ANALYSIS OF A SOFTWARE SECURITY SELF-ASSESSMSENT TOOL – CASE COMPANY: VISMA

Lappeenranta–Lahti University of Technology LUT

Master's Programme in Software Engineering, Master's thesis

2021

Elmer Häyrynen

Examiners: Professor Jari Porras

Associate Professor Ari Happonen

Co-advisors: Monica Iovan, D.Sc. (Tech.)

Professor Daniela Soares Cruzes

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
LUT School of Engineering Science
Software Engineering
Elmer Häyrynen

Analysis of a software security self-assessment tool – Case company: Visma
Master's Thesis 2021

71 pages, 10 figures, 4 tables, 1 appendix
Examiners: Professor Jari Porras, Associate Professor Ari Happonen
Co-advisors: Monica Iovan, D.Sc. (Tech.), Professor Daniela Soares Cruzes
Keywords: Security self-assessment tool, security assessment

Ensuring the security of a software product is an increasingly crucial task in the modern software development industry. To be able to assess if our software is secure, we must inspect a multitude of software security related topics, such as ensuring that best practices regarding software security are being used and that software security related processes and monitoring are established and sufficient. In addition to ensuring security, modern software is often required to comply with different legislations and standards, depending on the type of software and the area of operation for the company, which leads to even more topics to address. This study focuses in analyzing and examining the software security self-assessment tool in use at Visma on the time of writing this thesis, which is a tool designed to address these topics, and to ensure that all Visma's software products comply with the corresponding software security-, and legislative requirements. The software security self-assessment tool is analyzed from the perspective of literature, other similar tools present in the industry, and through a user study, to present a general overview of the tool alongside identified potential improvements regarding it. As a conclusion, the proceedings of the improvement suggestions are assessed, and additionally the security self-assessment tool is examined from a broad perspective, and its benefits, capabilities and limitations as a tool are addressed.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
LUT School of Engineering Science
Tietotekniikan koulutusohjelma
Elmer Häyrynen

Tietoturvan itsearviointityökalun analyysi – Case: Visma
Diplomityö 2021

71 sivua, 10 kuvaa, 4 taulukkoa, 1 liite

Työn tarkastajat:	Professori Jari Porras, Professori Ari Happonen
Työn ohjaajat:	Monica Iovan, TkT., Professori Daniela Soares Cruzes
Hakusanat:	Tietoturvan itsearviointityökalu, tietoturvan arviointi
Keywords:	Security self-assessment tool, security assessment

Modernissa sovelluskehityksessä riittävän tietoturvatason varmistaminen on alati tärkeämpää. Tietoturvatason arvioimiseksi on tarkasteltava useita tietoturvan eri osa-alueita, kuten tietoturvallisten käytäntöjen noudattamista sovelluksessa, sekä tietoturvaan liittyvien prosessien sekä valvonnan toteutumista ja tasoa. Tietoturvan varmistaminen ei myöskään yksinään riitä, vaan nykymaailmassa sovellusten täytyy myös sovelluksen tyypistä sekä markkina-alueesta riippuen olla yhteensopiva erilaisten lainsäädännöllisten seikkojen sekä standardien kanssa, mikä puolestaan lisää entisestään tarkateltavien osa-alueiden määrää. Tässä työssä keskitytään analysoimaan Vismalla työn kirjoituksen hetkellä käytössä olevaa tietoturvan itsearviointityökalua, jonka tarkoitus on käsitellä edellä mainittuja osa-alueita ja varmistaa, että konsernin sovellukset täyttävät niihin kohdistuvat vaatimukset sekä tietoturvan että lainsäädännön osalta. Kirjallisuuteen, muihin sovelluskehityksen teollisuudessa käytettyihin malleihin sekä käyttäjätutkimukseen perustuvan analyysin lopputuloksena muodostetaan yleiskuva nykyisestä mallista, sekä esitellään tutkimuksessa tunnistetut parannusehdotukset. Johtopäätöksenä tarkastellaan tutkimustulosten seuraamuksia, sekä korkeammalla tasolla itsearviointityökalun etuja, mahdollisuuksia sekä rajoituksia.

ACKNOWLEDGEMENTS

As my thesis comes to an end, so does my time in the Lappeenranta-Lahti university of technology. The time spent here has been the most eventful period of my life: happy and sad moments, a lot of shared experiences and new friends, and generally the best time of my life. While this time is nearing to its end and new chapters are about to begin in my life, I would like to take this opportunity to thank everyone that has been part of my life here in Lappeenranta: all my friends, Nöörd Hööрd sports team, Software Engineering student's guild Cluster, Videogame association Ruut, Student Union of LUT-University LTKY and the staff of the LUT-university.

A thanks belongs to also those that have helped me with this thesis: examiners, co-advisors, Visma personnel, and friends, out of which a few undoubtedly deserve special thanks. First, thanks to my co-advisor Monica, who introduced this topic to me, and throughout the entire process has guided me with endless patience and unparalleled expertise. A thanks also belongs to my other co-advisor Daniela, for all the support and genuine interest in my topic and work, asking the right questions and often pushing me to walk the extra mile with this thesis. Thank you also to Martin, for his guidance in the early stages of writing the thesis. I would also like to thank both of my examiners, especially Ari, for all their guidance and support throughout the journey. Lastly, thanks also belong to Henna, Niclas and Aaro for all your peer support and inspiration.

Lappeenranta, November 28, 2021

Elmer Häyrynen

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
ATVS	Automated Third-party Vulnerability Service
CAIQ	Consensus Assessments Initiative Questionnaire
CIA	Confidentiality, Integrity, Availability triad
CLASP	Comprehensive, Lightweight Application Security Process
CSA	Cloud security alliance
DAST	Dynamic Application Security Testing
EU	European Union
GDPR	European Union’s General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
HTTPS	Hypertext Transfer Protocol Secure
IAC	Infrastructure as Code
IEC	International Electrotechnical Commission
ISO	International Standards Organization
MSDL	Microsoft Secure Development lifecycle
MVP	Minimum Viable Product
NIST	National institute of Standards and Technology
OWASP	Open Web Application Security Project
PCI DSS	Payment Card Industry Data Security Standard
PET	Privacy enhancing technologies
PHP	Hypertext preprocessor
PMI	Project Management Institute
PSC	Visma product security catalog
RPC	Remote Procedure Call
SAMM	Software Assurance Maturity Model
SAST	Static Application Security Testing
SCA	Software Composition Analysis
SDLC	Software Development Life Cycle
SEDAN	Security Design Analysis
SQUARE	Security Quality Requirements Engineering

SREP	Secure Requirements Engineering process
SSA	Security Self-Assessment tool
SSO	Single Sign-on
STAR	Security, Trust, Assurance and Risk alliance
TLS	Transport Layer Security
VASP	Visma Application Security Project
VCDM	Visma Cloud Delivery Model

INTRODUCTION

Assessing and assuring software security is increasingly crucial in modern software development. As security cannot be considered as a feature or even a set of features, traditional dynamic testing methods and metrics are not suitable for assessing software security. (Arkin et al., 2005) Currently, software security is most often assessed through examining existing vulnerabilities and assessing the impacts of possible breaches, even though software security is really a broader term and in addition should focus for example in assessing the criticality of the software, standard and legislation compliance as well as existing countermeasures for possible breaches. Unfortunately, a metric which would include all the aforementioned factors does not exist. (Shubhamangala, 2015) Another more common approach to assessing software security is assessing the software development practices and processes and ensuring that the security is being built into the software. Secure software development best practices answer this need, and emphasize good engineering practices, understanding threats, including security early in the design process as well as thorough risk assessments and testing practices. (Mcgraw, 2004)

In addition to software security, modern software is required to comply with various existing regulations and standards. This is often a time consuming and expensive process, as the regulations for example in case of GDPR are not easily perceivable and thus understanding them thoroughly let alone being able to assess the compliance is quite impossible without the help of tools designed solely for this process. (Bihari, 2018; Chatzipoulidis et al., 2019) While this does also apply for standards, the main difference between legislation and standards in this case is that the standard compliance is audited by a justified authority before a compliance certificate is admitted, whereas legislation compliance is not audited by default by any authority.

Security self-assessments are effectively a tool built to streamline the process of assessing the software compliance or maturity and consists of questions which guide the auditing personnel to focus on relevant topics during the process. This thesis focuses on analyzing the software security self-assessment tool in use at Visma, which focuses on addressing the security of the company’s software products through examining the related security metrics, existing security policies, assessing possible risks, and examining software compliance with corresponding standards and current legislation effective in the European Union area.

1.1 Background

At the time of writing this thesis, the author is working as a security engineer in a Visma subcompany, and the background for writing this thesis emerges from the need for further development regarding the Visma’s software security self-assessment model. Visma is a multi-national software company based in Norway which focuses on digitalizing business processes in both private and public sector. The company has the greatest presence in the Nordic countries but today has a presence in more than 16 countries globally and continues to expand through numerous company acquisitions. As a result of an acquisition-based expansion, Visma has a substantial number of independent software organizations with their own respective software developer teams and software products. Managing the general security of all the existing companies and products has been a challenge, but despite the challenge the company has established a functional framework for assessing, maintaining, and improving security of both the development process and product security through the Visma Secure Software Development Lifecycle project. The product security assessments are included in a program titled Visma Application Security Program, which includes guidance and support for implementing a set of various automated security tools for any application, as well as a specifically developed security self-assessment. The security self-assessment is a good example of a company requirements-oriented security self-assessment: the same assessment is used by every subcompany and project, and it provides each software development team a way to assess the security of their service, which is both guided and documented by the questionnaire. While there are numerous challenges related to making an assessment to suit various product types, having a common questionnaire also has numerous benefits. For example, this gives the team a common approach to improving security measures and prioritization, guides them to focus on relevant security measures and topics related to their product, as well as helping them to spread awareness of security topics through the team members and share responsibilities. Additionally, as the security self-assessment has been developed through multiple iterations and implemented in various companies and products, the company has an ideal setting for collecting data about the process of conducting a security self-assessment, the effects that the self-assessments have overall, and an understanding of the process behind developing the current iteration.

While a lot of work and thought has been invested into developing the current security self-assessment at Visma, there are still various open research topics around it. Essentially the current model is mainly based on the company's needs and areas of software security which are considered most critical in the corresponding context, and the questionnaire is created by and based on state-of-the-practice and knowledge of company officials rather than being based on scientific study or peer-comparison of other existing security self-assessment frameworks. Topic of this thesis is founded on the idea that while the current model is widely adopted, there are still various areas to study around it, namely in the field of software security literature and legislation on which the assessment is based on, other models present in the industry and especially the experiences of the people working with the model to assess their own software. This thesis is a combination study consisting of a study towards the existing software security self-assessments and best practices in scientific literature and industry, as well as a deep analysis of the Visma security self-assessment tool. The collected results are then reflected on the current Visma security self-assessment for identifying potential improvements. The thesis topic was identified and refined in collaboration with Visma group security officials, and additional help regarding the study was received through various security research contacts. Daily work as security engineer at Visma enables the author to have good insights on how the process of security self-assessment is carried out while also providing excellent contacts within the company, creating an excellent ground for conducting this type of a research.

1.2 Goals and delimitations

The main goal of this study is to analyze the current software security self-assessment, SSA, in use at Visma and identify potential areas for improvement based on literature, industry practices and user research. After the analysis, the role, possibilities, and limitations of the SSA as a tool for improving software security are addressed from a broader perspective. The research considers three high-level topics, bolded in the below list, through the presented research questions.

- **Software security self-assessments from literature and industry perspective**
    - What areas do they focus on?
    - Where are they utilized?
- **Analysis of the current software security self-assessment tool in use at Visma**
    - What areas does it cover?
    - How does it function?
- **User study analysis of the current software security self-assessment tool in use at Visma**
    - What are the benefits?
    - What are the faced challenges?
    - How can we improve the current SSA?

The main limitations of this work are that the assessment will be targeted towards the case company's SSA tool, and as such the perspective is of large or medium software companies operating in the European Union region.

1.3 Methodology

The approach for analyzing the model is conducted in three steps, and each of these steps will utilize a different methodology. The first part of the thesis addresses the current state of software security self-assessments from the perspective of literature and industry and will be conducted utilizing literature review process and aims to answer the corresponding research questions of what areas do they focus on and where are they utilized. Additionally, within this section the background topics required for understanding the Visma SSA are covered to provide a base for conducting the following section. The second section of this thesis addresses the current software security self-assessment in use at Visma and will be conducted by examining the existing model and related company internal documentation. This section aims to answer the research questions regarding the covered areas of the model, as well as inspecting the related process. The third part of the thesis focuses in analyzing the user experiences of the Visma SSA, and answers to the research questions related to the benefits, faced challenges and any potential improvement suggestions with the SSA. This part of the study is conducted by utilizing two different methods of user research: a survey for gathering the general feedback with as large coverage of the internal company personnel that have worked with the SSA as possible, with the addition of an interview study to get a deeper view and understanding to the data collected through the survey.

1.4 Structure

The first part of the thesis focuses on studying the backgrounds of the security self-assessment tools. The viewpoints used here are those of scientific literature including the topics of software security and specifically software security assessment and measuring. From the industry point of view, the focus will be on identifying existing self-assessments and assessing their use cases and background. Additionally, the backgrounds regarding the areas covered by the Visma SSA are covered from the perspective of literature to form a basis of understanding of the topics that the model addresses.

In the second part, the focus is on conducting an analysis towards the SSA in use at Visma. The assessment will be analyzed with regards to its contents, background and how the various topics are addressed, documented, and assessed.

The third part of this thesis will address the results of the conducted user research and present the methodologies used and the results of the research. The potential effects of the uncovered results with regards to the model are further addressed in the discussions section.

In the fourth part, the identified improvement suggestions that were collected and analyzed from the results of the conducted user research are covered. This section effectively includes the outcomes for conducting the study in the form of concrete improvement suggestions for solving the identified challenges.

Lastly, the conclusion section will generalize results of the work and address the proceedings from the findings of this analysis, as well as address the benefits, challenges, and limitations of the SSA as a tool for the purposes of improving software security in other contexts similar with the case company.

2 SOFTWARE SECURITY ASSESSMENTS BACKGROUND

To understand software security self-assessments, one must first understand the background of software security. Software has been developed for more than 50 years and the focus has been in delivering high-quality functionalities to users and other stakeholders. However, it was not until the early 2000’s until the focus properly shifted towards creating secure code (Williams et al., 2018), even though the idea of creating secure software or engineering software in a way that it will function correctly even under malicious attack is much older. (McGraw, 2006, p. 24) At the time, the increase in e-commerce and increasing availability of software services to the common people foretold the technology revolution. In addition to common people being oblivious in the context of cyber security, the same problem also existed among the software practitioners as the general level of understanding about secure software was not ideal. While general cyber security practices such as firewalls and cryptography were widely known and applied, the topicality of software security gained a lot of attention due to the ineffectiveness of other security approaches. (McGraw, 2006; Viega & McGraw, 2001, pp. 1-2) From this identified need, the software security research field emerged with the focus in educating and helping practitioners and developers on how to create secure software, supported by several topical academical book releases. (Anderson, 2001; Howard & Leblanc, 2003; Mcgraw, 2004, p. 80; Viega & McGraw, 2001)

Software security as a scientific research topic is quite difficult to define and comprehend, as the focus is purely on virtual phenomenon and technical artefacts instead of a real-world phenomenon. Security in the context of information technology is often defined through three components: confidentiality, integrity, and availability, widely referred to as the CIA triad. Software security as a field of study focuses in studying how software can be built in a way that it will continue to function correctly under possible malicious attacks (Mcgraw, 2004), or in other words, so that the confidentiality, integrity, and availability of the software are not compromised. In addition to being difficult to define, software security as a phenomenon is also quite elusive as it cannot be considered a static feature since security is relative to context and as such a concept of “complete security” cannot exist. (Mcgraw, 2004)

2.1 Assessing and measuring software security

In the past, the topic of measuring security has been widely discussed in scientific literature during the first decade of the 21st century. The research has resulted in several different techniques, models, and approaches for measuring software security which have been adopted by several standardization organizations such as the U.S. National Institute of Standards and Technology (NIST) and International Organization for Standards (ISO) and have been used as a base for establishing standardization frameworks concerning security metrics such as the ISO Common Criteria, ISO/IEC 27004, and NIST 800-55. (International Organization for Standardization, 2005, 2018a; Mellado et al., 2010; National Institute of Standards and Technology, 2008)

Today the security of software has become a critical requirement for any software. Even though security is required by various stakeholders and clientele from software, it is often unclear if all the effort and money aimed towards improving software security cause a desired effect. The reason behind this is that security often cannot be measured in a way that traditional quality attributes can, which is further complicated by the fact that a common set of properties of security does not exist. (Scandariato et al., 2006a) In order to measure and assess software security, we must first understand what security means in the context of the target software. Utilizing security goals is one of the common approaches for solving this problem, and essentially aims to answer the questions of what we are trying to protect and from whom. After the security goals are defined, the next step is to identify what areas do we need to deal with to achieve the desired goals. CIA triad is one of the earliest attempts to define these areas of software security, and while confidentiality, integrity and availability are often included in modern classifications, the newer adaptations also tend to additionally address topics such as organizational policies, security goals and general security policies. (Islam & Falcarin, 2011) In the past two decades, the scientific research regarding the areas of security has focused in identifying security goals from different perspectives: technical perspective, human factor, business and economic perspectives and governance and legislation perspectives. As a mutual understanding of these security goals does not exist, every author usually makes their own interpretation of them which leads to a multitude of different models examining the same phenomenon through different goals. (Cherdantseva & Hilton, 2014) As a result of missing the mutual understanding related to security goals, there are also numerous different approaches for measuring software security.

One of the approaches for ensuring security in software is to assess security through security requirements and features. In traditional requirements engineering, the security requirements of a software are considered non-functional properties of software which are realized through implementation of security features, such as password protection, firewall and authentication. (Ramachandran, 2016) Uncovering the security requirements for a software is an extensive task, since it requires lots of data to begin with: security goals, security risks and compliance requirements all need to be defined before the process can take place. A proceeding from Project Management Institute (PMI) Congress 2015 compares and summarizes security goals that various SDLC models such as the OWASP CLASP (Comprehensive, Lightweight Application Security Process), Microsoft SDL (Secure Development Lifecycle), SREP (Secure Requirements Engineering Process) and SQUARE focus in, and identifies the security goals as Confidentiality, integrity, Availability, Accountability and Conformance. The research generalizes the areas of security typically considered are Authentication and password management, authorization and role management, audit, logging and analysis, network and data security, Code integrity and validation testing, cryptography and key management, data validation and sanitization, and third-party component analysis. (Danziger & da Silva, 2015) Once the security requirements of a software have been identified, the general level of security of a software can be roughly assessed based on if the software fulfills all of them sufficiently. This type of an approach appears suitable for the purposes of creating a software security self-assessment questionnaire mainly because requirements engineering related literature identifies commonly used security requirements, which can be quite directly transformed into assessment questions. However, this approach is prone to misunderstanding security as merely a sum of different implemented security features and therefore this approach alone cannot be used to accurately assess or measure the overall level of software security.

Metrics in general enable more specific assessment and tracking of software security and development, and enable characterization, evaluation, prediction, and improvement of software security. (Savola, 2007) In addition, security metrics can also be effectively used to assess the security risks during the software development process. Security metrics can be derived from the identified software security requirements. Transformation can be carried out in many ways, for example by utilizing the Goal / Question metric approach. (Jain & Ingle, 2011) In a 2011 conference paper by Islam and Falcarin, “Measuring security requirements for software security”, they present a two-phase approach for the entire process. Their approach focuses in identifying software security requirements, and then utilizing Goal-Question-Metric approach for generating questions that define the requirements as completely as possible through quantification. Their research also uncovered some of the underlying problems with their proposed solution, which are related to the fact that security is a multi-faceted concept and state that subjective evaluation of these metrics is hard as there are numerous ways to fulfill the security requirements and comparing different solutions is difficult. (Islam & Falcarin, 2011) In this light, it is quite evident that deciding on what topics, features, and assets a software assessment should focus on to cover all aspects of security is quite difficult, as a commonly agreed collection of these topics does not exist in the scientific literature.

One of the attempts to avoid the previously described problem is BSIMM, a model for improving software security introduced by McGraw et al. in 2008. The idea behind the model is that while the security of a specific piece of software cannot be accurately measured, an approach to ensure security is to measure the processes and measures that companies that can be considered to produce secure software are doing. The BSIMM model consists of four domains of security and their corresponding practices, presented below in table 1. (McGraw et al., 2015)

Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and metrics	Attack models	Architecture analysis	Penetration testing
Compliance and policy	Security features and design	Code review	Software environment
Training	Standards and requirements	Security testing	Configuration and vulnerability management

Table 1: BSIMM framework (McGraw et. al., 2015)

All the presented sections and subsections contain activities related to software security, that are presented in three levels which represent the level of maturity. From the point of introduction in 2008, the BSIMM has been utilized by ever increasing number of companies, and the data in the current iteration, BSIMM 12, is gathered from more than 128 software security initiatives from different companies. In this light, the BSIMM can be quite effectively used to compare the maturity of your own processes and measures towards those of other companies in the industry. (Migues et al., 2021)

2.2 Security self-assessments

The concept of secure software development is either encouraged or enforced by various process methodologies, standards, and legal regulations. Process methodologies focus in defining how the software security best practices can be implemented into the software development lifecycle, examples of such security-aware methodologies are OWASP Comprehensive Lightweight Application Security Process (CLASP) and Microsoft Security Development Lifecycle (MSDL). (Scandariato et al., 2006b) Regarding security-aware standards, one of the most recognized is the ISO 27001 standard, which is applicable to all types and sizes of organizations and focus on presenting a model on how to set up and operate an information security and management system. (International Organization for Standardization, 2018b, pp. IV–V; Šikman et al., 2019) Other existing software security standards and regulations in particular tend focus in defining requirements for storing, handling and processing specific type of data: Payment Card Industry Data Security Standard (PCI DSS) focuses in credit card data (Payment Card Industry, 2013), Health Insurance Portability and Accountability Act focuses in patient data (HIPAA) (U.S. Department of Health and Human Services Office for Civil Rights, 2016), and the General Data Protection Regulation (GDPR) in Europe focuses in personal data. (European Parliament and the council, 2016) The main difference between standard and regulation compliance is that the regulation compliance is mandatory whereas the standard compliance is often more of a recommendation. The motivation for complying with software security standards for example in case of ISO 27001 series is varied: the standard helps with complying to legislation and regulations, helps to demonstrate “fitness for purpose”, different insurance reasons, gives a competitive edge and eases securing customer and supplier chain contracts as standard compliance clause is often required or at least highly valued by the client. (Humphreys, 2016, pp. 186–187) whereas in the case of

PCI DSS the certification is currently obligatory. (Everett, 2011; Payment Card Industry, 2013)

Security self-assessments are effectively tools that could be addressed as checklists: they are built to provide a structured way to assess the software compliance or security maturity and consists of questions that guide the auditing personnel to focus on relevant topics during the assessment process. Software security self-assessments are generally not recognized in the scientific literature, but self-assessment tools are used in the industry to ease the process of ensuring compliance with, for example, different legislation and standards. Today, various self-assessments for these purposes exist, and they are often either official self-assessments provided by corresponding authorities as in the case of PCI DSS (Payment Card Industry, 2018), or unofficial self-assessments created by third party actors, such as the GDPR related self-assessment questionnaire developed by the United Kingdom's Information Commissioner's Office (Information Commissioner's Office, 2021) and the ISO27001 maturity assessment tool developed in collaboration by Educause, the Higher Education Information Security Council and B. Benyammi as part of the IsecT ISO27001 security related development. (IsecT ltd., 2021)

The self-assessment tools can also be utilized to assess the general level of software security. Two examples of this type of assessments are the open-sourced Software Assurance Maturity Model (SAMM) developed by the renowned Open Web Application Security Project (OWASP), and the Consensus Assessments Initiative Questionnaire (CAIQ), developed by the Security, Trust, Assurance and Risk alliance's (STAR) (CSA, 2021b; OWASP, 2020). These software security assessments are essentially comprehensive tools for companies for defining their current standing in sense of software and information security and establishing a concrete set of topics to focus on improving and documenting. While the focus of the SAMM and CAIQ are similar with Visma's SSA in the sense that they aim to enforce software security, they differ based on what they define as secure: both essentially assess the compliance of the software with their corresponding security framework. SAMM is a risk-driven model aimed towards any type of an organization (OWASP, 2020) The CAIQ, on the other hand, is aimed towards Infrastructure as a Service, Product as a Service and Software as a Service companies and is used by various large actors such as GitHub and Adobe for example. (CSA, 2021a, 2021b) The CSA CAIQ is very extensive in comparison to SAMM for example, as in its current version 4.0.3 there are more than 250 questions. The CAIQ aims to ensure that the assessed software complies with the alliances proposed security standard. (CSA, 2021b) While the assessments are based on different frameworks, their general purpose is very similar and the key areas of focus for both assessments, illustrated below in table 2, also appear to cover very similar topics related to software security.

CSA CAIQ 4.0.3	OWASP SAMM v2
• Auditing and assurance policies • Application and interface security • Business continuity and operational resilience • Change control and configuration management • Cryptography, encryption, and key management • Datacenter security • Data security, privacy, and lifecycle management • Governance, risk, and compliance • Human resources • Identity and access management • Interoperability and portability • Infrastructure and virtualization security • Logging and monitoring • Security incident management, e discovery and cloud forensics • Supply chain management • Transparency and accountability • Threat and vulnerability management • Universal endpoint management	• Governance ○ Strategy and metrics ○ Policy and compliance ○ Education and guidance • Design ○ Threat assessment ○ Security requirements ○ Security architecture • Verification ○ Architecture assessment ○ Requirements-driven testing ○ Security testing • Operations ○ Incident management ○ Environment management ○ Operational management

Table 2: Areas of focus in CSA CAIQ and OWSP SAMM (OWASP, 2020, CSA, 2021b)

2.3 Data protection

Data protection is technically not a topic of software security in a traditional sense, but something that could be considered an asset which software security aims to secure. By definition, safety refers to a fundamental state of being free from all types of harm. (Merriam-Webster, 2021) The protection scheme in figure 1 illustrates the relationship of security and privacy: safety is the fundamental foundation for protection and provides measures against accidental and unintentional danger. Security is the second layer of protection, which is designed to deal with intentional attacks and provide measures to prevent causal means for danger. On the top of the pyramid is privacy, which is enabled only if the two previous layers are realized. Privacy refers to the protection of properties, rights and assets from an unauthorized intrusion or public's attention, and in the case of software security the data is the critical asset which we want to protect. (Eltahawy, 2021)

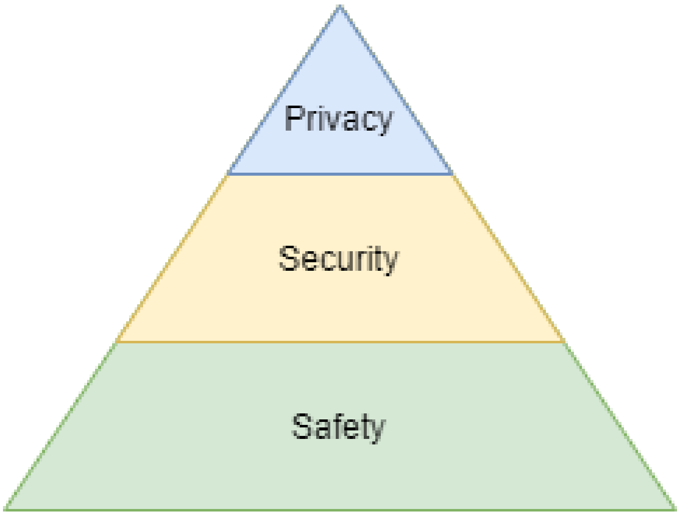

Figure 1: Protection scheme, based on lecture notes by Eltahawy, 2021

Privacy by design and data protection topics were present in the media during the release of GDPR in 2018, but privacy by design is not a new idea. The concept is very closely related to the concept of privacy enhancing technologies (PET), which was first introduced in the report "Privacy-enhancing technologies: the path to anonymity" in 1995. (Borking & Hes, 1995) From the release of the article to today, it could be argued that the concept of privacy enhancing technologies has been fully adopted. (Hustinx, 2010) Today, within the European Union data protection has been enforced by the Union's General Data Protection Regulation from 25 May 2018, and at the core it requires that all authorities that control or process personal data must comply with the technical and organizational requirements to ensure data security. The GDPR generally enforces the use of technologies and privacy and data protection by design in all software that handles personal data in any form, (European Parliament and the council, 2016) and when considering that Visma operates in EU, the software produced by the company is obliged to comply with the GDPR requirements. To ensure this compliance, the SSA contains a dedicated data protection section, which is further discussed in chapter 3.3.

2.4 Risk assessment and management

To better understand risk assessment and management, we first need to define what a risk is in the context of software security. Risks can be perceived as scenarios where the software security is compromised, and when reflecting on the CIA triad-based definition of software security, this would be a situation where confidentiality, integrity or availability of the software is compromised. At the core, security is about reducing the risks of an organization, business, or software to a tolerable level, and as such risk assessment and analysis is one of the core practices of software security. (H. Chivers et al., 2009) Risk analysis aims to identify possible risks related to a software and quantify them based on the required effort to fix them, as well as the potential likelihood and severity of the risk (H. Chivers et al., 2009; Mkpong-Ruffin et al., 2007), and thus enabling the management to make decisions about system security. When a risk is identified, there are two options: the risk can be either accepted and left undealt with, or the risk can be mitigated by actions such as enabling more controls within the system or reducing the potential impacts. (H. R. Chivers, 2006)

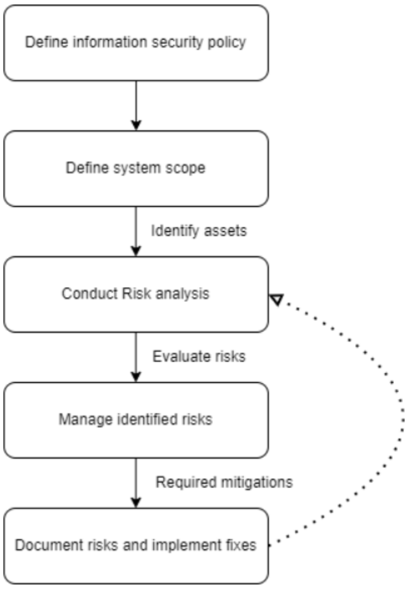

Figure 2: Typical risk management process, based on a model by H. R. Chivers, 2006

According to Chivers, a typical risk management process generally follows the steps presented in figure 2. Pre-requirements for performing a risk analysis are that the information security policy and system scope are defined. Information security policy refers to defining specific access controls and the scope of the system specifies the boundaries of the management process as well as identifies the system so that the risk analysis process can be started. Risk analysis process focuses in utilizing all available information sources to identify any potential threats to the software and enlist them along with their relative risks to the secure functionality of the software. After the risk analysis has been conducted, the next step is risk management, which focuses on how each of the identified risks should be managed. The last step of the risk management process is to document the results of this process and to carry out the potentially required mitigations. (H. R. Chivers, 2006)

3 VISMA SECURITY SELF-ASSESSMENT OVERVIEW

The Visma security self-assessment is an integral part of Visma Application Security Program, VASP. The VASP consists of various security solutions, such as static and dynamic application scanning services, automated threat vulnerability support services, software composition analysis services and even a bug bounty program that the software can be onboarded to. The application security program is also lightly gamified: the required level of security for each software is defined as a tier: platinum, gold, silver or bronze, and the level is monitored using a dedicated security index. In the index, the tiers have their corresponding score limits, and any unfinished onboarding processes or pending security fixes of identified vulnerabilities gives penalty points. The SSA is a part of the VASP, and it is used by the service development teams to assess and document the security of their software.

Effectively the SSA acts as sort of a checklist for the teams to focus and assess their software, as well as provide a structured way to then prioritize and carry out the identified improvements. Using the assessment also helps to spread awareness of security topics through the team members as well as share responsibilities related to security. Currently the security self-assessment being used is the 2nd iteration, and at its core the questionnaire focuses in identifying the assets of the software that require to be protected, potential threats related to the software security, and utilizing the data gathered by tools about the possible vulnerabilities within the software. This gathered knowledge creates a base for conducting a risk assessment study, where the knowledge of the aforementioned areas is combined and used to uncover any potential risks related to the software. After the risks have been identified, the general level of risk related to operating the software can be assessed and the uncovered risks can be managed accordingly.

3.1 Process

Filling SSA as a process is currently almost entirely a manual process, and similarly to onboarding a software to the other solutions included in the VASP, is generally on the responsibility of a dedicated security engineer. These security engineers are in most cases people with a technical background, who are additionally responsible for monitoring security activities and conducting the security work related to the software. The SSA itself is currently located in Confluence; a team workspace developed by Atlassian software. (Atlassian, 2021) The process is initiated by creating a dedicated page for the assessed software, based on the SSA template. The page contains three main sections: **Risk assessment and management**, **Data protection** and **Security**, which are individually addressed in the following chapters. Each of the main sections addresses a different high-level topic and includes subsections for addressing relevant lower-level topics. Each subsection contains a set of instructions explaining the lower-level topic and its background, and a set of questions which vary from simple yes or no -type of questions to generating detailed documentation or for example drawing a system diagram. Once all the questions and topics have been addressed, the assessment is sent for review. The reviewing process is also currently entirely manual and conducted by the SSA development team members for the security and risk assessment and management part, and by compliance managers for the data protection part. Once the SSA has been reviewed, the findings of the reviewing personnel are often addressed in a meeting with the development team members and product owners. After the discussion session, the SSA will be accepted, and the remediation of the possible findings takes place. The SSA is also revised yearly, and any changes in the system that also require the contents of the SSA to be updated need to be addressed. After reviewing the SSA every year, it is again sent for review and if necessary, another meeting with the team and the reviewing personnel will be conducted.

3.2 Section 1: Risk assessment and management

As discussed previously in section 2.4, to be able to conduct a comprehensive risk assessment, a vast amount of data is required to be collected from the system. Fortunately, much of this data is already available at later stages of the SSA through filling the prior data protection and software security sections of the Visma SSA, therefore conducting the risk assessment as the last step of the SSA makes a lot of sense.

The risk management component in the Visma Application Security Program generally follows the typical risk management process presented in figure 3 and serves a dual-purpose: it aims to illustrate the risks of a product to the corresponding managers, so that it is visible and can be managed, while also enabling the corporation to manage risks at the portfolio level and maintain a risk register and track the changes. The desired outcome of carrying out the risk assessment and management process is that users in various roles, such as developers, security engineers or other stakeholders have a straightforward way of identifying and registering a risk to be assessed by the management. The process also defines a tool and the process for managing the identified risks of the software, but leaves the process of defining the values, roles, and risk acceptance to the team to be decided independently to ensure compatibility with various products and subcompanies. The Risk assessment and management section consists of four individual subsections. **risk profile**, **risk review**, **risk assessment** and **risk register**, which are addressed in the following chapters.

3.2.1 RM01: Risk profile

Assessing the risk profile is the first step of security self-assessment. Risk profiles as a concept were first introduced as a part of Security Design Analysis (SeDAn) framework to illustrate the risks related to individual sub-systems and components of a software. Risk profiles are a result of attack analysis and aim to specify the risks that the system would be exposed to in a case that a sub-system or component would be compromised. They also indicate the extent to which it must be protected. (H. R. Chivers, 2006) and therefore are a logical first step for conducting the risk assessment.

Instead of identifying all different components of the software and generating individual risk profiles for them, a single profile is generated for the whole application. The profile is generated based on questions which determine what type of a service is being examined, identifying the types of data the service processes as well as identifying the customers and number of records which the service manages.

3.2.2 RM02: Risk review

Risk review is chronologically the third-last step in the security self-assessment and when compared to a traditional risk-assessment process as presented in figure 2, this step fulfills the first subsection of the risk analysis step. Main purpose of the risk review is to enlist all the uncovered risks related to the software. Risks listed here are divided into three categories based on the method they were uncovered by: firstly, the issues which have been uncovered as a part of the security self-assessment, secondly the open risk issues identified by a risk-assessment process, and lastly the risks uncovered by any other available information such as incident reports, automated security tools or previous assessments.

3.2.3 RM03: Risk assessment

Risk assessment could be considered the culmination point of the whole security self-assessment. At this stage, all information collected as part of the SSA process, including data and assets that the software is responsible for, identified vulnerabilities and threats in the application from the security assessment questionnaire, and possible risks derived from the identified vulnerabilities and threats are now available for conducting the risk assessment. Main purpose of this section is to inspect the general level of security of the software based on the forementioned factors, and by the reviewers best professional estimation to try to find an answer to the following question: "does the application provide a level of security that is appropriate to the risk represented by operating it?"

3.2.4 RM04: Risk register

Risk registering is the last step of software security self-assessment. Effectively, this section summarizes all the risks identified in the previous step and the main purpose is to show all risks associated with the software in a listing. It also helps the officials to keep track of the status of each risk and maintain a view of the software risks and the status of their remediation process in a single place, as the risks are additionally registered into a company-wide risk project register.

3.3 Section 2: Data protection

As previously addressed in section 2.3, it is evident that the GDPR has an impact on nearly every software product that the company offers as the main area of operation for the Visma corporation is in Europe. Data protection section of the software security self-assessment aims to ensure that the software is compliant with the corresponding requirements of GDPR and other enforcing factors for data protection, such as software related terms of service and customer contracts, which often address topics such as company related information confidentiality. These contracts also often define compliance with local legislation and regulations, for example in the case of bookkeeping data the duration from which the data must be accessible is often defined by the local laws and carried out by the software. The data protection section consists of five individual subsections addressing the following topics: **data list**, **data classification**, **privacy and data protection by design**, **formal requirements and standards**, and **customer contract and supported version**, which are covered in the following chapters.

3.3.1 DP01: Data list

Data list is a section where all data that is being used by the system is identified. Generally, this requirement defines the base for all latter subsections of the data protection section, as the types of data within the system define the level of security required for the application. The data is divided into two sections: user's personal data and customer data. Personal data management is more strictly enforced by the GDPR while the customer data is often defined as confidential by the corresponding terms of usage and customer agreements. The data which the system uses is gathered from the existing software documentation, or in cases where the documentation is insufficient, gathered from within the system database tables, column headers and similar sources. It is also noted that only data which the system is designed to process should be identified here, as in some cases it is possible for the customers to configure or use the system in unorthodox manner and store sensitive data which is outside of the knowledge or control of the system administrators.

3.3.2 DP02: Data classification

Once all the different data that the system is using has been identified, it must be classified to fully understand the requirements related to storing and processing it. This assessment process should be based on the assessment of risk related to the data as in "What is the state of security in our system?" and "What data do you have within the system- and what kind of threats possessing or processing this kind of data proposes?" The classification process uses a classification model which is essentially a combination of GDPR and CIA model. GDPR requires the teams to be able to identify the owner of the data and define the role of the company in processing this data. Additionally, the CIA model is utilized to create a three-staged classification grid that helps to classify the data to get an understanding of the required level of concealment.

Confidentiality	**Integrity**	**Availability**
Highly restricted	Assured	High

Restricted	Controlled	Standard
Public	Uncontrolled	Not time-critical

Table 3: Data classification categories

In the classification model presented in table 3, the strictest level of confidentiality of identified data is defined as highly restricted. This refers to data with high confidentiality, such as sensitive personal data, high profile, or politically exposed customers, which require extra levels of organizational and technical protection to ensure the confidentiality of the data within the system. The default level of confidentiality is restricted, which means that only individuals who have a need to access the data are granted access, which resembles the "least privilege" cybersecurity best practice. The least strict level of confidentiality is titled public, which refers to data being available in a public domain and as such no particular security measures are required to protect it.

Second area covered is integrity, and the strictest level requirement for data is defined as assured, which applies in cases where the data by nature requires extra levels of organizational and technical protection to ensure the realization of confidentiality, such as log-tamper proofing. Default setting in this case is that the data is controlled, which means that the data is being protected from unauthorized modification and access through typical solutions, such as security logging and identity and access management methods. The least strict level of integrity is titled uncontrolled, which means that the data's integrity is not crucial and as such no particular measures to ensure it are required.

The last area covered is the availability requirements for the data. The strictest level of availability is defined as high and refers to cases where the data is time-critical and as such requires higher availability than normal data. Standard case is titled default, which reflects the availability requirements included within the customer contracts. Least strict level of criticality is the data which is not time-critical, but often this only applies to rare instances of data such as test data or similar which has no requirements for availability.

3.3.3 DP03: Privacy and data protection by design

The privacy and data protection by design section is the largest individual section within the SSA, as it addresses a total of nine different topics that fall under this high-level category. These nine topics will be covered under this chapter.

The first topic of privacy and data protection by design is assessing purpose, minimalization and proportionality of the data. This section is heavily based on the principles related to processing of personal data introduced in article 5 of the GDPR. These requirements require the processing of personal data to be lawful, fair, and transparent, that the purposes of using the data should be limited, data should be minimized, data should be accurate, the time of storing the data must be limited and integrity and availability of the data must be ensured. (European Parliament and the council, 2016) This translated into three focus points for this section: purpose, minimization, and proportionality. Proportionality refers to cases where something corresponds in size or amount of something else, which within the context of privacy and data protection means that the processing of the data must be proportionate relative to the principles of data protection, such as minimization, lawfulness, and purpose limitations. Minimalization means that the system should only process the minimum amount of data necessary to complete the operations, or more specifically that the data must be adequate, relevant, and limited to what is necessary for the said purpose.

The second topic of privacy and data protection by design is data deletion. Data deletion policy in all Visma products is required to be documented in a data deletion policy document, which is based on the data deletion policy of the corporation. This section in the security self-assessment focuses on examining how the policy is implemented within the software and identifying if there is any need for improvements in the data deletion procedures.

Third topic addressed in this section is data export and return, which is addressed through a single question: “does the system provide functionality for the customer to export their data in a common format?” This is effectively to ensure compliance with the GDPR enforced right to data portability, which requires the data subjects to be able to obtain any of their data possessed by the controller in a commonly used, machine readable format and use it for their own purposes. (European Parliament and the council, 2016)

Fourth addressed topic is the systems capability to restore customer data, which is also covered through a single question: “Does the system have data restore capabilities?” It is also noted that this is merely a recommendation, so there is no requirement for this, which is quite surprising. Data restore can also be considered a fundamental requirement for an application from the perspective of infrastructure.

Fifth topic is customer guidance and instructions. As well as the two prior sections, this section also consists of only one question: “Is a guideline available to assist the customer in configuring and using the system appropriately?” The guidelines are also noted as a recommendation, so they are not strictly enforced.

Sixth of the topics covered is automated decision making, which focuses on determining if the system utilizes automated decision making in any area and if so, that the automated decision making is handled according to the article 22 of GDPR, the automated individual decision-making, including profiling. (European Parliament and the council, 2016)

Seventh topic is pseudonymization and anonymization of customer data. In this context, it is noted that pseudonymization should be considered a security measure only, and it does not permit processing for any different purposes for what the data was originally authorized. Anonymization is most often used when generating test data based on real data, and it is noted that anonymizing the data is a processing activity and therefore it cannot be used to permit processing for different purposes than what the data was originally authorized for.

Eighth topic considers the consent from the data subject. It is explained that consent is the legal basis for all processing of personal data, further defined by the contract with the customer and legitimate interest. The consent is always obtained directly from the data subject. This section can be considered sufficient only if the application fulfills all defined requirements, and the way that these requirements are fulfilled is properly documented.

The last topic addresses data breaches. The section contains only one question, which aims to ensure that the development team is aware of the corporation’s incident handling routines, and assistance that they can obtain in case of a possible breach.

3.3.4 DP04: Formal requirements and standards

The formal requirements and standards section refers to announcing Visma’s copyright and other proprietary rights in and to Visma’s software, as well as complying with any obligations towards third party licensors. All Visma software should contain a copyright notice and place it in a way that gives all users a reasonable and permanent notice of copyright. The copyright notice should be given before the product is purchased or accessed by the customer, as well as easy to find after the purchase or when the actual product is being used. The only exception to this would be any provided APIs, as by definition they are not subject to copyright and therefore should not carry any notices of copyrights. In addition to displaying copyright information, the trademark notice should be placed in the immediate vicinity of the copyright notice.

Another aspect is compliance with third party licenses. Modern software is often built by utilizing third party components. A typical component would be, for example, a code library, but also any code snippets or similar can be protected by copyright, or other similar licenses. As most components are licensed, it must be noted that these licenses often result in certain obligations and restrictions once they are used. Using open-source code in this sense is often quite risky and difficult, as numerous open-source licenses can be difficult to understand and the obligations are often based on how the component or library is being used. Copyleft refers to open-source licenses which permit an actor to freely distribute copies and modified versions of the software with a condition that your own software also uses the same open-source license. This within the context of Visma is particularly tricky, as once copyleft-licensed software is used within a software, it may result in that the software goes from being copyrighted to the developing company to being subject to the open-source license and as a result the software would no longer be the developing company’s intellectual property. Closed source refers to any copyrighted software from third parties which is used within Visma’s software. In these cases, the rights, duties, and obligations depend mainly on the contact between Visma and any related third parties.

Assessing all obligations resulting from different types of third-party licenses is quite impossible without generating a Bill of Materials first, which is a document containing all third-party components used within the application. The bill of materials can be generated manually by utilizing tools included in the used package management software, or automatically generated by an automated third-party component security scanning software, but due to the nature of modern software development going through the generated documents often prove to be very tedious due to the volume of 3rd party libraries.

3.3.5 DP05: Customer contract and supported version

This section covers any customer contracts and the supported versions of the software, which define the software's obligations towards the customer, including the legal basis for processing data discussed above. The standard base for contracts is often the corporation provided Visma Software Terms of Service, but in some cases the software may be sold under different contracts in different markets. The supported versions are often defined in the contract and pertain directly to the product warranty through backwards-support and compatibility that the software is required to provide to the customer. Fulfilling the requirements of these sections requires the software to have a Terms of Service bill, which includes accurate descriptions of these requirements.

3.4 Section 3: Security

As previously addressed in section 2, software security could be considered the elusive target for the software to minimize the potential risks as well as provide a foundation for privacy. Software security itself can be considered an emergent property of a software system, and takes to account different things, such as software vulnerabilities, security mechanisms and design for security. Identifying different vulnerabilities is achieved through code review, which is nowadays often an automated process conducted by static code scanning software. However, no such automated tool exists for inspecting the security mechanisms and secure design. To verify that the required best practices and safety measures and features are in place, a more in-depth review of the software is required, which is essentially the core of the security part of the Visma security self-assessment.

While the contents of the security section are not strictly based on any existing standard or other assessment but rather on the areas of security that the software security assessment development team has identified and considered to be of interest, it is evident that the different subtopics addressed in the security section cover very similar topics to those present in the two self-assessments presented in section 2.2. The security section consists of 17 individual subsections with different software security topics, and they are addressed in the following chapters, which concludes the SSA overview section.

3.4.1 SEC01: System diagram

The first step of the security assessment is drawing a system diagram. The type of required system diagram is not strictly defined, but the main idea is to document all external actors related to the system such as integrated systems, integrated support systems, data storage systems, end users. support users, operational users, and developers.

3.4.2 SEC02: Attack surfaces

Attack surfaces as a concept was introduced by Michael Howard for the purposes of software risk assessment in 2003, and defined the concept as a list of features that an attacker could attempt to compromise: any open sockets, open RPC endpoints, open named pipes, services etc. (Howard, M. 2003) According to Manadhata Et. al., the attack surface can be considered to consist of three different sets: a set of data entry and exit points, a set of system channels and a set of untrusted data items. (Manadhata, K., Karabulut, Y. & Wing, J., 2009)

The approach in the context of Visma SSA differs a little from the more traditional models described above, and instead provides a quite simplified approach to this topic. Within this section, the system is inspected from the perspective of an attacker. The focus is to identify and classify all interfaces of the software and assess their functionalities. These functionalities include things such as access control, methods of authentication and authorization, and used technologies, and as a result an overview of the system's interfaces, or attack surfaces, is generated.

3.4.3 SEC03: Access control quality

Access control quality refers to the technical measures related to managing the software access control. In the security self-assessment, the focus of this area is on the technical realization of the software access control and aims to gather information on minimum proportionality of the user data access, customer data isolation practices, technical access control check procedures as well as general monitoring and failure prevention and recovery mechanisms. As the inspection of these topics is conducted through a manual code review, the accuracy of the results is heavily dependent on the level of competence of the reviewer.

3.4.4 SEC04: Password storage

Password storage is a very shallow section and aims to document the practices related to user passwords storage. Additionally, the method of storing these user passwords is identified and verified to follow the best practices to provide increased security in case of a potential breach.

3.4.5 SEC05: Crypto/hash algorithms

The crypto and hash algorithms section aims to identify what data is being secured through them, as well as document on a deeper level how the algorithms have been configured. The tool used for this section is a simple table which collects information related to each algorithm that the system implements. The documented things are the purpose and usage and type of the algorithm, as well as more specific information of the algorithm including what kind of keys are used and where are they stored as well as if any specific cryptographic frameworks are used in conjunction.

3.4.6 SEC06: Application misuse

The application misuse uses misuse scenarios to assess the system’s responses to potential attacks. The scenarios are divided into three subsections, and each of them focuses on a specific scenario. For each scenario, a set of actions and desired responses are defined, and the software is assessed based on how well they fulfill these desired responses to the proposed activities.

3.4.7 SEC07: Software dependencies

Software dependencies section focuses on identifying how the existing and known vulnerabilities in the used 3rd party libraries are handled. In addition to enlisting all components used within a project, the necessity of all identified components is assessed as legacy components that are not in active use often pose an unnecessary risk. After analyzing the dependencies and existing vulnerabilities, the focus shifts to the secure development processes on how the teams oversee and monitor as well as update, fix and replace these components.

3.4.8 SEC08: File upload validation

If the assessed system supports file uploading by the user, the policies regarding file validation are documented and assessed. Firstly, the system itself or any component or external system that reads, parses, or maps the software in any way is identified. Then the focus turns to how the file storage and security validation is handled. The recommended best-practices are that the physical name or folder of the provided data should not be controlled by the user, and that the files should be either scanned for viruses utilizing an anti-virus solution or by utilizing other thorough file content checking methods.

3.4.9 SEC09: Secrets in source code

In this section, the reviewer is instructed to inspect critical parts of the source code and verify that there are no secrets, such as passwords or different encryption keys or authentication tokens, included in the source code. The requirement in this section is strictly that there should not be any secrets present in the source code, and in other cases the team should opt for a secret management solution.

3.4.10 SEC10: Secret management

The secret management section acts as an enlisting section for the secrets used within the application. The purpose of each secret, as well as the storage methods, confidentiality, secret strength, and change procedures for each individual secret are assessed. The suggested best practices for ensuring confidentiality are to use secret management tools. Regarding the change procedure, the teams are also instructed to include any existing internal documentation in case it exists.

3.4.11 SEC11: Phishing

By definition, “Phishing is a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords.” (OWASP, phishing.org). In case the system does send messages or notifications, teams are required to assess the security of these messages. The proposed best practices are that the messages should not contain any links to the software, any personal data or sensitive information or attachment files. Additionally, the content should not be based on user input e.g., controlling the subject field or message content from within the system to make it fraudulent should not be possible.

3.4.12 SEC12: Testing and quality assurance

Testing and quality assurance traditionally focus in ensuring the correct functionality of the system, but within the context of security self-assessment the focus is purely on security testing. The teams are required to set up measures for identifying any security issues during testing, which often requires team members to be educated in the use of specific security testing tools. In addition, the team’s post-launch procedures are assessed through three-staged grading: Robust in cases where the teams have procedures in place to log and monitor the software for any unexpected behaviors and a security-engineer to review and evaluate the situation, Weak in cases where something such as a spike in crash rates or other large-scale anomalies appear the team will most likely notice, but there is definite room for improvement, and lastly Nonexistent in cases where the team does not do any type of post-release monitoring.

3.4.13 SEC13: Secure deployment

The Secure deployment section aims to ensure that all used tools within the deployment process of the product are secured. Assessed deployment tools are the source code management systems, any build, continuous integration, and continuous deployment systems as well as any orchestration or deployment services. In addition to individual subcompany specific systems, the corporation does offer company managed services for each section.

3.4.14 SEC14: Infrastructure permissions

Infrastructure permissions section requires the teams to identify any users that have access to databases, storage, queues, service buses or alternative. After the users have been identified, their permissions should be verified to follow the principle of least privilege, as in have only necessary permissions required to fulfil their purpose.

3.4.15 SEC15: Host and network security basics

Host and network security section requires the teams to identify the responsible entities of their host patch management, such as applying operating system patches and updates to their environments, as well as host hardening and configuration such as setting security configurations and removing unnecessary software, as well as network security management such as managing ports and firewalls of the hosting environment. Most often these are managed either by the team internally, by another corporation entity, by an external vendor or by the customer.

3.4.16 SEC16: Security logging

Security logging section focuses on inspecting the logging procedures of the target application. Teams are required to use one of the provided centralized systems that the corporation provides or other standardized services such as those delivered by cloud services. Additionally, the logging capabilities of the system are assessed through a checkbox list containing a set of crucial activities to monitor, such as authentication successes and failures, authorization failures, application errors and system events etc. Teams are also required to be able to provide the logs to customer, user, or authority upon request, which is also addressed in the section DP02: Data Classification, as well as provide a time-estimate on how long it would take them to provide such logs in case of an incident. Other considered topics within this section are ensuring the integrity of the generated logs, monitoring the logs for suspicious activities, and managing the retention of the security logs.

3.4.17 SEC17: Threat intelligence

Threat intelligence is a shallow section focusing in identifying if the application has been onboarded into a cyber threat intelligence monitoring system. Additionally, the teams are required to assess if their internal threat agents such as bribed employees are considered and monitored.

4 VISMA SECURITY SELF-ASSESSMENT ANALYSIS

When conducting this research, it was clear that a generous amount of valuable user experience data as well as potential improvement suggestions could be gathered from the employees that have used the model to assess the security of their software. To collect this data, a survey was created and aimed at personnel that had prior experience working with the model. Approximately 140 eligible respondents, were identified by utilizing the data included in the existing security self-assessments and requested to partake in the study. From these people, a total of 61 responses were recorded. The survey questions were divided into two main categories. The first category aimed at identifying the most useful and least useful sections of the SSA:

- Which areas of the SSA were **most useful** for your team on improving product security? (Select max. 5)
    - What made these sections useful?
- Which areas of the SSA were **least useful** for your team on improving product security? (Select max. 5)
    - What are the biggest problems with the sections you selected, and do you have ideas on how we could improve them?

The second section focused on gathering feedback on a more general level:

- How would you assess the clarity of the instructions of the SSA? (Scale 1 to 5)
- How would you assess the time and effort required to fill out the SSA? (Scale 1 to 5)
- Do you think that the SSA helps to improve the security of your product? (Scale 1 to 5)
- How would you rate the return-of-investment (ROI) of the time invested in SSA to improve product security? (Scale 1 to 5)
- How much of what you learned while completing the SSA can you apply in your daily work? (Scale 1 to 5)
- Do you have any additional comments regarding the SSA? (Free text)
- What do you think is the highest priority thing we should do to improve the SSA right now? (Free text)
- Lastly, would you be willing to attend a short interview session about the SSA

process and future development? (yes/no)

  - Email for contact (Free text)

In addition to conducting the survey, the respondents were given an option to volunteer for an interview session to give more insightful views of their experiences with the SSA. From a total of 21 respondents willing to partake in the interview sessions, 11 were eventually available for a meeting during the four-week period reserved for this activity. The meetings were loosely arranged, and the aim was to gain more insights to the interviewee's experience with the SSA, the challenges they have faced with it. The results from the survey and the transcribed interviews were analyzed utilizing MAXQDA software. From the material, a total of 576 data points were identified and coded, and the used coding system is provided in appendix 1 for closer review. The data from the survey and interviews considering the contents and subsections of Visma SSA are addressed in chapter 4 of the thesis, and the improvement suggestions based on these results are addressed in chapter 5.

4.1 Version comparison

Regarding the received user experience feedback for both, the current iteration of the SSA 2.0, and the previous version SSA 1.0, a quite generous amount of feedback was received. A large deal of the received feedback regarding the version 2.0 was positive, and most often the feedback was feedback comparing it to the prior experiences with the previous version 1.0. The results are presented in table 4, and regarding the version 2.0, the most significant traits that were experienced as a negative were that it was considered less thorough in comparison to the SSA 1.0, and despite being experienced as less exhaustive it was still seen as quite a bureaucratic process. Other less statistically considerable negative traits regarding were that the SSA 2.0 sometimes focuses on documenting things that can be considered "default", as well as it being experienced as too complex to work with. On the other hand, the SSA 1.0 was quite clearly experienced as more problematic than the current version, notably for being even more complex than the current iteration.

Despite the challenges, the interview results presented in table 4 point out that the current version of SSA 2.0 is generally very liked among the respondents when compared to the older version, supported by 28 codes versus only four regarding the SSA 1.0. While most of the feedback for positive user experience is on a general level and does not relate to any particular trait, based on the data it is stated to be clearer than the SSA 1.0, more straight forward, compact, higher level in a positive sense and the inclusion of a new Data protection section was also generally liked. Other noteworthy positive feedback was related to the teamwork and knowledge sharing that the SSA 2.0 induces.

	Positive	**Negative**
SSA 2.0	General positive feedback (13) Trivial sections (1) Clearer (1) Not strictly Visma.net focused (1) Data Protection section (5) Straight forward (3) Compact (3) High-level (1)	Documenting defaults (1) Threat modeling is not thoroughly examined (1) Nonfunctional color-coding (1) Complexity (1) Bureaucratic (3) Less thorough in comparison to SSA 1.0 (4)
SSA 1.0	General positive feedback (2) More thorough (2)	General negative feedback (1) Too detailed (5)

Table 4: SSA 2.0 and 1.0 general user experience feedback

4.2 Template

One specific trend identified from the user experience feedback was that the respondents disliked the platform currently used to provide the assessment template, the Confluence. While there was a total of eight codes specifically reporting bad user experience with the platform, in further discussions it was revealed that there is nothing wrong with the platform, but rather it is the complexity of the form that brings a lot of challenge to filling it. The current template does not support collaborative filling and unfortunately the automatic saving functionality is not entirely functional: from the responses, we received feedback that in some cases following links to external information from within the SSA or uploading an attachment file would force a page reload and erase any unsaved work. Another process which is affected by the template issues is the renewal process, as currently the security engineers are required to create a new page for yearly updates and transfer the data from the previous version to the new one. Unfortunately, some of the structures for inputting data within the SSA do not support simple copy-pasting and make the process needlessly complicated.

4.3 Context

From the survey and interview results, another interesting topic was to identify what different types of applications are assessed with the SSA. From the results, the following set of application types were identified:

- Web applications
- Web application components
- API's
- On premise applications
- New applications (MVP)
- Old applications (legacy)
- Company internal tools

Based on this, we can say with certainty that the SSA is used assess a vast number of different applications, but the interview results also pointed out that the most common challenge for filling out the SSA is it being incompatible with some of the target applications. For example, on-premises applications sections considering hosting or data management are often inapplicable because the responsibility lies within the customer's end, and for API's the lack of user interface rules out a great deal of the SSA. While this is not technically a problem as the sections can be simply avoided, the problem lies in that the SSA does not currently provide an easy method to state that these things do not apply to your software. Another interesting aspect from the data are that the SSA requirements and suggested best practices and procedures are considered an overkill for applications that are in early stages of production, consuming a lot of time in the stage of development where more technical work is on high priority. Additionally, on the other hand when considering old applications with a lot of data and coherent documentation, filling out the SSA becomes a very grueling task as the sheer amount of data makes things a lot more complicated.

4.4 Section 1: Risk assessment and management

Based on the survey and analysis results, the risk assessment and management section got quite a low amount of feedback when compared to the other two sections, as illustrated by figure 3, and assessing each of these sections individually is not meaningful due to the lack of data. From further survey and interview analysis the reason behind the lack of responses started to unravel as particularly many of the interviewees had not actually conducted a risk assessment of their software. In the current SSA, filling out the previous sections usually leads to a list of vulnerabilities and general things to fix. These tickets are all listed as part of the risk review section at the very end of the SSA, after which the teams are asked to list their risk tickets. As these tickets resulting from the SSA are not automatically risks but things that could potentially lead to risks in the application, the process on how to use this information to your advantage and conduct a proper risk review and risk assessment process is considered unclear.

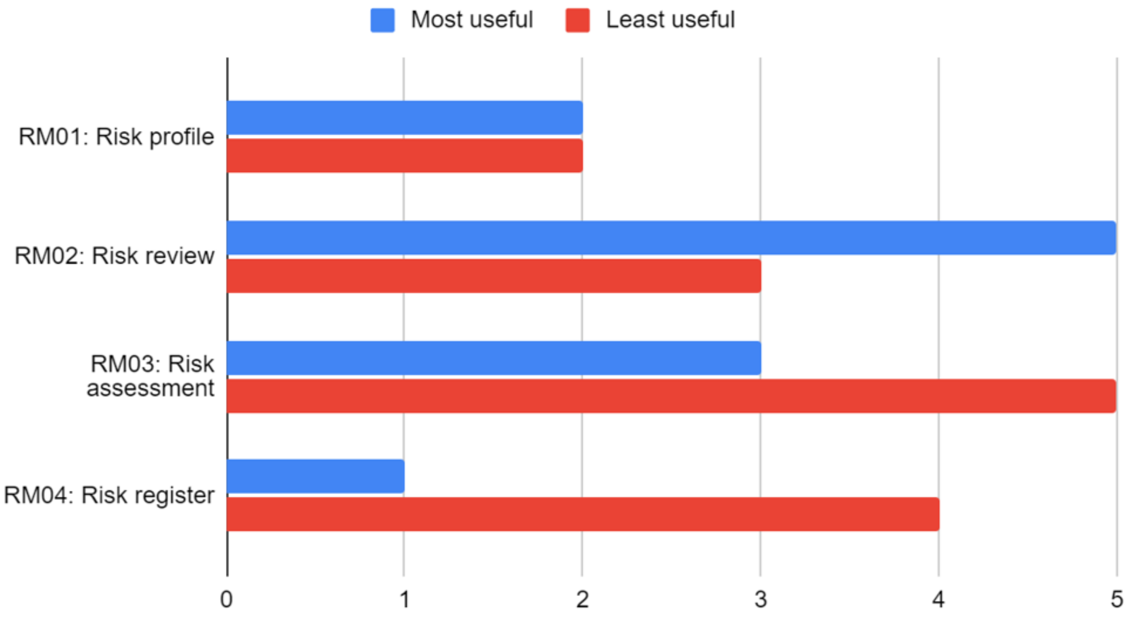

Figure 3: Risk management section usefulness

In the current SSA, the process of conducting the risk assessment is left for the team to conduct as they please. While this does not at first sound like a challenge, it is quite evidently experienced as such since when a common process on how to assess the risks is not provided, it results in uncertainty within the teams on how to approach this section. One of the mentioned methods identified from the responses is that some teams use a risk assessment tool such as the Microsoft Risk Management Tool that was used in the previous version of the SSA, but they also state that the tool is not suitable for all product types and the teams using it had experienced it to be not very user friendly. One of the interviewee's also explained that they use a custom solution for risk assessment which is completely external from this assessment, while more than a few others admitted that they had not really conducted risk assessment at all as they did not know how to approach this topic and do it properly.

4.5 Section 2: Data protection

Based on the survey and analysis results, the data protection section was very mixed with regards to as how useful people feel that it is, illustrated in figure 4. In the prior iteration of the SSA the data protection section was also present, but as an external questionnaire, and the inclusion of this section into the newest version of the SSA also caused a lot of responses from the respondents both with and against the change.

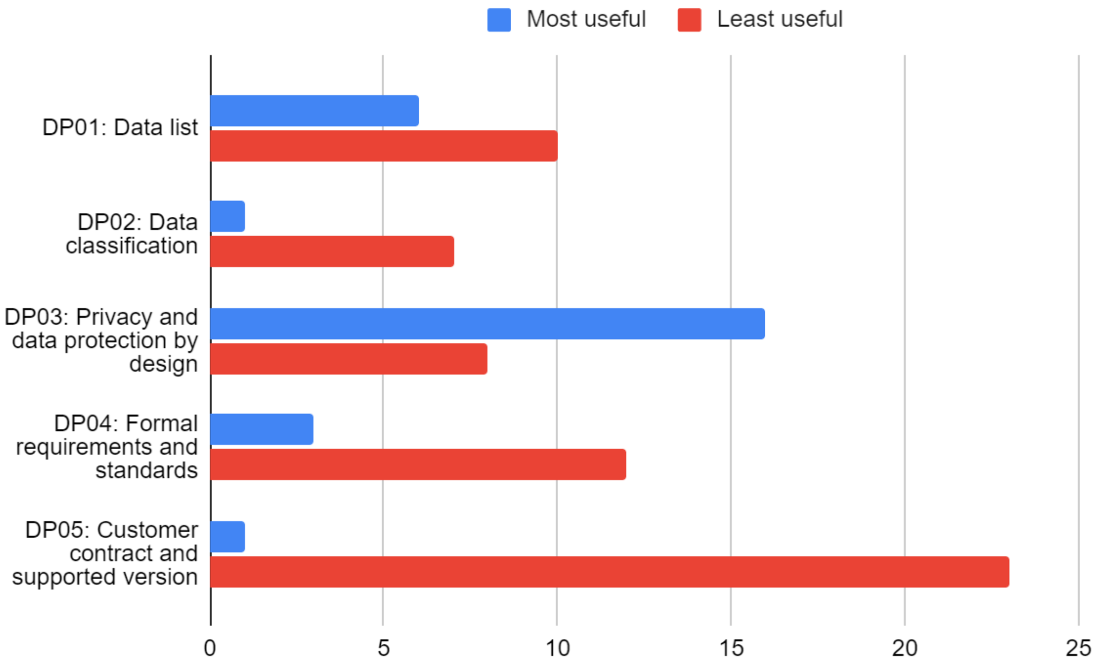

Figure 4: Data protection section usefulness

The strongest arguments from those against were that the data protection does not strictly belong together with information technology or software security and involves another department than development team, as the development team rarely has control over things such as customer contacts and policies. Some respondents also noted that this section does not apply well to their software as some products do not handle personal data. Those supporting the change are happy about the change of perspective that the data protection section brings to them, as it forces them to inspect security from the perspective of data protection and as one of the respondents responded, “keeps us on our toes on what data from our customer we store.” Also, when they had gotten acquainted with this section, based on the feedback they formed a better overview of their entire system. The results and identified challenged and benefits regarding each subsection are addressed in the following chapters.

4.5.1 DP01: Data list

When looking at the survey results on the usefulness of DP01: Data list, they were quite mixed. The positive feedback focused mostly on how the data list brings clarity and overview to the system, increases security awareness and thinking, results in a very deep inspection of the data. The resulting data listing also helps to answer some of the questions in the latter parts of the SSA in addition to being also useful outside of the SSA questionnaire in some cases. The negative feedback for this section was mostly not about people having many problems with filling this section, but with the required accuracy that the listing requires. Going through each column in your database for example and enlisting the columns containing information that may be considered personal often results in a very vast listing, which was also one of the identified points of feedback from the respondents. However, it must also be noted that on the positive note we also got some feedback stating that the strict level of required detail has also helped to uncover some underlying issues within some applications. Another issue with the data listing section is that it is effectively quite fluid section to document, as the data is prone to change as the software is being developed and keeping it up to date or fully revising it every year is quite a daunting task.

4.5.2 DP02: Data classification

Based on the survey and interview data, the DP02: Data classification section appears to be quite disliked but based on the data the main issue with this section instead of how it is conducted appears to be that it inherits the issues of the DP01: Data list section. In cases where the data listing resulted in a massive amount of different data fields, it also reflects directly to the data classification section as when the amount of data requiring classification becomes too vast this section also becomes very exhausting and overwhelming. Also, since the data classification is based on the fluid data list, this section is also very prone to change and will most often require revising yearly.

4.5.3 DP03: Privacy and Data protection by design

Looking at the figures 3, 4 and 5, DP03: Privacy and Data protection by design section appears to be experienced as one of the most useful sections of the SSA. The main positive feedback is related to the section providing a numerous amount of security findings and improvements, which is often reported to be a result of this section presenting a different angle to software security and uncovering various underlying problems in the system design that would otherwise be unthought of. Additionally, this section is reported to raise the general awareness of data protection things of the whole team. Regarding this section, there were not much if any challenges faced with this section at least according to the written feedback. The only identifiable concern is that this section is outside of the core area of expertise for some security engineers, which in turn requires them to put in more time and effort to complete it.

4.5.4 DP04: Formal requirements and standards

Based on the survey results, the DP04: formal requirements and standards section appears to be one of the areas that was experienced as most out of the expertise of the security engineers. In addition to struggling to understand the topic, the general feedback pointed out that the security engineers feel like they cannot affect these things through their daily work and thus this section was experienced as quite frustrating. In some instances, this section would also prove to be quite difficult if the responsibilities for complying with these requirements and standards is not clear to begin with.

4.5.5 DP05: Customer contract and supported version

Lastly, looking at the figure 4, the DP05: Customer contract and supported version section appears to be one of the sections that the security engineers find least useful in the current SSA. Based on further analysis, the key issues that people face with this section are quite like those of DP04: the competence of a security engineer is often not enough to answer these questions without studying the topic further, and they feel that they are unable to effect on this matter. Another identified challenge was that this section may also get overly complex and time-consuming in case there are different contracts for each market area, and in cases of old contracts finding the required information has sometimes proved to be difficult.

4.6 Section 3: Security

The security section is the greatest of all the sections within the SSA, and as such it was the section to provide most feedback in the survey and interview. On a general level, the feedback points that most of the sections are experienced as quite useful, with a few clear exceptions illustrated in figure 5. When looking at the figure more closely, it is to be noted that given the contents of this section, the role of competence regarding the technical knowledge of the target application is emphasized in comparison to the other sections. This is mainly since various sections in the security assessment part require you to have a deep understanding about the software’s internal code as well as the architecture to be able to answer the questions. However, this is often the core area of expertise for security engineers, which is also something that affects the results, for example in comparison to the previously addressed data protection and risk assessment section feedback. The results and identified challenged and benefits regarding each subsection are addressed in the following chapters.

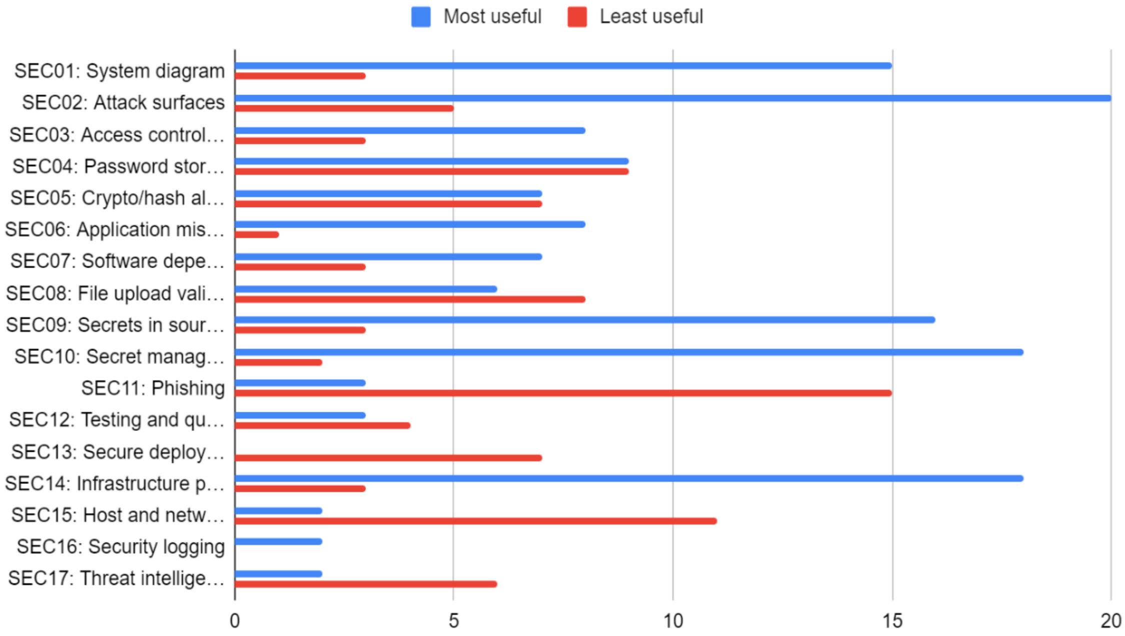

Figure 5: Security section usefulness

4.6.1 SEC01: System diagram

Based on all the figures 3, 4 and 5, the SEC01: System diagram section can be said to have received one of the best results in terms of usefulness. According to the received feedback, this is mainly due to it being perceived as a clear documentation that provides an excellent overview of the system and has also been helpful for the teams in the later stages of the SSA in some instances. The system diagram has also been reported to result in a very thorough inspection of the target system, which in turn has unveiled some underlying security issues within the systems and resulted in improved security. Negative written feedback regarding this section in the survey responses was basically nonexistent, but during the interviews some challenges related to this section was uncovered. The main issue that was identified with this section was that as the process is open, some teams find it difficult to understand exactly what kind of a system diagram is required, as a basic set of instructions given is quite vague, resulting in extra work looking through other team's responses and uncertainty on if the submitted diagram is sufficient.

4.6.2 SEC02: Attack surfaces

Looking at the figure 5, the SEC02: Attack surfaces has very similar excellent score as the previously addressed SEC01: System diagram section. This section was reported to be beneficial to assess the system from an attacker's perspective and raised general awareness within the team. Additional positive effects of this section were that it had also resulted in security findings and improvements, and in some cases to bring structure to the security assessment process and increase security awareness and thinking. Negative feedback regarding this section was generally nonexistent in the survey results, but some minor challenges with this section were uncovered as part of the interview process. Some respondents felt that it is quite difficult to understand what to document and how to thoroughly assess the vulnerabilities of the listed attack surfaces, and some also reported that this section was among the most time-consuming sections simply due to the number of different domains and attack surfaces present in their application. Lastly, it was also mentioned in the responses that filling in this section feels like duplicate work since most of the information required here is overlapping the information located on the VCDM service page.

4.6.3 SEC03: Access control quality

Section SEC03: Access control quality based on the received responses is considered a quite useful section and has resulted in security findings and improvements in multiple cases. It is also one of the sections that was most often mentioned to have increased security awareness and thinking within the development teams. While there was not much feedback related to the challenges faced with this section, in one of the responses this section was reported to be quite difficult to keep up to date since the access control measures in the given context were considered very fluid and prone to change, making it difficult to keep this section up to date.

4.6.4 SEC04: Password storage

Regarding the section SEC04: Password storage, the positive feedback pointed that it had successfully unveiled underlying issues in password handling and gained praise for guiding the teams towards using best practices and company provided solutions. Additional feedback noted that the contents of this section form a good base for later stages of the assessment. Regarding this section, no notable challenges were identified, and the feedback stating this section as useless revealed are based on cases where the software is already using best practices or company provided solutions for managing the passwords, in which case this section is undoubtedly not that useful.

4.6.5 SEC05: Crypto and hash algorithms

Regarding the SEC05: Crypto and hash algorithms section, the main identified benefits were that the section had successfully unveiled potential security issues, and that the section provides direct steps for improvement for these unveiled issues. The main challenges identified were that while it aims to document all the algorithms used throughout the system, they are quite fluid in some instances and prone to change, requiring yearly revision and recursive work with this section. Additionally, in case an application uses numerous different solutions and algorithms, documenting them can easily become overwhelming. Lastly, one of the respondents mentioned that this was the trickiest section for them since the instructions did not define on what level the algorithms should be documented. For example, it was not clear if things such as TLS versions should be included, which resulted in a vast number of required fixes after the initial review causing a considerable delay in the process of getting the SSA approved.

4.6.6 SEC06: Application misuse

When looking at the feedback of SEC06: Application misuse section, while the amount feedback regarding the SEC06: Application misuse section is not very high, it is one of the most positively weighted feedback in the entire survey results. The main identified benefits of this section were that it resulted in security findings and improvements in many cases, and that it provided concrete steps for improving the software security. Based on the lack of negative feedback, it can be said that this section generally does not face any challenges in its current state.

4.6.7 SEC07: Software dependencies

Based on the feedback, the section SEC07: Software dependencies appears to be very functional. The benefits identified for this section were that while it was not often reported to result in security findings and improvements, one of the respondents reported that it had resulted in better procedures and routines regarding monitoring 3rd party libraries. The main identified challenge that this section faces was that that keeping the component library and bill of materials up to date within the SSA is exceedingly difficult, as this topic is very prone to changes. Additionally, teams that were not onboarded to a 3rd party monitoring tool such as the corporation Automated Third-party Vulnerability Service (ATVS) or Software Composition Analysis (SCA) services struggled to provide the required documentation and to assess the security of their software libraries.

4.6.8 SEC08: File upload

Looking at the figure 5, the SEC08: File upload section appears to have quite mixed feedback. Based on the analysis, this section had resulted in notable amount of security findings and improvements when compared to many other sections. Apparently, the main reason behind the amount of negative feedback is that this section is quite often rendered unapplicable, since in many cases the assessed software does not provide file upload functionalities to begin with.

4.6.9 SEC09: Secrets in source code

Regarding SEC09: Secrets in source code, as illustrated by figure 5 it can be said to be among the most beneficial sections within the SSA. Based on the further analysis, the benefits of this section are that it had often resulted in security findings and improvements through cleaning the existing source code for any potential stored secrets. As we did not receive any negative written feedback regarding this section, it is quite safe to say that most security engineers did not face any challenges with this section.

4.6.10 SEC10: Secret management

Looking at the survey and analysis results, the SEC10: Secret management, similarly to the previous SEC09: Secrets in source code section, is also among the most useful sections within the SSA. These two sections are generally quite tightly coupled, and similarly to the SEC09, this section has also similarly resulted in numerous findings and improvements regarding the secret management procedures and tools and was also reported to result in better documentation. The key issue that this section faced was that it is one of the sections that was also considered to be very fluid and thus difficult to keep up to date. Additionally, one of the respondents claimed that it was one of the most time-consuming sections of the SSA. This was due to the required remediation of the application not using the best practices approach proving to require an extensive amount of work while implementing the secret management system into an older application.

4.6.11 SEC11: Phishing

Looking at the responses presented in the figure 5, the SEC11: Phishing section appears to be perceived as one of the least useful sections within the SSA. The benefits of this section were that in some cases it had resulted in security findings and improvements, but the amount of feedback remained quite low. However, the main reason behind the poor score of this section is again not a result of the section's contents, but rather that the phishing section does not apply to various assessed software, and therefore being considered quite useless in many cases.

4.6.12 SEC12: Testing and quality assurance

Regarding section SEC12: Testing and quality assurance, the general amount of feedback was quite low and mixed, and while in one instance it was reported to have resulted in improvements in quality assurance procedures and routines, the general benefits and challenges regarding this section could not be accurately identified from the data.

4.6.13 SEC13: Secure deployment

SEC13: Secure deployment did not receive any feedback regarding being among the most useful sections within the SSA, so addressing the benefits of this section is quite difficult. Regarding the identified challenges, this section was considered a duplicate of a similar section in another Visma Cloud Delivery Model (VCDM) assessment, and additionally considered to be quite fluid in its contents making it difficult to keep the contents up to date.

4.6.14 SEC14: Infrastructure permissions

SEC14: Infrastructure permissions appears to be among the most useful sections within the current SSA based on the survey results. It received one of the highest reported results for resulting in security findings and improvements through thoroughly inspecting the existing permissions and tightening the least-privilege policies. The only challenge identified with this section was that it also suffers from trying to document things that are prone to change, but generally that this section was considered highly functional.

4.6.15 SEC15: Host and network security

Feedback regarding challenges in SEC15: Host and network security is one of the sections that stands out from figure 5 by being one of the sections considered least useful in the security section. The reason behind this is based on the interviews is that this section is inapplicable to any on-premises application, and even in various cases of online applications the applications are hosted and managed by a 3rd party. Additionally, one feedback from a team that is self-managing their production environment stated that the section does not provide clear enough remediation instructions and best practices for ensuring the security of their development.

4.6.16 SEC16: Security logging

SEC16: Security logging received an exceptionally low amount of feedback in the survey, and while one of the responses reported that it had resulted in security findings and improvements in the assessed application, the general benefits and challenges regarding this section cannot be accurately addressed based on the gathered data.

4.6.17 SEC17: Threat intelligence

While being a very quick-to-complete and shallow section, according to the survey and interview analysis the SEC17: Threat intelligence section is experienced to be not that useful. While one of the benefits of this section was mentioned to be that it promotes the teams to use the corporation provided service, it is already enforced as a part of the Visma Application Security Program. This reflects onto the amount of negative feedback, since when the information regarding the onboarding status is already available and tracked in the index, asking it again within the SSA feels unnecessary. Additionally, one of the respondents argued that conducting a proper consideration of internal attackers is something that would be better covered if included in the process of threat modelling as opposed to simply ticking two boxes in this section stating that your service is onboarded to the threat intelligence and that you have considered internal threat actors.

4.7 General challenges

When looking at the larger picture, one of the most impactful fundamental challenges related to the SSA is the issue of compatibility, which was previously addressed in chapter 4.1. Additional identified general challenges with the SSA based on the results are general time consumption, the required competence, clarity of instructions, motivating teams to carry out fixes, documenting fluid things and lastly in this case the assessment renewal process.

One of the covered general topics through the survey was assessing how the security engineers personally experience the time and effort required to fill the SSA, which leaned quite heavily towards it being very tedious and time consuming to work with as illustrated in the figure 6. This is quite an interesting finding considering that the SSA 2.0 is already considerably more lightweight than its predecessor. When looking more closely at the data from survey and interview analysis it is important to note that the first-time assessment consumes the most time and effort as expected.

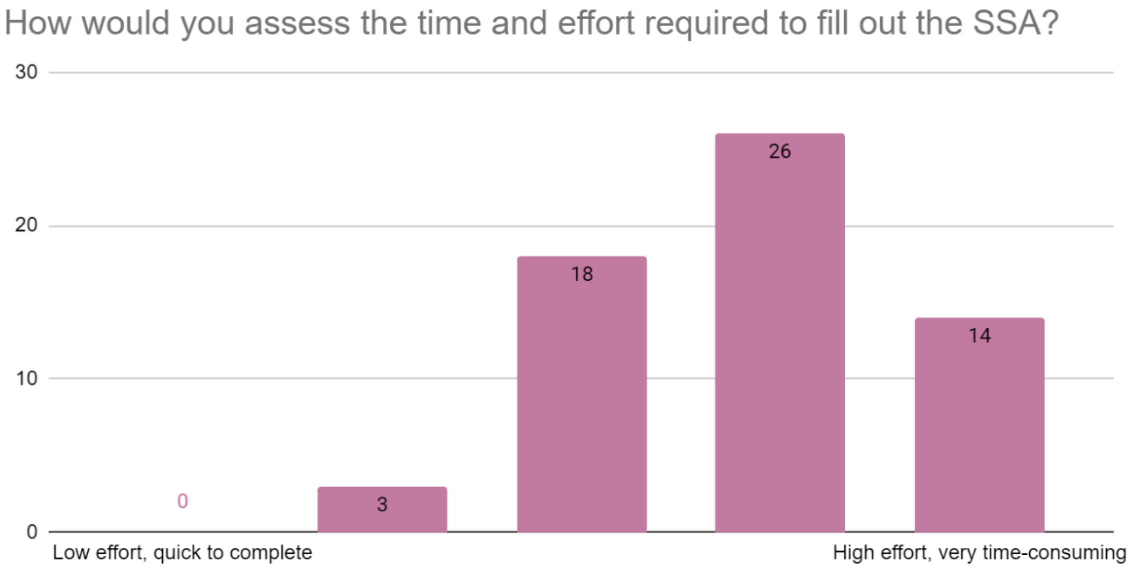

Figure 6: Required time and effort

The reasons behind this experience are a multitude of themes, some of which we have already covered. The time consumption and effort of filling out the SSA is dependent on the pre-requisites and skills of an individual security engineer, as well as the target application. When considering the individual skills, the areas that the security engineer should be very proficient at are for example legal requirements, data protection, software architecture and the domain of the software, which already is quite an extensive amount of knowledge areas for an individual to master. The level of competence tends to reflect onto the quality of the responses, as generally all the responses to the SSA and their correctness are based on the security engineer's knowledge, and in some cases straight out on their best judgment. During the interviews, some of the security engineers mentioned that they often gathering information from more proficient people for some questions, but some mentioned that sometimes it is still up to their interpretation on how to respond to a question and could result in a "broken phone" type of communication and answers. Some of the security engineers mentioned in the interviews that when they face a section that they are not familiar with, it first takes them a lot of time to comprehend the topic and what the question is about, after which it takes them often even more time to find answers to the questions. While this is not directly a result of unclear instructions, it is tied to the topic as the level of provided instructions and additional sources of information are key factors in easing the learning curve.

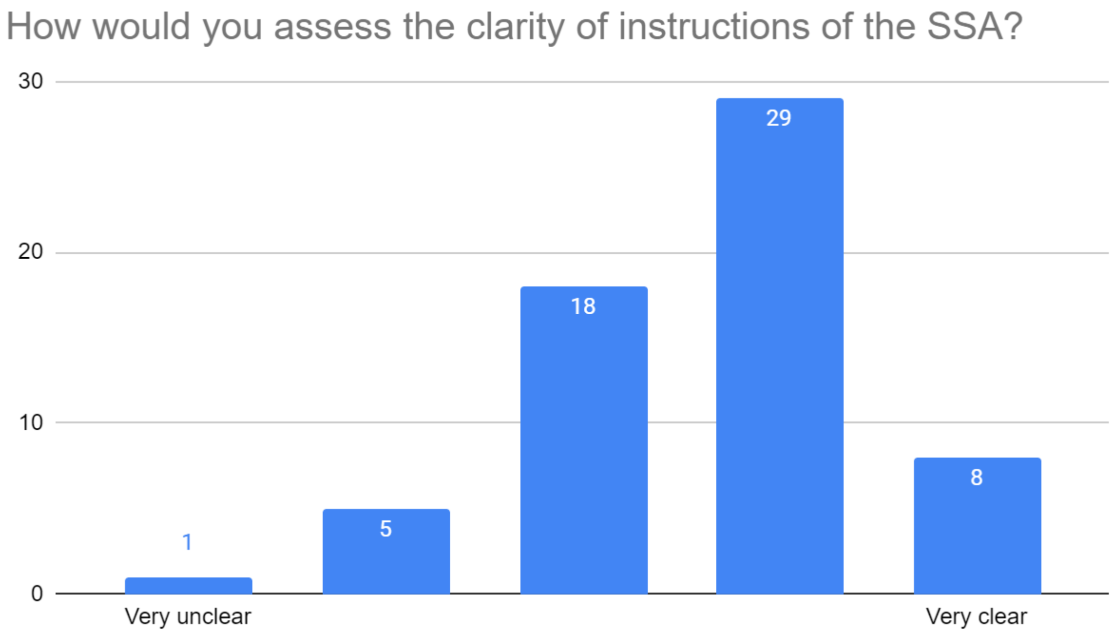

Figure 7: Clarity of instructions

As illustrated in figure 7, the instructions included are generally seen as quite clear, but it can be said with confidence that there is room for improvement as they are essentially the base for knowledge for the security engineers working with the SSA. Based on further survey and interview analysis, one of the main identified issues with the instructions is that people struggle at some points of the SSA due to not being able to fully comprehend what kind of an answer is expected to a section or a question. A concrete example of this kind of issue was mentioned to be in the section SEC05: crypto and hash algorithms, which requires enlisting all used encryption protocols in the transport layer such as HTTPS and TLS, which cannot be exactly declared based on the section instructions. While this is not a severe problem as the requirements will be cleared out eventually in the review process, it may take upwards of three weeks to get a review on your SSA in the current state and as such this kind of errors cause a considerable amount of delay to the approval.

Another main aspect regarding the topic of time-consumption and effort that was mentioned in section 4.2 is the type of the target application. Not all the sections within the SSA are applicable to all applications in a comparable manner. For example, applications that do not, for example, process any personal data are not that obliged by the data protection section. Another aspect in addition to the application type is the volume and lifecycle of an application. The analysis revealed a trend where the data protection section due to its vast requirements of GDPR knowledge and the creation of a data list, as well as the system and attack surface diagrams are among the most common sections to consume a lot of time and effort, which is especially true for applications with vast number of different types of data. As for software that is at a later stage of the life cycle, chances are that they contain legacy code and the amount of available documentation to support the process of filling out the SSA simply is not up to par or does not exist, increasing the required effort even more.

Outside the process of filling out the SSA, an aspect that consumes time and effort are the required fixes and security improvements resulting from the SSA. While most of the time the fixes are considered quite straight forward and easy to approach, some require more fundamental changes in cases where the software is not using the suggested best practices. An example of a section resulting in a very tedious repair process was mentioned to be the secret management section, as in some cases implementing a proper secret management solution into an application with vast number of secrets took upwards of a few months to complete.

Another surprising finding was that in addition to the feedback regarding the first time of filling out the SSA, the general feedback regarding the yearly revision was also quite negative, indicating that it is also perceived as a surprisingly tedious and time-consuming process. The key issues with this were related to the way the process is handled as of now, as every year a new SSA is created and the contents of the old SSA must be imported to the new one. This is entirely manual work and some of the newly introduced input methods are unfortunately unsuitable for easy importing as they effectively prevent simple copy-pasting.

4.8 Benefits

Regarding the benefits of filling the SSA, a general baseline was gathered through the survey where the security engineers were asked to assess if the SSA helps to improve the security of their product. As illustrated in figure 8, the results lean quite heavily to the positive side, which is also backed by the analysis data. An interesting discovery is also that the figure appears to have a very similar distribution to the Figure 8: required time and effort, and it could be argued that these graphs correlate with each other: the more time and effort is required to fill the SSA and carry out the required fixes, the more effect it has on the product security.

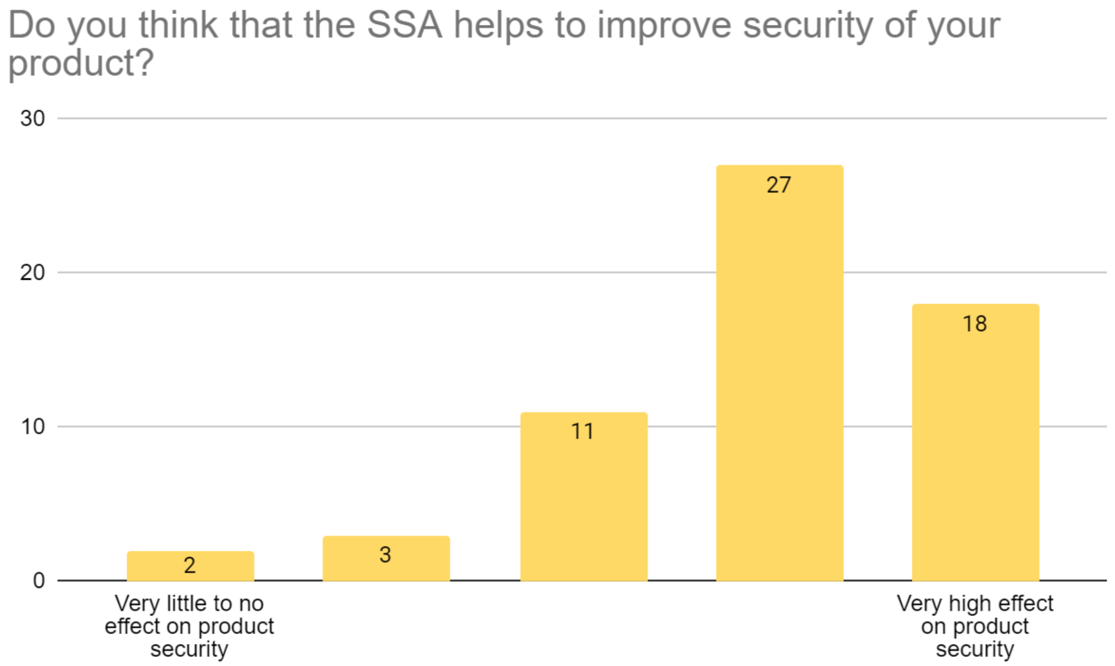

Figure 8: SSA improves the security of a product

Among the identified product security improvements and benefits, the most noted was that the SSA had resulted in security findings and improvements, which were specifically mentioned in 20 individual survey results and four interviews. Another major identified benefit was that the SSA reportedly increased security awareness and critical thinking within the teams, which was specifically mentioned in 9 surveys and two interviews. Other identified benefits from among the survey analysis were that the SSA brought structure to the security assessment process, provided concrete steps for improving security, resulted in thorough inspection and better overall documentation, and in some cases resulted in improved security procedures and routines. Additionally, the SSA was also reported to be beneficial as it can be referred to in additional audits, and that in some cases the most valuable result of the SSA were the discussions with the SSA assessment team following the process of filling the SSA. Technically most of these things are linked to each other: Firstly, the SSA provides the teams a way to systematically focus on the key areas of security related to their product, during which it helps to identify potential underlying vulnerabilities and risks from the software, which are then transformed into concrete tasks for improving security and later organized as part of the remediation process, which again raises the general level of security awareness throughout the whole development team.

While being the second most common benefit from the SSA, raising security awareness and thinking is as a topic quite closely tied to if the security engineers feel like they can adopt things they had to learn while filling the SSA into their daily work. To my surprise, out of all the generated figures, figure 9 is the most left shifted one, with 12 out of the 61 respondents leaning towards not being able to apply much if anything of what they learned to their daily work. However, this is still quite clearly leaned towards the positive side and the analysis data unfortunately did not shed any light on this graph being weighted to the left side and the reasons behind this are still unclear. We can only assume that some of the security engineers responding to this survey were already quite familiar with the covered topics so that they do not necessarily learn any new meaningful things from filling out the SSA.

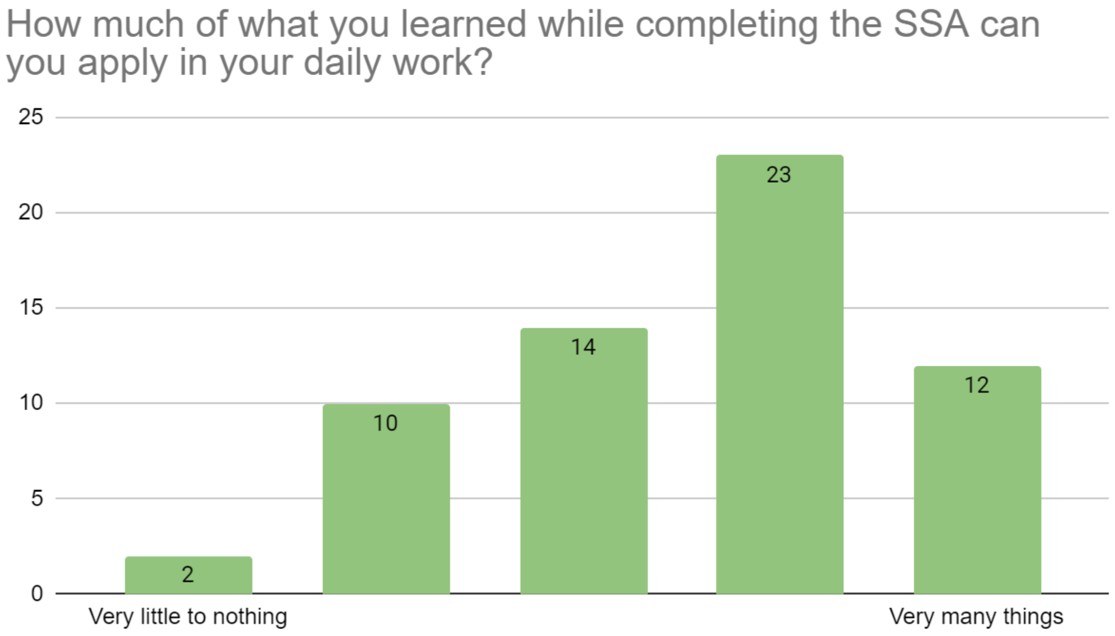

Figure 9: Can you apply your learnings in daily work?

Altogether, when looking at figure 10 depicting the return of investment, it can be said that despite the SSA being experienced as a quite time-consuming and tedious process, it is still mostly considered worthwhile of doing due to the benefits that it produces to the teams. One of the respondents stated that “Regardless of the time and effort spent doing the SSA, it is absolutely valuable and provides us with a better product in the end.”, which effectively describes quite accurately the general opinion regarding the return of investment.

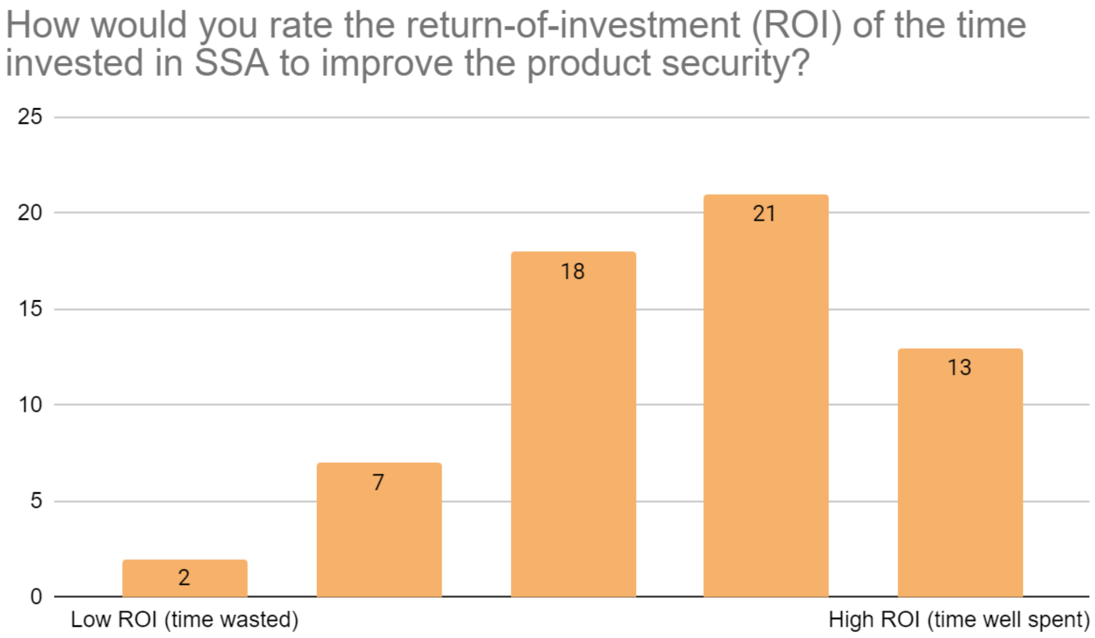

Figure 10: Return of investment

5 VISMA SECURITY SELF ASSESSMENT IMPROVEMENT SUGGESTIONS

While the general feedback regarding the return of investment is positive, in the light of identified challenges there is evidently a considerable amount of challenges to overcome and things to improve in the current model. To utilize the ideas and opinions of the personnel working with the model, the respondents were asked to provide their improvement suggestions regarding the model, and the topic was further discussed with some of the respondents during the interview sessions. From this data and further analysis of the challenges identified, a set of improvement suggestions was then generated. While some of these improvement suggestions did show some statistical grounds throughout the responses, in the following chapter the improvement suggestions are not heavily weighted based on this, but rather documented as all of them are based on a real experience by someone that has worked closely with the assessment, and both faced and overcome some of the identified challenges. Additionally, the decision on which of these improvements to implement and how they can be technically achieved is something that the authorities behind developing the SSA can discuss and decide on, and is further discussed in section 6.1. The following improvement suggestions were gathered from the data, and are addressed in the following sections of this chapter:

- **Include Visma Cloud Development Model approved templates**
- **Reduce general bureaucracy**
- **Make the SSA more lightweight**
- **Add JIRA-guidance**
- **Improve the current template**
    - Ensure that the macros work
    - Ease the renewal process
    - Change the platform
- **Emphasize knowledge sharing**
- **Improve the current process**
    - Change the risk management procedure in general
    - Reduce the review time

  - Organize an SSA introduction session to security engineers
  - Emphasize delegation and collaboration
  - Organize a meeting with the company security team and the development team prior to filling the SSA
  - Emphasize the honesty in filling the SSA
  - Involve developers in the renewal process
  - Involve system architects in the process
- **Make the SSA adaptive**
- **Structural improvement suggestions**
  - Section specific improvement suggestions
    - Introduce an infrastructure checklist
    - Identify and reduce overlapping sections, particularly with other Visma Cloud Delivery Model assessments
    - Detach the Data Protection section
    - Start with the Security section
    - Introduce new sections
      - Network overview
      - Infrastructure security
    - Improve section grouping
      - Based on similarity
      - Based on required effort
    - Data protection section improvements
      - Focus on logical grouping of data rather than listing all available data
      - Split the DP03: Privacy and data protection by design to smaller subsections
  - Question and instruction improvement suggestions
    - Include code checklists
    - Focus on things that are not automatically handled by frameworks / reduce documenting obvious things
    -
    - Add links to external information

            - Improve instructions
                - SEC14: Infrastructure permissions
                - SEC10: Secret Management
                - SEC01: System Diagram
            - Include reasoning for questions
            - Include links to existing company guidelines
            - Make the questions less ambiguous
            - Include example answers
                - Utilize the existing data to create these examples
            - Include predefined answers
- **Automation / continuous assessment**
    - Improve 3rd party attribution and licenses checking
    - Utilize automated data gathering
    - Include endpoint scanning
    - Utilize automation to address fluid sections
    - Introduce Infrastructure as Code scanning

5.1 General structural improvements

When considering the general structure of the SSA, despite the SSA being much lighter than its predecessor, some respondents still wish for it to be even more light weighted and to reduce the general bureaucracy. Achieving this is tricky without compromising the functionality of the SSA, but what can be done is to make the SSA more lightweight by reducing the amount of content within the SSA that is not applicable to the target application through making the SSA more adaptive, which is discussed in further detail in section 5.10.

5.2 Questions and instructions

Regarding the questions and instructions, the most desired improvement was to make the questions less ambiguous, which was explicitly mentioned in three interviews and four survey results. The rest of the suggested improvements are tightly coupled with this issue, as the respondent wish for better and more in-depth instructions for answering the questions including also links to external sources for deeper knowledge, example answers on how these questions should be understood, as well as predefined answers so that they can just select the suitable option from the dropdown and provide additional information only in case none of the alternatives are fit for them. They also pointed out that to be able to generate a set of example answers, the team behind developing the SSA can utilize the existing data from the SSA's. Additional thoughts were that there should always be reasoning included with the questions, or more closely to define why certain things are asked for and documented with examples of potential security issues caused by misconduct. This would help the teams, especially in cases where they find it hard to motivate the developers to make improvements, as it would give them the ability to better justify the need for improvements.

Regarding the instructions, the most identified desire was to improve the overall level of detailedness with the instructions and to include them for the currently open processes, such as designing the system diagram and conducting the risk review. Additionally, there was a general desire for the instructions to provide links to additional sources of information, such as OWASP listings, different types of code cheat sheets and corporation internal guidelines to better support the security engineers in familiarizing themselves with the addressed topics.

Lastly, an interesting improvement suggestion regarding this topic was to reduce the focus on the more obvious things. The reasoning behind this is that one of the interviewees explained that currently the SSA includes a lot of obvious things which are automatically managed by frameworks and things that should be obvious to developers. Documenting these kinds of obvious things quickly becomes tedious and the parts of the SSA that require attention and focus, and real problems are cluttered by documenting these things that should be automatically checked for by the developers or automatically handled by certain frameworks.

5.3 Data protection section improvements

From the data it was also evident that the sections are one of the most discussed areas for improvement suggestions. One of the more interesting things based on the data we can see is that some of the respondents are against the inclusion of the data protection section in the SSA and would prefer it to be detached into its own separate assessment. However, this topic is quite contradictory, as generally during the interviews it was quite evident that the data protection section is also quite generally liked, making this the most controversial topic of discussion related to the assessment. Detaching the data protection section entirely from the security self-assessment could provide some solutions to the two identified major challenges, lack of competence and the inability to affect things, as these are both related to the security engineer's and if detached the responsibility for filling the data protection section could be moved to someone more suited for the task, such as a product owner for example. A downside of detaching the data protection section is that many of the respondents also claimed that completing the section improved their general understanding of their system and helped them to assess the software security from another perspective, which could easily be missed in case the data protection section is detached from the SSA.

Regarding the contents of the data protection section, one of the most notable challenges for the teams, particularly in case of a large application, was gathering an accurate data list. A concrete improvement suggestion to try and make DP01: Data list more lightweight was proposed by an interviewee and suggests shifting the focus of the data list to identifying and addressing logical grouping of sensitive data, rather than focusing on listing all separate sensitive data types that you have in your system. For example, in the case of a payslip application, utilizing this kind of approach could work as follows: fist, you generalize that your application stores payslip data. Based on this generalization you should proceed to group any sensitive data that the payslip data includes, after which you proceed to classifying the identified data group and possibly separating the locations used for storing it and rethinking your process of where the data needs to be available within the application. If utilizing the current approach, the team should conduct a deep inspection of their system and go through every database field, after which all the identified data would be classified and assessed separately. Despite the increased effort in the latter, the outcome of this section and the required changes related to ensuring protection of the personal data would most likely be quite similar.

One of the more interesting, yet statistically less noteworthy improvement suggestions was to break the largest current section within the SSA, the DP03: Privacy and data protection by design, into smaller subsections. As of now, the section is quite vast and includes 30 different questions, which are organized by topics already. While the section already has a subsection system for the addressed topics and questions since they are grouped within the section, it should still be considered if these sections could be further detached into their own separate sections to make it appear more approachable.

5.4 Risk management section improvements

While there was only a single survey response specifically naming the risk management procedure as something that should be redesigned, it is clear based on the survey data that this is one of the most problematic areas in the current SSA and should be revised. The main identified issue with this section based on the survey data is the lack of instructions, so that it is not evident for the security engineers on how to approach this section. As an improvement suggestion, a common procedure for conducting the risk assessment should be introduced to the teams that they can follow in case they do not have a specific method or process established for the purpose. When creating the instructions, it would be beneficial to further interview the teams that have conducted the risk management process on some level and try to generate the model based on existing functional processes used by the teams.

5.5 Software security section improvements

Regarding the software security sections, the identified issues were mostly related to fundamental issues with the SSA: compatibility with different applications and documenting fluid things. As such, addressing these topics is the suggested improvement for making software security section more meaningful, and the suggested solutions are making the SSA more adaptive, and emphasizing automation to reduce the amount of manual work required to document fluid content sections. However, as these topics are more on the general level, they are addressed individually in sections 4.4.10 and 4.4.11.

5.6 New sections

Other than the discussion about existing sections within the SSA, from the responses a few suggestions for new sections were identified: cloud and infrastructure security section and network overview section. Cloud and infrastructure security were both individually named, but on a closer analysis they are focusing on the same target: making sure that the infrastructure of the software is secure. To perform this kind of assessment, it was suggested that in case the service is using a cloud service, then automated tools such as infrastructure as code scanners should be utilized to manage the process automatically. Additionally, in cases the service does not have infrastructure as code, it was suggested to approach this section similarly to DP03 by providing a simple checklist that would guide you to inspect the critical areas regarding the infrastructure. Another of the suggested new sections is a network overview section. This would act as a visualization on how various products are linked together and to better be able to see and understand how the data flows between them to determine how can system data leak outside and how any breached component or integration would affect the overall security.

5.7 Section grouping

Another identified improvement suggestion category was section grouping. The respondents wished for the sections to be grouped together more logically and based on section similarity. For example, the SEC04: Password storage, SEC05: Crypto and hash algorithms, and SEC09: Secrets in source code and SEC10: Secret management are effectively things that are all related together since they focus on defining how delicate information is stored and secured within the application but are still scattered within the SSA resulting in some "jumping back and forth" as stated in one of the interviews. A similar situation was identified also with SEC07: Software Dependencies and DP04: Formal requirements and standards, since the intellectual property guidelines require you to also understand the 3rd party software components that your application uses.

5.8 Reducing overlap

Reducing overlap was an interesting improvement suggestion, as the general feedback regarding this was not that there were overlapping sections directly within the SSA, but that some sections within the SSA are overlapping with sections from other assessments and documentation conducted and created as a part of the VCDM and VASP process. As an example, things such as a brief service description and an assumption of the number of users for the application is required in multiple occurrences, and instead of documenting the same things in different form to all different documentation it could be unified. This is something that should be verified and examined, since if we can utilize existing data from sources other than the SSA to fill it, or the other way around and port data from the SSA to them, it would reduce the amount of duplicate work.

5.9 Process

Regarding the process, the most outstanding improvement suggestion was to arrange a meeting with the security team prior to filling out the SSA, and some of the respondents also wished for the team to conduct a small audit prior to the meeting. The benefits of such a meeting would be that it would enable the teams to include the personnel required to fill out the SSA prior to starting the work. In the meeting, they could further go through the SSA with the guidance of the security team and agree on collaboration and delegation, as well as go through any open questions with the security team such as discuss the compatibility issues and agree on the required scope.

Another topic that was already mentioned is that considering the process of filling out the SSA, it should be more clearly emphasized that the responsibility is shared. In addition to the security engineers a lot of people are involved with the process of filling out the SSA, such as developers, software architects, data protection managers and marketing department personnel, all with their personal area of expertise. Currently, the security engineers often consider the SSA to be on their personal responsibility rather than being a team effort for all the people working with the product, and as such emphasizing collaboration and even delegating sections to people proficient at those sections eases the pressure of an individual security engineer and in turn will also improve the quality and correctness of the responses as well as the general level of security awareness.

5.10 Adaptivity

One of the most impactful challenges related to the SSA is compatibility with several types of applications and improving adaptivity of the SSA is seen as the solution to this issue. The general desire is that the contents of the SSA would be updated based on the type of application that you are trying to assess so that instead of having to answer every question. This would be achieved by defining the type of application at the beginning and the SSA, after which the template would automatically adjust the contents to include things that are relevant and meaningful for target application. In addition to adapting the contents based on the software type, it was also stated that each individual section should also be able to adapt to a situation where the section is inapplicable to you for any reason.

5.11 Automation

To ease the process of filling out the SSA one of the most trending suggestions was to implement automation into the assessment, as some of the data required in these sections can be either collected by an automated tool such as a SAST or DAST tool, or already have been collected and documented as part of the VCDM onboarding process. Creating simple tools to fetch this information from other documentation sources or tools might not be that difficult to achieve and it would also simplify keeping the constantly changing data sections up to date. Of course, utilizing automation will raise the possibility of the fetched information being invalid, and therefore it should also be noted that the responsibility for ensuring the validity of the data should still be on the responsibility of the security engineers rather than blindly on the automation.

5.12 Template

While the gathered user feedback points towards the template issues being not strictly bound to the platform, it is quite evident that the platform does play a significant role in the way the SSA is currently established. While most of the identified issues addressed in chapter 4.2, such as collaborative filling and improving the automatic saving functionalities can be fixed within the current platform by ensuring that all the template macros and settings are revised and configured properly, the general feedback leaned towards simply changing the platform to something that is better suited for hosting the SSA template. Transferring the SSA to another environment could potentially solve all the identified issues natively, and potentially provide a better environment for future improvements regarding the SSA.

6 DISCUSSIONS

This study has provided a general baseline understanding of the current model: the fundamental processes it is based on, the actual contents of the model as well as an insight into the experiences of the security engineers using the model. Additionally, the challenges in the current model are examined and the improvement suggestions from the feedback are included, which can be utilized to further develop the current SSA to be a more ideal version of itself. What this also enables us to do is to be able to start to look at the bigger picture and think about the role of SSA more thoroughly.

6.1 Improvement suggestions proceedings

After the initial model analysis in this thesis was completed, the results along with the specific improvement suggestions were discussed with the team behind developing the SSA. The team has recently been working with developing version 2.1 of the SSA. The next version's contents are similar with the contents of SSA 2.0, and the main change will be in the platform in which the SSA resides which will automatically solve a great deal of identified usability issues linked to the current platform, and additionally the data protection section will be separated from the security assessment. Both changes are backed up by the data of this research, which gives the team an added level of certainty that the development is going in the right direction. While going through the collected improvement ideas, it was also quite evident that a lot of the presented improvement suggestions were something that the team members had already considered at some point.

Regarding the risk management section improvements, changing the risk management procedure was considered difficult, as generating a general set of instructions that would suit every situation is a daunting task. What was agreed is that the instructions for the risk management section should be generally improved, and it should be emphasized that the risk tickets should not be duplicates of the tickets resulting in from the SSA; but rather tickets describing all other things the team is aware of and which pose a potential security risk.

When discussing about the changes to the data protection section, in addition to detaching it from the SSA, it was also mentioned that that the responsibility for filling it is planned to be transferred from the security engineer to the product owner. The change of platform covers the proposed suggestion to include subsections for the DP03: Privacy and data protection by design section, and the last covered topic was to discuss the current state of conducting the DP01: Data list. It was identified that the current idea that all the data within the system must be listed in detail is a remnant from the past, and it was agreed to be something that the team should discuss further in the future and rethink entirely how this section is carried out.

Regarding the security section improvements, it was agreed that the sections that had been identified to have a potential for improvement in the instructions, SEC01 System diagram, SEC10: Secret management and SEC14: Infrastructure permissions should be revised, as improving the instructions is seemingly an easy task which can be conducted as part of the transition from version 2.0 to 2.1. The suggestion to improve additional sections such as the network overview section or the cloud and infrastructure security section was also addressed, but the main issue with including these kinds of sections was that it should be further discussed if they should be included in the scope of the SSA. While they could be beneficial and something that could be added to the SSA, it was agreed that at this stage these will not be included in version 2.1 at least and should be further considered in the future. Additional discussions related to security section were the proposed changes to grouping some of the sections closer to each other based on their similarity in the addressed topics: the SEC04: Password storage, SEC05: Crypto and hash algorithms and SEC09: Secrets in source code and SEC10: Secret management, which was considered a reasonable improvement to carry out.

Regarding rest of the proposed changes, some of the identified improvement suggestions, such as improving and clarifying instructions, adding external links to other sources, or improving the instructions in general to be more easily consumable were considered “low hanging fruits”, and initially planned to be added into the 2.1 version prior to its release. The more fundamental discussions about improving the adaptivity of the SSA, as well as utilizing automation and existing data to improve efficiency were also discussed and considered to be possible as the new platform provides capabilities for displaying adaptive content, but it was agreed that all these more thorough changes to the SSA will be included in the version 3.0 later in the future. The general discussion also considered some of the challenges of utilizing automation as the data should still be verified by a human in all cases, and the problem of accurately defining and selecting the correct product type to adapt the contents of the SSA accordingly.

Other noteworthy outcomes emerging from this discussion session was that to improve the general perception of the SSA as a process, the aim should be to improve the overall image of the SSA in general. A proposed approach to this could be to encourage development teams to share their success stories with the SSA to other teams, and to better illustrate and emphasize all the benefits of the SSA on a larger scale. Additionally, to ease the process of filling out the SSA and improve knowledge sharing, a proposition to establish a companywide SSA discussion channel was presented. The channel should generally be open to any employee within the organization to ask any questions related to the SSA, and act as a channel of communication between the users of the SSA and the development team, as well as a general channel for asking for help.

6.2 Comparison with industry models and standards

Comparing the Visma SSA to the industry standards and other similar tools is quite difficult. This is mainly since the SSA does not aim to assess the compatibility with a set standard or framework, such as such as the PCI DSS, CSA CAIQ, and OWASP SAMM, while similar internal security assessment tools used in the industry by other software companies are generally not publicly available. An insight to the industry's current best practices and contents is reflected by the BSIMM report, but in comparison to the Visma's SSA the focus of the BSIMM is arguably broader on the covered topics, and comparing the procedures and practices presented in the BSIMM would be more meaningful to compare with the entire contents of the VASP, rather than just the security self-assessment component.

When comparing the topics addressed by the Visma SSA to those of similar tools, such as the CAIQ and SAMM presented in section 2.2, they appear to be very similar. One of the more notable differences in comparison, particularly with the CAIQ, is that the Visma SSA is considerably lighter to complete. For example, the CAIQ addresses human resource management, and cloud and infrastructure security on a much deeper level, while the Visma SSA covers them on a very high level through only a few questions. During the improvement suggestions proceedings meeting, this revealed to be a conscious choice: while the inclusion of a new section for cloud and infrastructure security, addressed in section 5.6. was discussed, the response was that generally the focus of the SSA is intentionally kept more strictly on the topic of software security, and that the strict infrastructure security is currently a little bit out of scope for the SSA. The question raised by this is that that since there are no key differences in the covered topics but rather the depth on which things are addressed, can it be said that addressing and assuring the security of a software using for example CAIQ in comparison to the Visma SSA would result in more secure software? For this question, a thorough answer cannot be formed based on this research, and if we consider that according to BSIMM the software security should be addressed through the measures companies take to assess and ensure security of their products (Jaatun, 2012; McGraw et al., 2015), it is obvious that the SSA itself is already a good measure to improve security that could most likely be utilized in even more companies.

6.3 Fundamental challenges and proposed solutions

When inspecting thinking about the SSA on a higher level, the fundamental identified issues came down to the living nature and heterogeneity of the assessed software. In its current form, the SSA aims to document a lot of variables, which effectively have two extremes: some things documented are things that change so rarely that they can technically be considered one-time setups. On the other end we have things that are very prone to change on a daily or weekly basis. Fundamentally thinking, the SSA is effectively a snapshot of the software in each given time, but this raises a concern: When conducting the SSA for the first time, it is the most tedious to work with but also the most rewarding. However, when doing it again, the onetime setups most likely have not changed, and the more fluid sections have almost certainly changed. But when the fluid things change, it is important to be agile and react to the changes also from a software security perspective, rather than wait for another year when you are redoing the SSA. As a suggestion for this fundamental issue, the SSA contents could be split based on their tendency to change. The content that is not prone to change very often could be included in a dedicated assessment or a section and would effectively require less monitoring and should be re-assessed on a yearly basis. The other section should contain the fluid things, and be revised more often, either on a more frequent basis or based on any need for revision emerging from the changes in the software. In its current state, we could think that the value of the SSA is equivalent to a checklist: after initially ticking every box, the value is somewhat diminished unless there are changes in the context. But in a way this is what the SSA as a tool was designed to be: a very extensive checklist on critical security things, that you should examine, and fix related to your product to achieve a certain level of base security. Further studying the effects of SSA regarding the security of a product in the long term, as well as the entire process of conducting the SSA is still an open question that would require further studying to fully understand.

Regarding software heterogeneity, addressing the issue of compatibility by making the SSA more adaptive is eventually a quite straightforward solution to execute. The other identified issue, the issue of complexity of the assessed software is much more challenging to solve, as it undoubtedly has a huge effect in terms of the time and effort required to fill the SSA and is something that the changes in the SSA cannot affect to. When considering the SSA as a process, it is entirely manual and a general solution to speeding up a manual process is to try and automate it. A benefit of automation is not only that it could speed up the process of filling out the SSA, but additionally it could be utilized to make the documentation more living and reactive to changes. This could also enable a change in the current process of revising the SSA: instead of going through the changes and assessing their effect on the general security of the software cyclically every year, it would enable an approach where the need of revising the SSA would be based on detected changes of the software in real-time, which would in turn reduce the time of getting the required security patches applied to the software. Additionally, one key process where the automation would also have a profound impact is the process of reviewing the assessment. In its current state, the assessment process is also entirely manual and the expected time for getting your SSA reviewed is usually a few weeks. If the process of reviewing the questions could be automated through an automated scoring system, for example, the response time for getting a score for the assessment would be significantly improved. Creating the automation framework around the fluid contents of the SSA as well as an automated scoring system would be undoubtedly a challenging task given the heterogeneity of the assessed software products and is a topic that would require further studying to fully understand.

6.4 Perspective

Lastly, in this thesis the focus point has been the security engineer perspective of things. It is evident that in addition to security engineers a lot of other people are involved in the process of filling out the SSA such as product owners, data protection managers and software architects. In further discussions about the results with the company authorities responsible for reviewing the SSA's, it became quite evident that some of the gathered feedback would look quite different from the perspective of these other user groups. Additionally, it is now acknowledged that in its current state the process of filling out the SSA lies almost entirely on the responsibility of the security engineers, and in this light the suggested improvements for emphasizing collaboration and knowledge sharing can be seen as even more vital than what the results are showing.

7 CONCLUSION

From the perspective of software security, having and conducting an SSA makes a lot of sense: as software security and data protection compliance are effectively overly broad and difficult topics to handle, it effectively provides the teams with a ready framework for identifying any potential misconducts and underlying issues with their product. Even in cases where no actual concrete improvements are required, it will still improve the overall understanding of the target application, provide information security documentation for use in other contexts and helps to spread the security awareness to the personnel.

Based on this study, when designing an SSA for a company, the key focus points should be on identifying what is valuable to the company and to the software. Topics of software security and data protection are overly broad and highly dependent on the context, such as the type of software, internal company policies, local legislation, formal requirements and required standard compliance. The point of design should be to design the SSA in a way that it covers compatibility with these different requirements as well as the crucial areas of software security.

Once the SSA has been created, it must be emphasized that it should be constantly evolving, as the field of software security, technologies, legislation, standards and even the commonly accepted best practices are always changing. To also support the development of the model, collecting and analyzing user experiences as was done in this thesis can help to uncover any underlying difficulties and issues within the current model that would be hard to identify otherwise, and enables the companies to utilize the collective ideas of the people working with the assessment to improve the process and the contents of the SSA.

While this study does point towards the SSA providing a substantial number of benefits regarding software security, it has also raised the question about the diminishing returns of concurrent assessments for any single application in the longer term. Therefore, the question regarding if the SSA is actually the best tool for the purpose of improving overall level of software security and awareness is something that cannot be reliably answered to without future studies.

8 REFERENCES

Anderson, R. J. (2001). *Security Engineering: A Guide to Building Dependable Distributed Systems* (2nd ed.). John Wiley & Sons, Inc.605 Third Ave. New York.

Arkin, B., Stender, S., & McGraw, G. (2005). Software penetration testing. *IEEE Security and Privacy Magazine*, *3*(1), 84–87. https://doi.org/10.1109/MSP.2005.23

Atlassian. (2021). *Confluence: Accomplish more together*. https://www.atlassian.com/software/confluence

Bihari, E. (2018). GDPR – A Y2K-II for Business? *EDPACS*, *57*(2), 1–9. https://doi.org/10.1080/07366981.2018.1426929

Borking, J. J., & Hes, R. (1995). *Privacy-Enhancing Technologies: The Path to Anonymity*. https://www.researchgate.net/publication/243777645_Privacy-Enhancing_Technologies_The_Path_to_Anonymity

Chatzipoulidis, A., Tsiakis, T., & Kargidis, T. (2019). A readiness assessment tool for GDPR compliance certification. *Computer Fraud & Security*, *2019*(8), 14–19. https://doi.org/10.1016/S1361-3723(19)30086-7

Cherdantseva, Y., & Hilton, J. (2014). Information Security and Information Assurance. In I. M. Portela & F. Almeida (Eds.), *Organizational, Legal, and Technological Dimensions of IS Administrator* (pp. 167–198). IGI Global. https://doi.org/10.4018/978-1-4666-4526-4.ch010

Chivers, H., Clark, J. A., & Cheng, P.-C. (2009). Risk profiles and distributed risk assessment. *Computers & Security*, *28*(7), 521–535. https://doi.org/10.1016/j.cose.2009.04.005

Chivers, H. R. (2006). *Security design analysis*. University of York.

CSA. (2021a). *Cloud Security Alliance: Security, Trust, Assurance and Risk Registry*. https://cloudsecurityalliance.org/star/registry/

CSA. (2021b, January 4). *Cloud Security Alliance: Consensus Assessment Initiative Questionnaire v4.0.3*. https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4/

Danziger, M., & da Silva, M. A. (2015, May). The Importance of Security Requirements Elicitation and How to Do It. *PMI Global Congress Proceedings*. https://www.researchgate.net/publication/304539849_The_Importance_of_Security_Requirements_Elicitation_and_How_to_Do_It

Eltahawy, B. (2021). *Lecture 1: Introduction to Cyber Security, Management of Cybersecurity TITE3370*. University of Vaasa School of Technology and Innovations.

European Parliament and the council. (2016). *REGULATION (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*.

Everett, C. (2011). Is ISO 27001 worth it? *Computer Fraud & Security*, *2011*(1), 5–7. https://doi.org/10.1016/S1361-3723(11)70005-7

Howard, M., & Leblanc, D. E. (2003). *Writing Secure Code* (2nd ed.). Microsoft PressDiv. of Microsoft Corp. One Microsoft Way Redmond, WAUnited States.

Humphreys, E. (2016). *Implementing the ISO/IEC 27001 ISMS standard* (Second Editon). Artech House.

Hustinx, P. (2010). Privacy by design: delivering the promises. *Identity in the Information Society*, *3*(2), 253–255. https://doi.org/10.1007/s12394-010-0061-z

Information Commissioner’s Office. (2021). *https://ico.org.uk/for-organisations/data-protection-advice-for-small-organisations/checklists/data-protection-self-assessment/*. https://ico.org.uk/for-organisations/data-protection-advice-for-small-organisations/checklists/data-protection-self-assessment/

International Organization for Standardization. (2005). *ISO/IEC 15408-1/2/3:2005 - Information technology — Security techniques — Evaluation criteria for IT security*.

International Organization for Standardization. (2018a). *ISO/IEC 27004:2016 Information technology — Security techniques — Information security management — Monitoring, measurement, analysis and evaluation*.

International Organization for Standardization. (2018b). *ISO/IEC 27000:2018 Information technology — Security techniques — Information security management systems — Overview and vocabulary* (Vol. 5, pp. 1–27).

IsecT ltd. (2021). *ISO27k Information Security Program Maturity Assessment Tool*. https://www.iso27001security.com/ISO27k_Information_security_program_maturity_assessment_tool.xlsx

Islam, S., & Falcarin, P. (2011). Measuring security requirements for software security. *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, 70–75. https://doi.org/10.1109/CIS.2011.6169137

Jaatun, M. G. (2012). Hunting for Aardvarks: Can Software Security be Measured? *Multidisciplinary Research and Practice for Information Systems Lecture Notes in Computer Science*, *7465*, 85–92.

Jain, S., & Ingle, M. (2011). A Review of Security Metrics in Software Development Process . *International Journal of Computer Science and Information Technologies*, *2*(6), 2627–2631. https://www.researchgate.net/publication/267829664_A_Review_of_Security_Metrics_in_Software_Development_Process

Mcgraw, G. (2004). Software security. *IEEE Security & Privacy Magazine, 2*(2), 80–83. https://doi.org/10.1109/MSECP.2004.1281254

McGraw, G. (2006). Software Security: Building Security In. In *2006 17th International Symposium on Software Reliability Engineering* (1st ed.). Addison-Wesley Professional.

McGraw, G., Migues, S., & West, J. (2015). *Building Security In Maturity Model 6*. https://www.inf.ed.ac.uk/teaching/courses/sp/2015/lecs/BSIMM6.pdf

Mellado, D., Fernández-Medina, E., & Piattini, M. (2010). A comparison of software design security metrics. *Proceedings of the Fourth European Conference on Software Architecture Companion Volume - ECSA ’10*, 236–242. https://doi.org/10.1145/1842752.1842797

Merriam-Webster. (2021). *“Safety.”* Merriam-Webster.Com Dictionary. https://www.merriam-webster.com/dictionary/safety.

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

Mkpong-Ruffin, I., Umphress, D., Hamilton, J., & Gilbert, J. (2007). Quantitative software security risk assessment model. *Proceedings of the 2007 ACM Workshop on Quality of Protection - QoP ’07*, 31–33. https://doi.org/10.1145/1314257.1314267

National Institute of Standards and Technology. (2008). *SP 800-55 Rev. 1 Performance Measurement Guide for Information Security*.

OWASP. (2020). *Open Web Application Security Project Software Assurance Maturity Model*. https://owasp.org/www-project-samm/

Payment Card Industry. (2013). *Data Security Standard: Vol. 3.0*.

Payment Card Industry. (2018). *Data Security Standard - Attestation of Compliance for Onsite Assessments – Service Providers* (pp. 1–11).

Ramachandran, M. (2016). Software security requirements management as an emerging cloud computing service. *International Journal of Information Management*, *36*(4), 580–590. https://doi.org/10.1016/j.ijinfomgt.2016.03.008

Savola, R. (2007). Towards a Security Metrics Taxonomy for the Information and Communication Technology Industry. *International Conference on Software Engineering Advances (ICSEA 2007)*, 60–60. https://doi.org/10.1109/ICSEA.2007.79

Scandariato, R., de Win, B., & Joosen, W. (2006a). Towards a measuring framework for security properties of software. *Proceedings of the 2nd ACM Workshop on Quality of Protection - QoP '06*. https://doi.org/10.1145/1179494.1179500

Scandariato, R., de Win, B., & Joosen, W. (2006b). Towards a measuring framework for security properties of software. *Proceedings of the 2nd ACM Workshop on Quality of Protection - QoP '06*, 27–30. https://doi.org/10.1145/1179494.1179500

Shubhamangala, S. (2015). The Need for Measuring the Quality of Application Security. *Software Quality Professional*, *17*(2), 30–43.

Šikman, L., Latinović, T., & Paspalj, D. (2019). ISO 27001 - INFORMATION SYSTEMS SECURITY, DEVELOPMENT, TRENDS, TECHNICAL AND ECONOMIC CHALLENGES. *Annals of the Faculty of Engineering Hunedoara*, *17*(4), 45–48.

U.S. Department of Health and Human Services Office for Civil Rights. (2016). *HIPAA Administrative Simplification* (pp. 1–115).

Viega, J., & McGraw, G. (2001). *Building secure software; how to avoid security problems the right way*. (1st ed.). Addison-Wesley Professional.

Williams, L., McGraw, G., & Migues, S. (2018). Engineering Security Vulnerability Prevention, Detection, and Response. *IEEE Software*, *35*(5), 76–80. https://doi.org/10.1109/MS.2018.290110854

APPENDIX 1. Code system

Code System	Frequency
Total codes	576
Context	0
Context\SSO portal	1
Context\Internal tool	1
Context\Web application	3
Context\Web application\PHP	1
Context\Web application component	3
Context\API	1
Context\New software	2
Context\Old software	2
Context\On Prem	6
Deviations from the process	0
Deviations from the process\External risk management system	1
Deviations from the process\Skipped a section	3
Deviations from the process\Non-formal approach	2
Deviations from the process\Cloud services manage things automatically	1
Benefits	0
Benefits\Increases awareness of corporation provided services	1
Benefits\Increases awareness of corporation provided services\SEC17	1
Benefits\Resulted in thorough inspection	3
Benefits\Resulted in thorough inspection\DP01	1
Benefits\Resulted in thorough inspection\DP03	1
Benefits\Resulted in thorough inspection\SEC01	2
Benefits\Resulted in thorough inspection\SEC02	1
Benefits\Resulted in thorough inspection\SEC06	1
Benefits\Resulted in thorough inspection\SEC10	2
Benefits\Resulted in thorough inspection\SEC14	1
Benefits\Resulted in thorough inspection\RM03	1
Benefits\Resulted in better documentation	1

Benefits\Resulted in better documentation\SEC04	1
Benefits\Resulted in better documentation\SEC09	1
Benefits\Resulted in better documentation\SEC10	1
Benefits\Provided a good base for answering further questions	3
Benefits\Provided a good base for answering further questions\DP01	1
Benefits\Provided a good base for answering further questions\SEC01	4
Benefits\Provided a good base for answering further questions\RM03	1
Benefits\Inspected the application from an attacker's perspective	2
Benefits\Inspected the application from an attacker's perspective\SEC02	1
Benefits\Inspected the application from an attacker's perspective\SEC06	1
Benefits\Inspected the application from an attacker's perspective\SEC11	1
Benefits\Security findings and improvements	24
Benefits\Security findings and improvements\DP01	1
Benefits\Security findings and improvements\DP03	10
Benefits\Security findings and improvements\DP04	2
Benefits\Security findings and improvements\SEC01	3
Benefits\Security findings and improvements\SEC02	5
Benefits\Security findings and improvements\SEC03	4
Benefits\Security findings and improvements\SEC04	4
Benefits\Security findings and improvements\SEC05	2
Benefits\Security findings and improvements\SEC06	3
Benefits\Security findings and improvements\SEC07	1
Benefits\Security findings and improvements\SEC08	5
Benefits\Security findings and improvements\SEC09	6
Benefits\Security findings and improvements\SEC10	6
Benefits\Security findings and improvements\SEC11	1
Benefits\Security findings and improvements\SEC12	1
Benefits\Security findings and improvements\SEC14	8
Benefits\Security findings and improvements\SEC16	2

Benefits\Security findings and improvements\SEC17	1
Benefits\Security findings and improvements\RM02	1
Benefits\Better security procedures and routines	1
Benefits\Better security procedures and routines\SEC07	1
Benefits\Provided direct improvement steps	3
Benefits\Provided direct improvement steps\DP03	2
Benefits\Provided direct improvement steps\SEC04	1
Benefits\Provided direct improvement steps\SEC05	1
Benefits\Provided direct improvement steps\SEC06	2
Benefits\Provided direct improvement steps\SEC07	2
Benefits\Provided direct improvement steps\SEC08	1
Benefits\Provided direct improvement steps\SEC09	1
Benefits\Brings structure to security assessment process	5
Benefits\Brings structure to security assessment process\DP03	3
Benefits\Brings structure to security assessment process\SEC01	1
Benefits\Brings structure to security assessment process\SEC02	2
Benefits\Brings structure to security assessment process\SEC05	1
Benefits\Brings structure to security assessment process\SEC04	1
Benefits\Brings structure to security assessment process\SEC09	1
Benefits\Brings structure to security assessment process\SEC16	1
Benefits\Brings structure to security assessment process\RM01	2
Benefits\Brings structure to security assessment process\RM02	1
Benefits\Brings structure to security assessment process\RM03	1
Benefits\Brings structure to security assessment process\RM04	1
Benefits\Increased security awareness and thinking	11
Benefits\Increased security awareness and thinking\DP01	2
Benefits\Increased security awareness and thinking\DP02	1
Benefits\Increased security awareness and thinking\DP03	5
Benefits\Increased security awareness and thinking\SEC02	5
Benefits\Increased security awareness and thinking\SEC04	1
Benefits\Increased security awareness and thinking\SEC05	1
Benefits\Increased security awareness and thinking\SEC06	2
Benefits\Increased security awareness and thinking\SEC07	1

Benefits\Increased security awareness and thinking\SEC08	1
Benefits\Increased security awareness and thinking\SEC09	6
Benefits\Increased security awareness and thinking\SEC10	2
Benefits\Increased security awareness and thinking\SEC12	1
Benefits\Increased security awareness and thinking\SEC14	2
Benefits\Increased security awareness and thinking\RM02	1
Benefits\Increased security awareness and thinking\RM03	1
Benefits\Can be referred to in additional audits	1
Benefits\Discussions with the security team	2
Benefits\Directs to further information	1
Challenges	0
Challenges\Data Protection	0
Challenges\Data Protection\DP01: Data list	0
Challenges\Data Protection\DP01: Data list\Generally difficult	1
Challenges\Data Protection\DP01: Data list\Exhaustive for large application	1
Challenges\Data Protection\DP01: Data list\Copy pasting from existing documentation	1
Challenges\Data Protection\DP01: Data list\Fluid section	1
Challenges\Data Protection\DP02: Data classification	0
Challenges\Data Protection\DP02: Data classification\Fluid section	1
Challenges\Data Protection\DP03: Privacy and data protection by design	0
Challenges\Data Protection\DP03: Privacy and data protection by design\Inability to affect	1
Challenges\Data Protection\DP04: Formal requirements and standards	0
Challenges\Data Protection\DP04: Formal requirements and standards\Outside area of expertise	2
Challenges\Data Protection\DP04: Formal requirements and standards\Inability to affect	1
Challenges\Data Protection\DP04: Formal requirements and standards\Responsibilities	1
Challenges\Data Protection\DP05: Customer contract and supported versions	0
Challenges\Data Protection\DP05: Customer contract and	3

supported versions\Outside area of expertise	
Challenges\Data Protection\DP05: Customer contract and supported versions\Inability to affect	3
Challenges\Data Protection\DP05: Customer contract and supported versions\Responsibilities	1
Challenges\Data Protection\Overlapping	1
Challenges\Data Protection\Required competence	5
Challenges\Data Protection\Unclear instructions	1
Challenges\Data Protection\Inapplicable questions	7
Challenges\Data Protection\3rd party data	6
Challenges\Security	0
Challenges\Security\SEC01: System Diagram	0
Challenges\Security\SEC01: System Diagram\Time-consuming	1
Challenges\Security\SEC01: System Diagram\Requires looking at other SSA's for example	1
Challenges\Security\SEC01: System Diagram\Unclear instructions	3
Challenges\Security\SEC02: Attack surfaces	0
Challenges\Security\SEC02: Attack surfaces\Overlapping	1
Challenges\Security\SEC02: Attack surfaces\Time-consuming	1
Challenges\Security\SEC02: Attack surfaces\Unclear instructions	3
Challenges\Security\SEC03: Access control quality	0
Challenges\Security\SEC03: Access control quality\Fluid section	1
Challenges\Security\SEC05: Crypto and hash algorithms	0
Challenges\Security\SEC05: Crypto and hash algorithms\Inapplicable questions	2
Challenges\Security\SEC05: Crypto and hash algorithms\Fluid section	2
Challenges\Security\SEC05: Crypto and hash algorithms\Too large	1
Challenges\Security\SEC05: Crypto and hash algorithms\Unclear instructions	1
Challenges\Security\SEC07: Software dependencies	0
Challenges\Security\SEC07: Software dependencies\Difficult without ATVS	1

Challenges\Security\SEC07: Software dependencies\Fluid section	2
Challenges\Security\SEC08: File upload validation	0
Challenges\Security\SEC08: File upload validation\Inapplicable questions	1
Challenges\Security\SEC10: Secret management	0
Challenges\Security\SEC10: Secret management\Time-consuming fix	1
Challenges\Security\SEC10: Secret management\Fluid section	3
Challenges\Security\SEC11: Phishing	0
Challenges\Security\SEC11: Phishing\Inapplicable questions	5
Challenges\Security\SEC13: Secure deployment	0
Challenges\Security\SEC13: Secure deployment\covered by VCDM	1
Challenges\Security\SEC13: Secure deployment\Fluid section	1
Challenges\Security\SEC14: Infrastructure permissions	0
Challenges\Security\SEC14: Infrastructure permissions\Fluid section	2
Challenges\Security\SEC15: Host and network security	0
Challenges\Security\SEC15: Host and network security\Unclear instructions for remediation	1
Challenges\Security\SEC15: Host and network security\External hosting service	1
Challenges\Security\SEC17: Threat Intelligence	0
Challenges\Security\SEC17: Threat Intelligence\Duplicate work (already listed in PSC)	1
Challenges\Security\SEC17: Threat Intelligence\Ineffective in comparison to threat modeling	1
Challenges\Security\Required competence	1
Challenges\Security\Inapplicable questions	7
Challenges\Security\3rd party facilities and tools	2
Challenges\Risk management	0
Challenges\Risk management\Unclear instructions	6
Challenges\Risk management\Duplicates	1
Challenges\Risk management\Microsoft Threat Modeling Tool	2
Challenges\Risk management\Microsoft Threat Modeling Tool\Too low level	2

Challenges\Risk management\Risk issue visibility	2
Challenges\Risk management\Unknown variables	1
Challenges\Time consumption	3
Challenges\Time consumption\Reserving time from other people	1
Challenges\Required competence	3
Challenges\Required competence\Insufficient existing documentation	1
Challenges\Required competence\Reflects onto the responses	1
Challenges\Required competence\No introduction	1
Challenges\Required competence\Uncertainty of your answer's correctness	1
Challenges\Instructions	0
Challenges\Instructions\Unclear	1
Challenges\Instructions\Lacking links to external sources	3
Challenges\Instructions\Requires looking at other SSA's for instructions	1
Challenges\Compatibility with different applications	1
Challenges\Compatibility with different applications\Level of accuracy	1
Challenges\Compatibility with different applications\PHP	1
Challenges\Compatibility with different applications\Exhaustive for large systems	1
Challenges\Compatibility with different applications\Too heavy for early-stage software	5
Challenges\Compatibility with different applications\On Prem applications	6
Challenges\First time assessment	0
Challenges\First time assessment\Time-consuming	3
Challenges\First time assessment\Required competence	4
Challenges\Assessment renewal	0
Challenges\Assessment renewal\Time-consuming	2
Challenges\JIRA	2
Challenges\JIRA\Not using JIRA	1
Challenges\JIRA\Managing identified vulnerabilities	1
Challenges\Motivating teams to carry out fixes	3

Challenges\3rd parties	1
Challenges\Documenting fluid things	2
Improvements	0
Improvements\Include VCDM approved templates	1
Improvements\Reduce bureaucracy	1
Improvements\JIRA guidance	1
Improvements\Improve template	1
Improvements\Improve template\Ensure the macros work	1
Improvements\Improve template\Easier renewal	2
Improvements\Improve template\Change platform	5
Improvements\Share knowledge	1
Improvements\Process	0
Improvements\Process\Change Risk Management procedure	1
Improvements\Process\Reduce reviewing time	1
Improvements\Process\Introduction to Security Engineers	1
Improvements\Process\Improve delegation	2
Improvements\Process\Meeting / Audit with SEC team prior to filling the SSA	7
Improvements\Process\Be honest	1
Improvements\Process\Involving developers in the renewal process	1
Improvements\Process\Involving system architects	2
Improvements\Adaptive SSA	15
Improvements\Structure	0
Improvements\Structure\Sections	0
Improvements\Structure\Sections\Infrastructure checklist	1
Improvements\Structure\Sections\Reduce overlap	2
Improvements\Structure\Sections\Reduce overlap\With other VCDM assessments	3
Improvements\Structure\Sections\Skip the obvious things	1
Improvements\Structure\Sections\Detach DP section	7
Improvements\Structure\Sections\Start with the Security section	1
Improvements\Structure\Sections\New section: Cloud security	1

Improvements\Structure\Sections\Focus on logical grouping of data rather than listing	2
Improvements\Structure\Sections\Split DP03 to subsections	1
Improvements\Structure\Sections\New section: Network overview	2
Improvements\Structure\Sections\New section: Infrastructure Security	2
Improvements\Structure\Sections\Grouping based on similarity	3
Improvements\Structure\Sections\Grouping based on required effort	1
Improvements\Structure\Questions and instructions	0
Improvements\Structure\Questions and instructions\Code checklist	2
Improvements\Structure\Questions and instructions\Focus on things not automatically handled by frameworks etc.	1
Improvements\Structure\Questions and instructions\Add links to external information	3
Improvements\Structure\Questions and instructions\Better instructions	1
Improvements\Structure\Questions and instructions\Better instructions\SEC14: Infrastructure permissions	1
Improvements\Structure\Questions and instructions\Better instructions\SEC10: Secret Management	1
Improvements\Structure\Questions and instructions\Better instructions\SEC01: System diagram	1
Improvements\Structure\Questions and instructions\Include reasoning for questions	2
Improvements\Structure\Questions and instructions\Guidelines	1
Improvements\Structure\Questions and instructions\Less ambiguous	10
Improvements\Structure\Questions and instructions\Example answers	5
Improvements\Structure\Questions and instructions\Example answers\Utilize existing answers to create examples	1
Improvements\Structure\Questions and instructions\Predefined answers	3

Improvements\Automation / Continuous assessment	0
Improvements\Automation / Continuous assessment\3rd party attribution and licenses checking	1
Improvements\Automation / Continuous assessment\Automated data gathering	3
Improvements\Automation / Continuous assessment\Endpoint scanning	1
Improvements\Automation / Continuous assessment\Fluid sections	2
Improvements\Automation / Continuous assessment\IaC scanning	4
Improvements\Make it lighter	3
User experience	0
User experience\Negative	0
User experience\Negative\Changing tools	1
User experience\Negative\Review process	1
User experience\Negative\Review process\Time-consuming	1
User experience\Negative\One-time things	1
User experience\Negative\Confluence	8
User experience\Negative\New SSA	0
User experience\Negative\New SSA\Documenting defaults	1
User experience\Negative\New SSA\Threat modeling is not thoroughly examined	1
User experience\Negative\New SSA\Nonfunctional color coding	1
User experience\Negative\New SSA\Complexity	1
User experience\Negative\New SSA\Bureaucratic	3
User experience\Negative\New SSA\Less thorough	4
User experience\Negative\Old SSA	1
User experience\Negative\Old SSA\Too detailed	5
User experience\Negative\Transition from old to new	1
User experience\Negative\Transition from old to new\Copy paste	1
User experience\Positive	0
User experience\Positive\Old SSA	2
User experience\Positive\Old SSA\More thorough	2

User experience\Positive\Tools	3
User experience\Positive\Tools\Microsoft Project Management tool	1
User experience\Positive\Tools\Microsoft Threat Modeling Tool	2
User experience\Positive\Reduced scope	3
User experience\Positive\Reduced scope\For On Premise	1
User experience\Positive\Reduced scope\For API's	2
User experience\Positive\Requesting review	1
User experience\Positive\Renewal process	2
User experience\Positive\Teamwork	7
User experience\Positive\Checklists	1
User experience\Positive\New SSA	13
User experience\Positive\New SSA\Trivial sections	1
User experience\Positive\New SSA\Clearer	3
User experience\Positive\New SSA\Not Visma.net focused	1
User experience\Positive\New SSA\DP section	4
User experience\Positive\New SSA\Straight forward	3
User experience\Positive\New SSA\Compact	3
User experience\Positive\New SSA\High level	1
User experience\Neutral	0
User experience\Neutral\Boundary between DP and SEC	1
User experience\Neutral\Changing tools	1
User experience\Neutral\Effort	1
User experience\Neutral\Confluence	3