing what they have been told, the team will not understand the purpose of the change, and therefore the change process can be hard to achieve.

A good environment for communication within the team and good communication from the management is also an essential factor in yielding a successful change process. As a manager of a change process, it is crucial to know why a change is to be done and have the right tools to communicate this to the development team. Listening to the development team about what they experience as essential elements is equally important. Consolidate and make sure that everyone in the change process is aware of what to expect from everyone. Uncertainty can be avoided if there is clarity on whom to contact when in need.

Lastly, adaptability to the change process can enable a change process to go from sufficient to great. Planning and structuring a change process is essential but knowing that a plan is never executed as designed is even more critical. Looking at a change process as a climbing wall can be a good metaphor. One can ideate where the finish line is, but the path is not crystal clear. To get there, one needs to adapt and excel in the strive to succeed.

Being aware of these three factors creates awareness of what Lind (2017) defines as organisational inertia: awareness of what affects the effectiveness and efficiency of a specified change process. In other words, being aware of the importance of education and training, appropriate communication, and adaptability to the change process creates awareness that inertia is an ever-existing factor that needs to be considered and dealt with to succeed in a change process.

Recommendations to Ericsson is that in a future change process, have these three factors in mind and fortify these in the agile workflow. As an agile work process is suitable when initiating a change process, we do not see any incentives to change this. However, we have deduced, in this specific study, that there is no expressed change process model, for example, the eight-step model by Kotter (2012), the ADKAR-model by Hiatt (2006), or the PDCA-model by Koiesar (1994). We believe that the PDCA model could be beneficial due to the PDCA being made for small scale projects, and there is a reevaluation phase included in the model. As such, usage of the PDCA model can be a way to avoid organisational inertia.

This thesis research question was how the concept of inertia could provide a theoretical lens through which the change process can be understood better, that is which factors describe the effects on an agile software development team when new security tools and methods are included in the project requirements.

We can establish that the concept of inertia, as a theoretical lens, has given us insight into what can mitigate a change process and what possible effects an agile software development team experiences when new security tools and methods are included in the project requirements.

This insight involves the following three factors: personnel training and education, appropriate communication, and adaptability to the change process.

As these factors indicate, this thesis has successfully addressed its research question and, in doing so, contributed to research on how the concept of inertia affects small-scale projects when new security tools and methods are introduced and implemented in the agile workflow.

8.0.1 Future work

As this has been a study on a small development team, we believe that to draw more extensive conclusions and find more factors affecting an agile software development team experience when new security tools and methods

are included in the project requirements. A more comprehensive case study must be conducted. We see the possibility of adding to the definition by Lind (2017) of organisational inertia to make it more specific toward smaller changing processes in an agile work environment. Adding to the definition by Lind (2017) could improve understanding of the social aspects of technical change processes, in other words, the socio in sociotechnical. Furthermore, we consider that more research is needed regarding implementing technical tools for teams in software development.

Bibliography

Abasi, F. (2021). *SAST, SCA, DAST, IAST, RASP: What they are and how you can automate application security.* https://forwardsecurity.com/2021/07/13/automatingappsec/ (collected 25.03.2022)

Aladwani, A. M. (2001). Change management strategies for successful ERP implementation. *Business Process Management Journal*, *7*, 266–75.

Allwright, D., & Bailey, K. M. (1991). *Focus on the language classroom: An introduction to classroom research for language teachers.* Cambridge University Press.

Alphabet Inc. (2021). *Accelerate state of DevOps 2021.* https://services.google.com/fh/files/misc/state-of-devops-2021.pdf (collected 19.04.2022)

Appelo, J. (2011). *Management 3.0: Leading agile developers, developing agile leaders: The addison-wesley signature series.* Pearson education.

Appelo, J. (2012). *How to change the world: Change management 3.0.* Jojo Ventures BV.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Higgsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for agile software development.* (collected 11.02.2022).

Besson, P., & Rowe, F. (2012). Strategizing information systems-enabled organizational transformation: A transdisciplinary review and new directions. *Journal of Strategic Information Systems*, *16*, 3–18.

Bryman, A., & Bell, E. (2011). *Företagsekonomiska forskningsmetoder* (2:1). Liber AB.

Bullen, C. V., & Rockart, J. F. (1981). A primer on critical success factors. *CISR*, *69*, 1220–81.

Chrusciel, D., & Field, D. W. (2006). Success factors in dealing with significant change in an organization. *Business Process Management Journal*, *12*, 503–516.

Cocks, G. (2014). Optimising pathways for an organisational change management programme. *The TQM Journal*, *26*, 88–97.

Davenport, T. H., Harris, J. G., & Cantrell, S. (2004). Enterprise systems and ongoing process change. *Business Process Management Journal*, *10*, 16–26.

Douven, I. (2017). *Abduction.* https://plato.stanford.edu/archives/sum2017/entries/abduction/ (collected: 16.03.2022)

Geels, F. W. (2005). The dynamics of transitions in socio-technical systems: A multi-level analysis of the transition pathway from horse-drawn carriages to automobiles. *Technology Analysis & Strategic Management*, *17*, 445–476.

Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2012). Seeking qualitative rigor in inductive research: Notes on the Gioia methodology. *Organizational research methods*, *16*(1).

Graetz, F. (2000). Strategic change leadership. *Management Decision*, *38*, 550–64.

Griffith, T. L., & Dougherty, D. J. (2001). Beyond socio-technical systems: Introduction to the special issue. *Journal of Engineering and Technology Management*, *18*, 207–218.

Guimaraes, T., Igbaria, M., & Lu, M.-t. (1992). The determinants of DSS success: An integrated model. *Decision Sciences*, *23*(2), 409–29.

Hiatt, J. (2006). *ADKAR: A model for change in business, government, and our community* (1st ed.). Prosci Learning Center Publications.

Hotek, D. R., & White, M. R. (1999). An overview of performance technology. *The Journal of Technology Studies*, *25*, 43–50.

IBM Cloud Education. (2020). *What is DevSecOps?* https://www.ibm.com/se-en/cloud/learn/devsecops (collected 11.02.2022)

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, *51*, 915–929.

Jun, L., Qiuzhen, W., & Lin, G. (2010). *Application of agile requirement engineering in modest-sized information systems development in 2010 second world congress on software engineering.* https://doi.org/10.1109/WCSE.2010.105 (collected: 11.04.2022)

Källberg, N. (2013). *Förändringsprocesser i sjukvården: En studie av aktörer på en röntgenavdelning.* Stockholm: Handelshögskolan.

Kauffman, S. (2003). Molecular autonomous agents. *The Royal Society*, *15*, 1089–99.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, *23*, 67.

Klein, L. (2014). What do we actually mean by "sociotechnical"? on values, boundaries and the problems of language. *Applied Ergonomics*, *45*, 137–42.

Koiesar, P. J. (1994). What deming told the japanese in 1950. *Quality Management Journal*, *2*, 9–24.

Kotter, J. P. (2012). *Leading change*. Harvard Business Review Press.

Lambert, A. (2022). Interview [Regarding "Imagine possible"] 19.04.2022.

Leidecker, J. K., & Bruno, A. V. (1984). Identifying and using critical success factors. *Long Range Planning*, *17*, 23–32.

Lind, T. (2017). *Inertia in sociotechnical systems: On IT-related change processes in organisations*. Uppsala: Informationsinstitutionen.

Measey, P. (2015). *Agile foundations: Principles, practices and frameworks*. BCS Lerning & Development Ltd.

Micro Focus. (2022a). *What is Dynamic Application Security Testing (DAST)?* https://www.microfocus.com/en-us/what-is/dast (collected 11.02.2022)

Micro Focus. (2022b). *What is Open Source Security?* https://www.microfocus.com/en-us/what-is/open-source-security (collected 11.02.2022)

Micro Focus. (2022c). *What is Static Application Security Testing (SAST)?* https://www.microfocus.com/en-us/what-is/sast (collected 11.02.2022)

Microsoft Corporation. (2016). *Security for modern engineering: Information security & risk management*. https://www.microsoft.com/en-us/download/details.aspx?id=54092 (collected 08.02.2022)

Moen, R., & Norman, C. (2006). *Evolution of the PDCA cycle*. Asian Network for Quality.

Neess, D. (2022). Interview [Regarding "Imagine possible"] 19.04.2022.

Pasmore, W., Francis, C., Haldeman, J., & Shani, A. (1982). Sociotechnical systems: A north american reflection on empirical studies of the seventies. *Human Relations*, *35*, 1179–1204.

Ramaprasad, A., & Williams, J. (1998). The utilization of critical success factors. *Proceedings of the 29th Annual Meeting of the Decision Sciences Institute*.

Schwaber, K., & Sutherland, J. (2017). *Scrum guide*. https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf (collected 15.02.2022)

Silverman, D. (2017). *Doing qualitative research* (5th ed.). SAGE.

Snowden, D. (2015). *Change through small actions in the present*. https://thecynefin.co/change-through-small-actions-in-the-present/ (collected 14.02.2022)

Snowden, D. (2016). *The adjacent possible*. https://thecynefin.co/the-adjacent-possible/ (collected 14.02.2022)

Sutanto, J., Kankanhalli, A., Carnegie, J. T., Raman, K. S., & Tan, B. C. Y. (2008). Change management in interorganizational systems for the public. *Journal of Management Information Systems*, *25*, 133–175.

van Eijnatten, F. M. (1993). The paradigm that changed the work place: Annals of STSD. *Van Gorcum Publishers*.

Weber, P. S., & Weber, J. E. (2001). Changes in employee perceptions during organizational change. *Leadership and Organizational Development Journal*, *22*, 291–300.

Ziemba, E., & Obłąk, I. (2015). Change management in information systems: Projects for public organizations in poland. *Interdisciplinary Journal of Information, Knowledge, and Management*, *10*, 47–62.

Appendix

A.1 Pilot study

A.1.1 Prior experience

1. What are your relations to security within software development?
2. Can you give an example of what your security approach has looked like in the past?
3. Have there been any challenges with security aspects in the past?

A.1.2 Security to day

4. What does your security work look like today?
5. What do you consider to be the most significant challenges when it comes to implementing safety routines in your daily work?
6. What different types of security procedures are you familiar with today?
7. What do you consider to be a reasonable routine to implement in your work?
8. How do you perceive the existing security directives?
9. Is there anything that can be improved?
10. How is security work-integrated today?

A.1.3 Future of security

11. What role does security work play in today's software development?

12. What are your thoughts on implementing security in the daily workflow? For example, Fortify.
13. What tools do you want to implement in the daily work regarding security? In the IDE or pipeline, or both?
14. According to you, how can security implementation improve your daily work?
15. As a developer, what do you require from your superior in terms of communication when new work methods are about to be implemented?

A.2 Case study

1. Can you describe how top management has supported you in both words and actions. *(Top management, in this case, is referring to a managerial level above your team manager)*
2. How was the need for change presented to you?
3. Can you state an objective for what was intended to change and a possible vision for how?
4. Can you give an example of how the planning activities were conducted, did this clarify necessary tasks and required resources?
5. How would you describe the managerial commitment and involvement at the line level?
6. Were you satisfied with the planned outcomes? Or the actual outcomes?
7. Please describe how the communication, regarding the particulars of the planned and performed changes as well as for sustaining engagement and motivation, has been.
8. Has there been any readiness to deal with the resulting changes? *(That is, appropriate adjustments to the changes in the workflow)*
9. Can you give an example of how you, as an employee, has received training to facilitate the transition to a new way of working?
10. Can you describe your involvement in the change process? Were you satisfied with the planned outcomes? Or the actual outcomes?
11. Has there been a continuous information flow, regarding the state of the change project?
12. Can you give an example of continuous performance measurements, evaluating the progress of the project against goals and objectives?