MASTER'S THESIS 2020

Sharing is caring

Communicating recipient-tailored OSS vulnerability information on an online platform

Emmy Dahl & Michaela Karlsson

ISSN 1650-2884
LU-CS-EX: 2020-48

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2020-48

Sharing is caring

Communicating recipient-tailored OSS vulnerability information on an online platform

Emmy Dahl & Michaela Karlsson

Sharing is caring

Communicating recipient-tailored OSS vulnerability information on an online platform

Emmy Dahl
ine14eda@student.lu.se

Michaela Karlsson
jur13mk1@student.lu.se

August 22, 2020

Master's thesis work carried out at
the Department of Computer Science, Lund University.

Supervisor: Martin Hell, martin.hell@eit.lth.se

Examiner: Martin Höst, martin.host@cs.lth.se

Abstract

Title
Sharing is caring: Communicating recipient-tailored OSS vulnerability information on an online platform

Authors
Emmy Dahl & Michaela Karlsson

Supervisors
Martin Hell, Faculty of Engineering, Lund University
Martin Höst, Faculty of Engineering, Lund University

Background
With IoT and digitisation in general comes initiatives for malicious actors to exploit possible vulnerabilities in the software used. Both the usage of open source software and detected vulnerabilities have increased in the last decade and it is becoming important for companies to know what weaknesses they have in their software systems, to enable secure products. The total cybersecurity of a product often depends on cooperation between several actors. Presently the communication regarding vulnerabilities in organisations is done reactively instead of proactively. This even though research has shown that sensitive information sharing increases the performance of the actors in a network. This insinuates an industrial need for structured communication between organisations regarding software vulnerability management.

Purpose
The purpose of this thesis is to investigate how vulnerability information can be tailored to different recipients. More specifically how vulnerability information can be grouped and presented on an online platform, where different views can be accessed by the targeted recipients.

Research Questions
RQ1 Who are the typical recipients of vulnerability information within a company?
RQ2 What kind of vulnerability information does each type of recipient need within the frame of their profession?
RQ3 How can vulnerability information be tailored and presented on an online-sharing platform for each type of recipient?

Methodology
The study was conducted as a qualitative case study, consisting of 12 rounds of interviews with people that come across vulnerabilities in their everyday work. The

interview results were analysed using the qualitative analysis program NVivo and then compared to existing theoretical frameworks. Lastly prototype views were created, evaluated and adjusted. The findings were discussed and concluded.

Conclusions
The most prominent recipient groups that are identified are triage responsible, development team, communication function, support function, management and board and customer. The recipient groups triage responsible and development team need more technically detailed information, because of their deep involvement with the technical aspects of handling vulnerabilities. Communication function, support function and management and board need vulnerability information that is related to their more business, communication and customer inclined responsibilities. Customer needs information that is as straight forward and solution oriented as possible. Prototype views are designed for each prominent recipient group. These views are called the Triage view, the Development view, the Communication function view, the Support function view, the Management and Board view and finally the Customer view.

Keywords: vulnerability, cybersecurity, open source software, software development roles, HAVOSS, vulnerability handling, vulnerability management, Vulnerability Information Recipient Onion Model

Acknowledgements

This master thesis was written during the spring of 2020 as a final project within the study program Master of Science in Industrial Engineering and Management, at the Faculty of Engineering at Lund University. The study was conducted in cooperation with the research project HATCH financed by Vinnova and Debricked's SecT project. It has been an opportunity for us to apply knowledge from our studies in practice and to get interesting insights into the world of vulnerabilities and cybersecurity.

First of all, we want to express our sincere gratitude and appreciation to our supervisor Martin Hell at the department of Electrical and Information Technology and our examiner Martin Höst at the department of Computer Science, both at the Faculty of Engineering. You have provided us with valuable guidance and support throughout the entire duration of the study and without your constant challenging of our aspirations, ideas and comments, this thesis would not have been possible.

We want to express our sincere gratitude towards the interviewees, for devoting their time and sharing valuable insights to our work. We want to thank Vinnova and the HATCH project for providing a great foundation for our research to be carried out. We also want to thank Debricked for giving us the opportunity to conduct our thesis in collaboration with them and including us in their team of highly competent co-workers.

Finally, last but not least, we want to thank our families and friends for their support during the sometime crooked path to our graduation as engineers. "It takes a village to raise a child" is an African proverb summarising our academic years very well, with only a slight adjustment being required: It takes a village to raise an engineer. Thank you.

Emmy Dahl & Michaela Karlsson
Lund, June 2020

Acronyms

CVE Common Vulnerabilities and Exposures

CVSS Common Vulnerability Scoring System

HAVOSS HAndling Vulnerabilities in third party OSS

NCSC National Cyber Security Center

NVD National Vulnerability Database

OSS Open Source Software

RM Risk Management

SDLC Software Development Life Cycle

SPOC Single Point of Contact

SQuaRE Software Quality Requirements and Evaluation

SR Software Requirements

TKOM Tacit Knowledge Onion Model

VIROM Vulnerability Information Recipient Onion Model

VM Vulnerability Management

XP Extreme Programming

List of Figures

List of Tables

Contents

Chapter 1
Introduction

The Introduction chapter describes the general subject of the thesis and contains relevant information such as background, problem description, purpose, research questions and delimitations. The outline of the report presents a description of every chapter.

1.1 Background

1.1.1 Cybersecurity

In 1996, a group of Swedish hackers broke into the computer systems of not only the U.S. space agency NASA, but also the U.S. Air Force, the Army and the Marines [38]. The hackers called themselves *Fragglarna*, were in their early 20s and claimed they did it for the sake of a challenging learning experience. Despite their actions causing substantial economic losses for the affected agencies, the hackers were spared extradition and thus avoided being sentenced to 60 years in a U.S. prison [7].

Both the World Wide Web and cybersecurity were fairly new concepts for the public during the 1990s and not as developed technologies as they are today [41]. The immaturity of the Internet as an information technology can perhaps be considered an explanation of how it was possible for a couple of youths to remotely access a great power nation's deepest secrets. Yet a quarter of a century later, malicious cyber activity is getting more widespread and targeting individuals, companies and even whole societies. Information and communication technologies have with all their possibilities evolved into general purpose technologies, integrated with various parts of everyday life [39]. A downside to widespread connectedness in a society is that the more cyber power a society possess, the higher the cyber threat gets [4]. The more humans that rely on connected digital solutions, e.g. using online banking and communication apps or inviting Internet of Things products into their physical lives, the more vulnerable societies become to malicious actors [39].

A company not being able to keep vital data secure can cause dire consequences for its

business, damaging its reputation and economic situation [5]. In 2019 exploitation of vulnerabilities was amongst the top common reasons to security incidents [17] and the global average cost of a data breach was estimated to 3.92 million U.S. dollars [18]. Meanwhile there is a lack of 2.93 million cybersecurity employees around the world, leaving many companies without the competence to develop and implement necessary defence strategies [5].

1.1.2 Vulnerabilities

A vulnerability is defined as "a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source" [28]. Common Vulnerability and Exposures (CVE) is the industry standard for identifying vulnerabilities and exposures. For more than 20 years publicly known cybersecurity vulnerabilities and exposures have been published in a database called National Vulnerability Database (NVD), which can be accessed at `https://nvd.nist.gov/`. Every CVE has an ID number and every CVE entry in the database contains a description of the issue, its vulnerability severity rating according to the common vulnerability scoring system (CVSS) and relevant references to related advisories and reports [28]. In Table 1.1 the ranges for severity rating of vulnerabilities according to the two versions of CVSS are presented.

In 1999, the same year as *Fragglarna* were charged for hacking the U.S. defence agencies [38], the number of reported vulnerabilities was 894. During 2017-2019 the number of reported vulnerabilities reached 43444, which responds to a yearly average of 14481 [6]. Far from all vulnerabilities are represented in these numbers, simply because they are yet to be discovered, are not reported to the database or are not public due to security reasons [28]. One of the most famous vulnerabilities is the Heartbleed Bug, with ID number CVE-2014-0160. As the ID number tells it was discovered in 2014. This was a vulnerability in the OpenSSL cryptographic software library that allowed stealing of information encrypted by SSL/TLS encryption. It affected almost every corner of the Internet, making the Heartbleed Bug a sensational part of the history of the digital era. [10]. Figure 1.1 depicts how information about the Heartbleed Bug is presented in NVD [25].

Table 1.1: NVD vulnerability severity ratings [29].

Severity	CVSS v2.0 base score range	CVSS v3.0 base score range
None	-	0.0
Low	0.0-3.9	0.1-3.9
Medium	4.0-6.9	4.0-6.9
High	7.0-10.0	7.0-8.9
Critical	-	9.0-10.0

1.1.3 Open Source Software

Software with source code that can be modified and shared by anyone is called open source software (OSS). The source code can be described as the DNA of a software, as it decides how

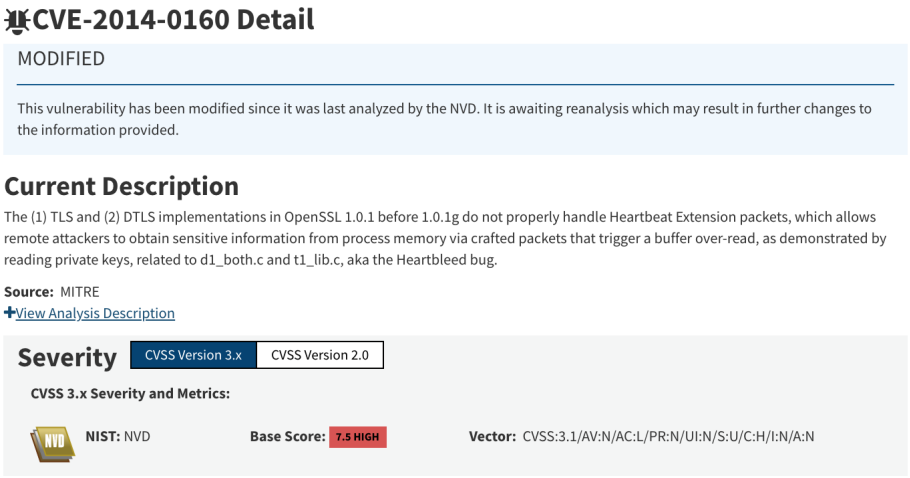


Figure 1.1: NVD information about the Heartbleed Bug [25].

a program or application originating from it should operate. Since OSS is more or less publicly accessible, it is possible for anyone with the right set of skills to adjust the software for their own purpose. The transparency and collaboration in OSS facilitate development of new software [31], has turned OSS into best practice among large companies [11]. In 2019, 96% of all companies were found to use open source code in their software development [37]. When OSS is used in software development, the integration of the developed components into systems creates an interesting flow of information. In many cases only the software developing actor possess knowledge of what possible vulnerabilities might occur in the software, while the integrating actor lacks detailed knowledge of the software in their systems [14]

1.2 Problem Formulation

The vulnerabilities in OSS are usually published in public databases, entailing continuous risks for malicious actors taking advantage of known existing vulnerabilities. Even though development security controls are applied in order to mitigate risks, vulnerabilities still occur and thereby enable insecure products [16].

According to a survey done in 2019 [30], the total cybersecurity of a product depends on cooperation between several actors. Presently there seem to exist an openness amongst companies towards communicating vulnerabilities with fellow industry actors, but the communication is mainly done reactively instead of proactively. The survey concludes that future studies can go deeper into communication of vulnerabilities and result in possible guidelines or recommendations for how the communication can be structured. This both internally and externally, especially between developer and integrator, but also generally how sensitive information can be tailored to different recipients.

In the complex software ecosystem, a value chain of industry actors can be identified. These actors often shift between being producers and integrators of components and systems of components [16]. For a majority of these industry actors, the vast amount of new vulnerabilities published every day generates extensive information that is difficult to efficiently review and process. In industrial networks where symbiotic activities are conducted, sensitive information sharing increases the performance of the actors in the networks more

than non-sensitive information sharing does. It does so regardless of what way the networks have been constructed, whether if it is done top-down or ad-hoc [9]. Hence improved vulnerability management and increased knowledge thereof can render more agile response to possible risks [16].

The above insinuates an industrial need for structured communication between organisations regarding software vulnerability management [16]. An investigative case study into the area can render a differentiation of information recipient groups within companies, as well as suggestions on how communication can be handled and in what way suggested information can be presented to various professional roles. Companies often consist of several employees that contribute to the organisational activities in different ways, depending on what their professional roles are. Different professional roles imply varying employee knowledge of software, vulnerabilities and their importance for the companies.

Figure 1.2 presents a possible solution that can improve vulnerability management between industry actors. In this scenario, the supplying actor uploads information about every known vulnerability that may be hiding in their implemented code as well as which of their products that may be affected by vulnerabilities, to the database S. The integrating customer actor does a similar thing by uploading information about their devices that originally are products from the supplying actor, to the database C. By comparing the data from databases S and C, commonalities between the databases can be detected and potential vulnerabilities and security threats identified. Such a configuration creates a security source from which valuable information for both actors can be extracted. In order to make the information structured and accessible in an efficient way for both actors, it can be presented on a tailored online-sharing platform. From this platform the different types of professional roles employed at each actor can have access to the information that is necessary for each role to carry out their work responsibilities. This possible solution to the problem formulation will be further explored and investigated in the study.

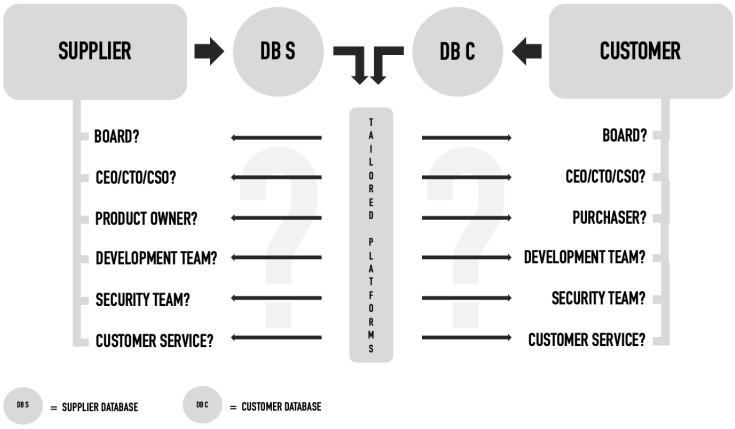


Figure 1.2: Illustration of the problem formulation.

1.3 Purpose

The main purpose of this thesis is to investigate vulnerability management of open source components within the industrial sector. The focus is to identify the recipients of vulnera-

bility information and the vulnerability information that each recipient type require within the frame of their profession. Suggestions on structuring relevant vulnerability information on an online-sharing platform are to be derived from the findings. The outcome of the conducted work will be a master thesis report, prototype views for the online-sharing platform, a conference paper and an input report to the HATCH project and Debricked's SecT project.

1.4 Research Questions

The research questions in this thesis are the following:

***RQ*1** Who are the typical recipients of vulnerability information within a company?

***RQ*2** What kind of vulnerability information does each type of recipient need within the frame of their profession?

***RQ*3** How can vulnerability information be tailored and presented on an online-sharing platform for each type of recipient?

1.5 Delimitations

This thesis is limited to focus on vulnerabilities in open source software and thus will not focus on vulnerabilities in networks. The study is limited to focus on vulnerabilities in third party components. All participators in the study will be from companies that have offices in Sweden but might also have other offices abroad. The study will not be limited to a specific industry as the final results are sought to be of general character and applicable regardless of industry. Also, the development of examples of how vulnerability information can be presented on an online-sharing platform is restricted to the design of the examples, not actual code implementation of any type of software.

1.6 The Outline of the Report

The report consists of the following eight chapters and appendices:

- **Chapter 1: Introduction** Chapter 1 describes the general subject of the study and contains relevant information such as background, problem description, purpose, research questions and delimitations. The outline of the report is presented with a description of every chapter.
- **Chapter 2: Theory** Chapter 2 describes the frame of reference for the study and is where the theoretical framework for the research is presented. The choice of framework is outlined and motivated in the beginning of the chapter, in order to facilitate the reader's understanding of the conducted work.
- **Chapter 3: Methodology** Chapter 3 describes the process of the study and the main methods applied during it. The motivation behind each methodological decision is

accounted for, as well as a discussion of whether the chosen methodology will result in a credible study.

- **Chapter 4: Interview Results** Chapter 4 contains the presentation of the empirically collected data from interviews with participating companies.
- **Chapter 5: Analysis** Chapter 5 contains the analysis of the presented data from the Interview Results chapter. The foundation of the analysis is the theoretical framework from the Theory chapter.
- **Chapter 6: Prototype Views** Chapter 6 contains the resulting prototype views that are based on the information obtained after the analysis. The chapter also contains the evaluation of the resulting prototype views, which is based on feedback from the interview respondents.
- **Chapter 7: Discussion** Chapter 7 contains the discussion of the conducted work and the results. The contribution of the thesis is discussed and evaluated.
- **Chapter 8: Conclusions** Chapter 8 contains the answers to the research questions and is where conclusions are drawn. Future research recommendations are given.
- **Appendices** Appendices contains the interview guides, the interview data and the prototype views for each recipient group.

Chapter 2
Theory

The Theory chapter describes the frame of reference for the study and is where the theoretical framework for the research is presented. An outline of the framework is illustrated in the beginning of the chapter, in order to facilitate the understanding of the conducted work.

2.1 Overview

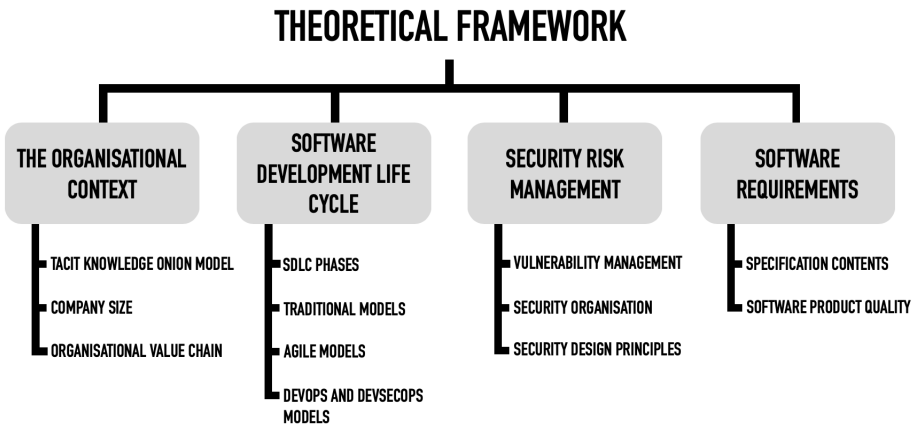


Figure 2.1: Overview of the theoretical framework.

Figure 2.1 illustrates an overview of the theoretical framework for the thesis. Section 2.2 contains theory surrounding company characteristics such as company size definitions, how knowledge can be regarded as a multilayer concept and the organisational value chain. Section 2.3 covers the fundamental theory of the software development life cycle, with various models of how to structure development and the different professional roles that are involved in each model. Section 2.4 focus on the theory of security risk management, how vulnerabil-

ity management is related to it and what professional roles a security organisation consists of. Section 2.5 covers the theoretical area of software requirements and their importance for creating high quality products.

2.2 The Organisational Context

2.2.1 The Tacit Knowledge Onion Model

Tacit knowledge is knowledge primarily based on accumulated experience, with its origin being difficult to point out for the holder of the knowledge in question. When seen as a multilayered phenomenon, tacit knowledge can be studied via the tacit knowledge onion model (TKOM), depicted in Figure 2.2. TKOM pinpoints the fact that tacit knowledge does exist in various layers. It shows how tacit knowledge is created in each layer and the interrelations of knowledge between the different layers. TKOM regards tacit knowledge being of both hierarchical and dynamic character, as there are different layers of tacit knowledge at different levels of the holder's consciousness. The further into the core of TKOM, the less aware the holder is of the tacit knowledge and the more abstract it gets. The outer layer represents knowledge that is explicit and the person who possess the knowledge is conscious of that. The second layer of hidden practical knowledge, is knowledge that comes to show in certain practical situations, while the third layer of reflective tacit knowledge is knowledge that is more of an unconscious character and could be accessed when the individual actively reflects upon a matter. Lastly the fourth layer of tacit knowledge that only can be demonstrated, is unconscious knowledge that can not really be converted into practical or reflective knowledge, but instead only shows itself when a task is performed successfully. [2].

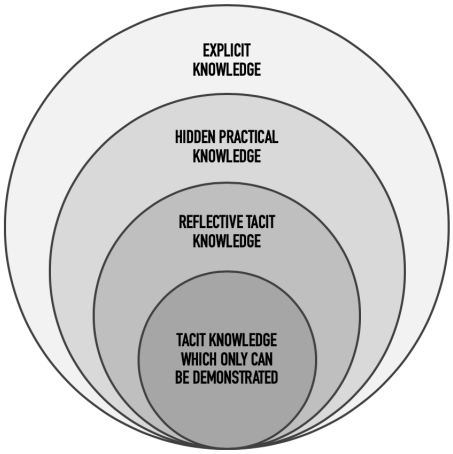


Figure 2.2: TKOM [2].

2.2.2 The Organisational Value Chain

In 1985, Michael E. Porter defined a model for value creating activities that are typically present within in a company that delivers products or services, see Figure 2.3. The purpose of

the model is to illustrate which organisational activities that actually create the value of the product or service and which activities that are more of a supportive character. The activities are therefore categorised as either primary or support, with primary being activities directly concerned to the creation process of the product or service, while support being activities improving the efficiency of the primary activities. As seen in Figure 2.3 the primary activities are the vertical boxes at the bottom part of the model, while the secondary activities are the boxes that are horizontal, in the upper part of the model. In Table 2.1 follows a description of the activities [21]. In an attempt to construct a version of the value chain adapted to activities surrounding the delivery of software products, inbound logistics can be replaced by intellectual property licensing. Operations can be broken down into the three activities research, product management and development. Finally, outbound logistics can be replaced by release [23]. In Figure 2.4 the interorganisational links and relationships necessary for the creation of a product or service, also known as the value network, are illustrated. It is a rare phenomenon for a single organisation to carry out all value activities from the design to the delivery of a product or service, as activities not centrally important for the organisation's strategic capabilities often are better being outsourced. It is not sufficient for an organisation to only make its own value chain efficient, as value creation occurs throughout the whole supply chain. Hence the other value network organisations' activities also affect the final quality of the product [21].

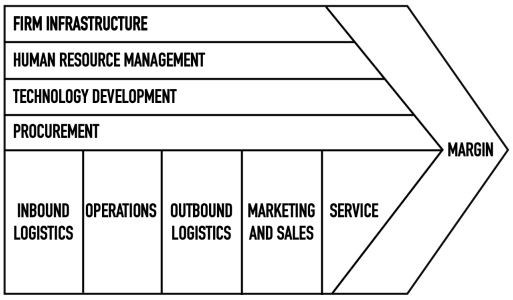


Figure 2.3: The value chain model for products and services [21].

Table 2.1: Value chain activities [23].

Primary activities	**Meaning**
Inbound logistics	Transportation and storing of work materials.
Operations	Transformation of work materials into products.
Outbound logistics	Transportation and storing of products.
Marketing and sales	Communication and sales of products.
Service	Enhancement and maintenance of products, customer care.
Support activities	**Meaning**
Firm infrastructure	Strategy, finance and information management.
Human resource management	Recruitment, management and development of employees.
Technology development	R&D and process development.
Procurement	Acquisition of resource inputs.

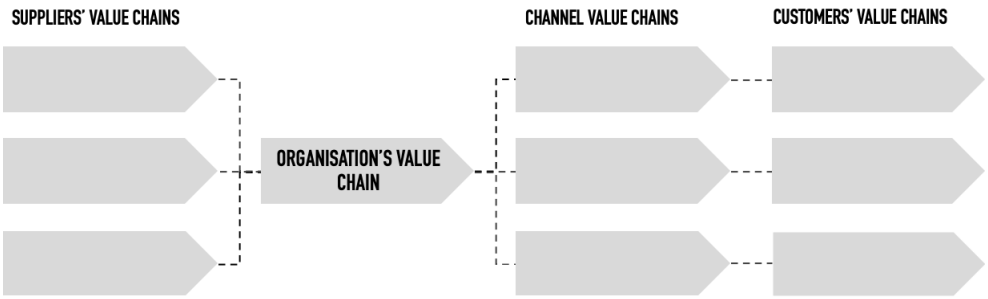


Figure 2.4: The value network [21].

2.3 Software Development Life Cycle

Software development life cycle (SDLC) is a systematic approach to organise the development of a software product. It covers all the various phases a software project goes through from initiation to implementation and maintenance. Below follows a description of the general phases of SDLC, some common SDLC models and professional roles that are active in different approaches to SDLC [1].

2.3.1 Software Development Life Cycle Phases

There are several different SDLC models between which the sequence and the extent of the phases varies. In Figure 2.5 the main phases of the various SDLC models are illustrated [1].

Figure 2.5: Main phases of SDLC [1].

Recognition of Need

Initiation and planning of the project. During this phase the focus lies on defining the problem or the need the product resulting from the project is aimed to solve. An investigation of the situation is carried out to set references for the project team to determine what the existing shortcomings are and how these should be addressed. It is during this phase the necessary requirements of the product are investigated [1].

Phase 2: Feasibility Study

Project identification and selection by investigating and evaluating user needs versus available resources. The study investigates the project's feasibility from an organisational, economical, technical and operational perspective in order to give a realistic picture of the context of the project. The study should consider alternative projects or solutions in order to determine the optimal choice [1].

Table 2.2: Feasibility perspectives [1].

Perspective	Meaning
Organisational feasibility	Alignment with organisational strategic objectives.
Economic feasibility	Economic implications of proposed project.
Technical feasibility	Alignment with existing technical solutions.
Operational feasibility	Alignment with solution stakeholders, e.g. end users.

Phase 3: Project Analysis

Analysis of relevant system operations and their correspondence within and beyond the system. Iteration of activities such as examining information flows and evaluating design options to achieve a preferred solution is key and needs to be conducted in cooperation with affected users. The analysis should render specifications that meet user requirements and contains recommendations for the new solution [1].

Phase 4: System Design

Design of the system. The final product and the process by which it will be achieved is described and mapped out [1].

Phase 5: Coding

Implementation of the system design i.e. translation of system design into code in a particular language. This should be done in a structured way since the quality of the code affects the understandability of the product and hence also how easy it is to test and carry out maintenance [1].

Phase 6: Testing

Quality control of conducted work. By testing the software errors within requirements, design and coding, errors can be detected. Faults should be reported in test and error reports, which describe what and how the testing has been carried out and what errors were detected [1].

Phase 7: Implementation

Realisation of the product. Programmers, users, operations management and system analysts are all involved in one way or another in order to put the product into use. Providing security, final testing and user satisfaction monitoring are examples of actions necessary to carry out during this phase [1].

Phase 8: Maintenance

Modification of the product after implementation. Changes of the product after its launch are made in order to either correct errors, adapt to external environmental change, enhance performance or prevent obsoletion of the product [1].

2.3.2 Traditional Models

Below, introductions to some of the traditional SDLC models are given [1]. In Table 2.3 the roles within traditional software development are presented. It is common for teams within traditional software development to consist of strictly defined roles, leading to their activities being carried out in isolated environments [43].

Waterfall Model

Popularised in the 1970s, the waterfall model has for a long time been the industry standard when structuring software development [1]. It is also known as the linear sequential life cycle model due to the fact that all the previous phases of the SDLC need to be completed before the next phase can be initiated [36]. Review of the conducted work in order to ensure compliance with the requirements is in the waterfall model not done until the end, when each phase has been completed. Execution of activities within each phase may be carried out by different people who in their turn lack insight into other parts of the project besides their own [1].

Prototyping Model

The prototyping model begins in the communication between customer and developer. Together both parties define the objectives for the project, they identify known requirements and areas that need further investigation. A design is proposed and from this a prototype is constructed, which further on is reviewed by the customer in order to identify additional requirements. These steps are iterated until the customer is satisfied with the solution. The prototype is not put into production until customer satisfaction is achieved and necessary requirements identified [1].

Spiral Model

The spiral model combines the iteration of the prototyping model with the systematisation of the waterfall model. The model is cyclic and iterative, with each cycle consisting of four stages. The aim of the first stage is to identify the objectives of the current phase and possible alternative solutions for this phase. The second stage focuses on evaluation of the alternatives from a risk perspective, given existing objectives and restraints. The third stage involves activities such as benchmarking, simulation and prototyping, in order to develop strategies to minimise uncertainties and risks. In the fourth and last stage the objective of what to be accomplished in the next cycle is determined [1].

Iterative Enhancement Model

In the iterative enhancement model the software is divided into several modules that are incrementally developed and delivered. The core module is first developed and successively added with additional functionalities and hence a more sophisticated product is created for each increment. User feedback on the current product is used as input for the development of the next increment, in order to create a product that better meets the needs of the customer [1].

Table 2.3: Roles within traditional software development and primary activities [43].

Roles	Activities
Project manager	Resource allocation and budgeting.
Software developer	Development activities.
Software tester	Test developed software.
User interface designer	Design screen interfaces.
Database designer	Data modeling.
Software architect	Software modeling.
Business analyst	Stakeholder management and documentation.
Requirement engineer	Gather requirements.
Software quality assurance	Create and maintain quality.
System analyst	System construction.

2.3.3 Agile Models

Agile software development emphasises cross-functional collaboration within the development team as well as with the customers. It focuses on constant change, rapid releases and iterative processes that are adaptive to their changing environment, enabling the team to always create topical products. Customer feedback is important in agile development in order to create products that match the customer needs. Below two common agile SDLC models are presented and in Table 2.4 and Table 2.5 the roles within each model are accounted for [22].

Scrum Model

Scrum focus on iterative and incremental processes that enables the development team to dedicate their efforts and activities to achieve established goals and not waste valuable time on less important tasks, e.g. documentation or equivalent bureaucracy. The model does not advocate certain development techniques, but rather the interaction between team members to enable team autonomy and flexibility. The project leader, also known as the scrum master, only establishes what goal the team needs to accomplish and then facilitates their cooperation as the team decides on its own how to develop and achieve the goal for each iteration [34].

Table 2.4: Roles within scrum and primary activities [43].

Roles	Activities
Scrum master	Manage scrum team.
Product owner	Product management decisions.
Customer	Evaluation of backlog items.
Scrum team	Organise itself for time boxed goals.
Management	Evaluate decisions and goals.
User	Evaluate system functionalities.

Extreme Programming Model

The philosophy in extreme programming (XP) is that the core activity of software development should be writing programs. Team collaboration is of importance as well as personal development of each team member. Four activities are considered being of importance: coding, testing of all modules, communication and collaboration between developers and client, design to build efficient systems and reduce unnecessary dependencies [3].

Table 2.5: Roles within XP and primary activities [43].

Roles	Activities
Programmer	Maintain and test software.
Customer	Manage business decisions.
Tester	Help customers with functional test cases.
Tracker	Feedback and estimations.
Coach	Team supervision.
Consultant	Guide team for problem solving.
Manager	Management.

2.3.4 DevOps and DevSecOps Models

DevOps is short for development and operations. It is a software development approach that focuses on continuous delivery, enabling customer feedback to be quickly included in the development process and new market opportunities being seized. The approach is based upon lean and agile principles including collaboration between the teams of development, quality assurance and operations [35]. DevSecOps is short for development, security and operations. When applying DevSecOps the emphasis lies on integrating security practices with software development and operations, see Figure 2.6. The result of this is that more secure products are produced in a more efficient way [24].

Within DevSecOps the overall security goal for the development team is to deliver secure design and implementation. To achieve this, threat assessment, security requirements and secure architecture should be applied during the construction of the products. There is a trade-off between the security quality of a software product and its time-to-market, as security suffers if the timeline to release is short, which often is the goal within DevSecOps. Aiming to balance this trade-off by integrating appropriate security controls within the development process is thereby of essence. Security requirements could be of help with this and also help to build consensus between the different departments of the organisation, as the business department otherwise may want to push a release that the security department obstructs. Hence, a minimum expected security requirement baseline should be defined as a part of the release plan. For the operations team the main activities are issue management of security incidents, environment hardening and operational enabling. A policy for handling vulnerabilities is an essential part of issue management, meaning processes for vulnerability and security incident response together with root cause analysis should be defined. The security purpose of quality assurance is to assess security issues related to the software, which

might lead to security risks in the finished product. Such issues can be code-level vulnerabilities, logical errors or misconfigurations. Key security activities in quality assurance are reviewing of design, implementation and carrying out security tests. For review of third party components, it is recommended to have a third party software evaluation checklist with criteria for introducing such components and to have an internal third party component database available for cross-referencing between projects. It is also recommended to keep track of CVE status of integrated third party components and to ensure security patch updates of such components are a part of the routine tasks [12]. To succeed with DevSecOps and deliver secure products, the five principles mentioned in Table 2.6 should be applied [24].

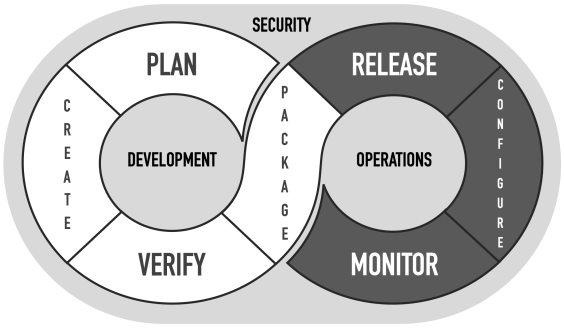


Figure 2.6: The DevSecOps approach [32].

Table 2.6: Principles of DevSecOps [24].

Principles	Activities
Build security	Build in security controls as well as security practices.
Adopt an enabling attitude	Create synergy between the different departments.
Integrate continuous learning	Learn from mistakes and evaluate vulnerability root causes.
Promote open collaboration	Involve the security team throughout the process.
Share threat intelligence	Share knowledge of vulnerabilities and threats.

2.4 Security Risk Management

The purpose of information security is for the responsible organisation to ensure confidentiality, integrity, availability and accountability of its resources [42]. Security standards and policies vary between different organisations and so does the level of assurance. Hence a general solution to create more secure products for all organisations to apply is difficult to achieve. Threats can target data in its various stages, while it is at rest in storage, being processed by software components and in transit being electronically transmitted, which makes risk analysis vital during each data stage [42]. Risk management (RM) is "the identification, assessment, and prioritisation of risk followed by coordinated and economical application of resources to minimise, monitor, and control the probability and/or impact of unfortunate event" [13]. According to Foreman [8] there are three options when addressing risks. Retention is to accept the risk by not taking any actions towards it. Mitigation is to take actions that prevent the risk from happening. Reduction is to take actions that reduce the

consequences if the risk happens. Wheeler [42] adds an additional fourth option, which is to transfer the responsibility or liability for the risk to another party.

2.4.1 Vulnerability Management

Vulnerability management (VM) is "the cyclical practice of identifying, classifying, remediating and mitigating vulnerabilities" [8]. The purpose of VM is to discover any existing coding flaws that might render risks or might be exploited, for the organisation responsible of the code or any organisation that possibly can be affected. VM includes live monitoring of the environment for arising vulnerabilities. Whether the software and system configurations are integrated in organisations within the private, the public or the military sector, VM is to be considered a core activity in order to counteract potential threats of vulnerability exploitation by malicious actors. In regards of its importance as an IT security disciplinary, it has in general not been thoroughly prioritised or applied by the industry [8]. Despite the cyber threat vulnerabilities pose to the developing companies and their customers' activities, the field of VM concerning communication and collaboration between companies and their customers is fairly unexplored [30]. The difference between RM and VM, is that VM can be considered a segment of RM where the focus lies on vulnerabilities related to technical software and system configurations [8].

HAVOSS is a maturity model for handling vulnerabilities in third party components, based on well-known software security and software development life cycle maturity models. HAVOSS, depicted in Figure 2.7, provides a framework for handling vulnerabilities and consists of the following capability areas: product knowledge, identification and monitoring of sources, evaluation of vulnerabilities, remedy of vulnerabilities, delivering updates and communication. Product knowledge is the company knowledge of their product components and a prerequisite for the other capability areas, as without it efficient handling of vulnerabilities is unlikely. Identification and monitoring of vulnerability sources are the vital activities surrounding the sifting of the constant information flow for newly discovered vulnerabilities. Evaluation of vulnerabilities is the capability area focused on assessment of the company's capability to evaluate severeness and relevance of discovered vulnerabilities [27]. A vulnerability assessment is often conducted through a scanning or configuration analysis, in which vulnerabilities are identified and rated, but in general no analysis of threats or applicability is done simultaneously. Thereby a vulnerability assessment is not equal to a risk assessment. Several risks can be connected to a single vulnerability and each one of them can have different risk exposure ratings. Vulnerability assessments include checking software patch levels, deficient security controls, weak authentication credentials, vulnerable protocols and unauthorised or vulnerable services, software or configurations. Identified vulnerabilities should be registered in a central repository and depending on the prioritisation of the vulnerability, action to fix it should be taken. Storing of the vulnerabilities and the handling of it should be carried out for historical trending. Any data concerning vulnerabilities and their implications for the organisation should be handled according to the organisation's data classification and handling policy. Depending on the CVSS scoring of the vulnerabilities, the mitigation of them should be prioritised according to their severity. Figure 2.8 presents a vulnerability qualification workflow for sifting detected vulnerabilities based on their severity scoring [42]. Remedy of vulnerabilities is the capability area of addressing the vulnerabilities and deciding the most efficient and fitting measures. Vulnerabilities can basically be divided into three cat-

egories based on their required remedy: those in need of urgent changes, those that can await patching until the next release and those that do not require any fixing. Delivering updates are the activities surrounding deployment of new firmware or updated software and making sure this is actually carried out by the users. Finally, external and internal communication of vulnerability and security information creates security awareness and contributes to more secure products. In Table 2.7 the different types of communication regarding vulnerability and security information are described [27].

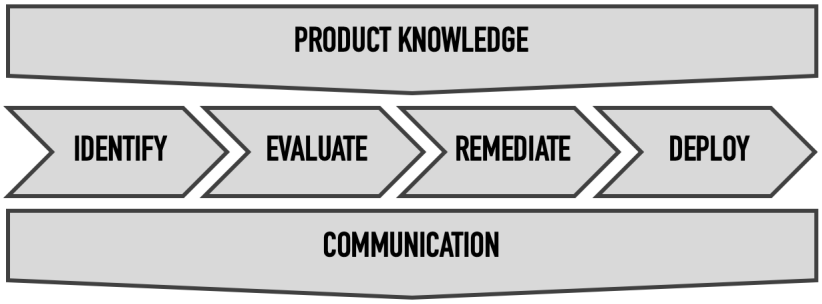


Figure 2.7: The HAVOSS model [27].

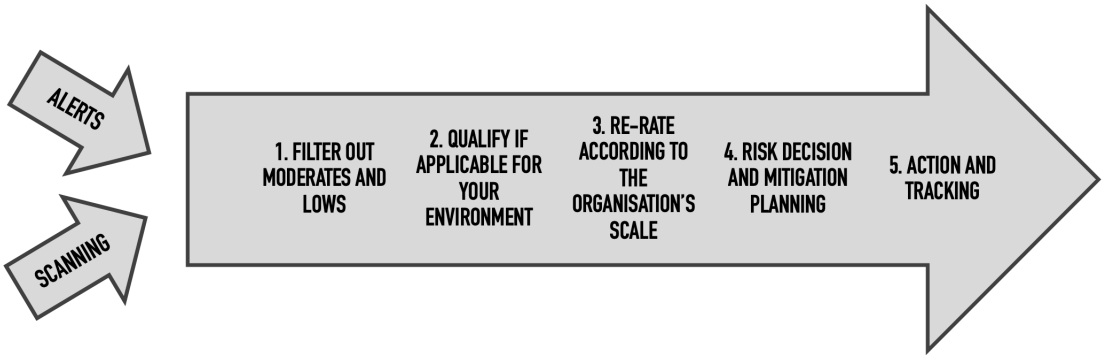


Figure 2.8: The vulnerability qualification workflow [42].

Table 2.7: Communication of vulnerability and security information [27].

Communication practices
CI1. Internal communication when vulnerabilities are identified, e.g. inform employees.
CI2. External communication when vulnerabilities are identified, e.g. public advisories.
CI3. Communication with media when vulnerabilities are handled.
CI4. Communication with important customers about critical vulnerabilities.
CI5. Information to customers about the patching status of products.
CI6. Transfer of related security information during delivery of patches, e.g. guidelines.

2.4.2 Security Organisation

The scope of security practices often varies with the business growth of a company. As an example, a startup is more likely to hire external help to protect its services and data while a larger enterprise may build a customised security program based on its own business needs. This means that the security organisation structure also varies with a company's business growth. At first there might not even exist a dedicated security team within the company, as focus may be on creating security controls and rely on external security tools. As the business matures, the focus can shift into setting up a security testing team, while the development team starts applying secure coding methods earlier on in the development process. Besides having its own security testing team, a monitoring team might also be instituted and tailored security services be developed within the company. By then, the company's security assurance program may be covering partners and the ecosystem of the company as well. Security organisations can be structured in different ways. One alternative is to institute a security office directly under the CTO, as depicted in Figure 2.9. Implications of such structure are that there is no dedicated chief security officer (CSO), the security team might be quite small and assigned to consult the company's ongoing projects. Another alternative is to institute dedicated security teams under management of an official CSO. Functions such as security management, testing, engineering, monitoring and services contribute to incorporate security within the company. Figure 2.10 depicts this kind of structure and Table 2.8 describes the activities of each function [12].

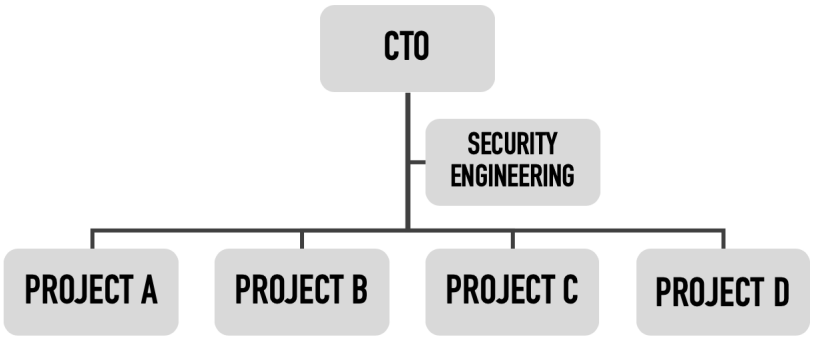


Figure 2.9: Security office under a CTO [12].

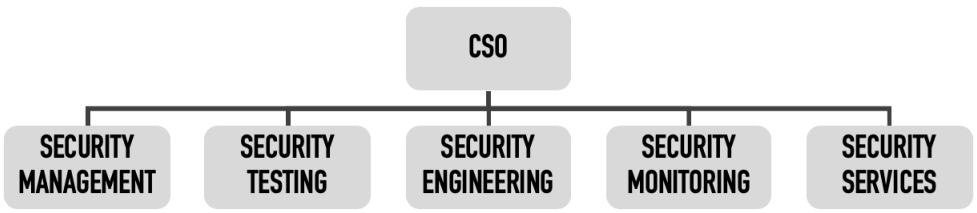


Figure 2.10: Dedicated security teams under a CSO [12].

Table 2.8: Activities of security functions [12].

Function	Activities
Security management	Defines security guidelines, processes, policies and requirements.
Security testing	Performs security testing before release of in-house products.
Security engineering	Provides security framework, architecture etc.
Security monitoring	Monitors security status for online services.
Security services	Develops security services.

2.4.3 Security Design Principles

There are three essential principles when it comes to designing security standards, guidelines and controls: least privilege, defense in depth and separation of duties. The least privileged principle states that no user should have access to excessive information that is not needed for them to carry out a specific chore or function, no communications or activities should be possible to execute unless there is a specific need for that transaction or access. The defense in depth principle states that multiple security techniques or layers of controls are necessary to reduce the risk of exposure. In case of one of the security controls getting compromised the others will provide protection despite the first one being faulty. The separation of duties principle states that authority to perform privileged functions should not be reserved for a single person or team, due to the risk of exploitation of access privileges for personal gain and prevention of single points of knowledge. This is important especially regarding functions for creating and handling sensitive information [42].

2.5 Software Requirements

Software requirements (SR) are essential for all SDLC models in order to ensure creation of high-quality products while developing new software. SR are the capabilities the suggested software product should have in order to fulfill its purpose and the user needs. If there are any errors in the SR, these will likely be present as defects in the finished product code. A software requirements specification (SRS) is a way of ensuring the correctness of the SR as it bridges the communication gap between customers and developers and can be used as a reference for validation of the final product. It also reduces development costs as it decreases errors. In Figure 2.11 the process of determining necessary SR is illustrated. The process mainly consists of three stages that overlap each other and between which there are feedback loops: problem analysis, product description and validation. During problem analysis the aim is to understand what the software need to perform and this is primarily done via meetings with the customers. During validation the quality of the SRS is controlled as well as its content, to make sure all needed SR are included in the specification [20].

2.5.1 Types of Requirements

Below, the basic contents of a SRS are described: functional requirements, performance requirements, design constraints and external interfaces [20].

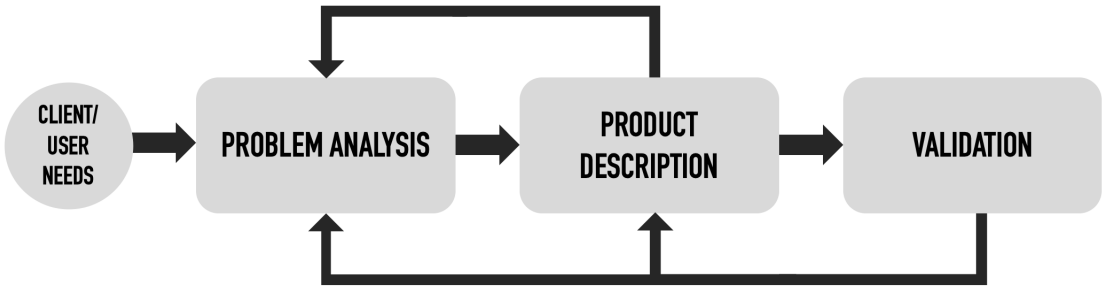


Figure 2.11: The system requirement process [20].

Functional Requirements

Functional requirements specify information about the behaviour of the software in terms of what the corresponding data output are to the given input. The specification should contain technical information such as what the necessary operations are in order to achieve the intended output or what the formula computing the output is. Normal as well as abnormal input need to be detailed, so the consequential behaviour of the software for abnormal situations is also given [20].

Performance Requirements

Performance requirements specify the performance constraints of the software. There are two types, static and dynamic requirements. The static requirements are also called capacity requirements and does not enforce any constraints on the execution performance of the software. An example of a static requirement is the supported number of terminals. The dynamic requirements enforce constraints on the execution performance, e.g. specifies response time and throughput constraints in measurable and quantifiable terms [20].

Design Constraints

Design constraints are factors such as resource limits, mandatory standards or policies that must be followed, hardware limitations, reliability and fault tolerance requirements. Security requirements are also considered as design constraints, as they can restrict the use of certain commands, control access to data, demand different types of access requirements depending on who the user is or require proper assessment of security threats [20].

External Interfaces

In the external interface the interactions of the software with external parties such as users, hardware and other software are specified. It describes content such as the characteristics of the user interface as well as the user manual, the hardware memory restraints and the operating system [20].

2.5.2 Software Product Quality

According to the ISO/IEC 25000 series of standards, software product quality comprises eight dimensions: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability. These dimensions are referred to as System and Software Quality Requirements and Evaluation (SQuaRE) [19]. Reliability is considered being the major dimension. If a software product is deemed unreliable it is usually because of the occurrence of defects in the coding. An exact definition of defects is difficult to establish as this can vary between different projects and companies. But a general definition of software defects is that they are problems in the software that cause the software to crash or create incorrect outputs [20]. Functional suitability is how much a product or system's functions meet the requirements described in the product requirements specifications. Performance efficiency is the performance of the product given the amount of resources used. Compatibility is how much a product can exchange information with its surrounding components according to its required functions, while sharing the same hardware or software environment. Usability is how much a product can be used by specified users and achieving specified goals in an effective, efficient and satisfactorily way. Security is how much a product protects data so its users, either people or objects, have access to information that is appropriate to their types and authorisation levels. Maintainability is how much a product can be modified to changes in requirements or its environment. Finally, portability is how effectively and efficiently a product can be transferred from one operational environment to another [26].

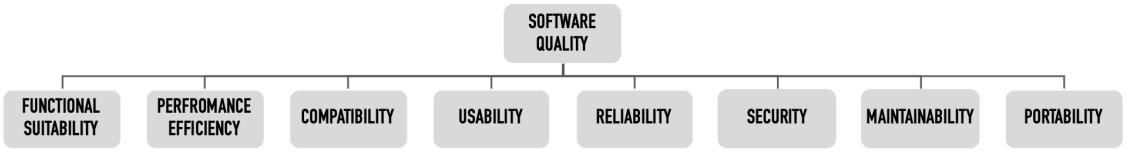


Figure 2.12: Dimensions of software product quality [19].

Chapter 3
Methodology

The Methodology chapter describes the process of the study and the main methods applied during it. The motivation behind each methodological decision is accounted for, as well as a discussion of whether the chosen methodologies will result in a credible study.

3.1 The Outline of the Study

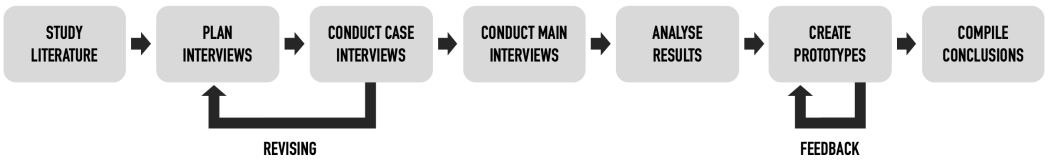


Figure 3.1: Outline of the study.

The initial information gathering in the study was conducted via literature search, in order to anchor the study in existing research. Information was also gathered from Debricked and the Hatch project, to gain more industry specific knowledge. In the beginning of the thesis it was desirable to gather as much information as possible about vulnerabilities to be able to break it down before dividing it among different roles. Terms used to search for information about vulnerabilities were vulnerability management, vulnerabilities, open software, open source, risk management. After the initial literature study, an interview guide was created in order to start interviewing respondents from different companies. The interviews were structured into two rounds, first case interviews and then main interviews. During the two case interviews the initial interview guide was tested and critical variables were identified. Later the interview guide was modified and ten main interviews conducted. The data gathered through the interviews was then compiled and analysed. To analyse the collected data, the first step was to quantify what type of recipients of vulnerability information that

were found and what type of vulnerability information that was needed in general. The next step in the analysis was to group all the different roles into different recipient groups using theoretical models, as well as connecting the information to different roles and functions described in theory. Prototype views were derived from the analysis results. The prototype views were later shown to the interview respondents and their feedback was the basis for the evaluation of prototype views. Conclusions were formulated along with the prototype views. In the discussion part the credibility of the study was discussed, as well as the validity and possible factors that could have influenced or affected the results of the study.

3.1.1 Research Purpose

The selection of what methodology to use should be based upon the purpose and objectives of a study. Depending on the purpose a study can be of descriptive, exploratory, explanatory or problem-solving character. Some studies can consist of more than one study type, to address different purposes of the study [15]. The different study types are further presented below.

- **Descriptive Study** A descriptive study aims to find out how something works or is being executed, and then describe that phenomenon [15]. This method is suitable when there is a clear picture of the phenomenon for which it is desirable to collect data, before the data is collected [33].
- **Exploratory Study** An exploratory study aims to dive deeper into how something works or is being executed [15]. This method is used to seek more insight on a matter or to gain clarity of a problem. An exploratory study is mainly conducted through a literature study, or by interviewing experts or focus groups [33].
- **Explanatory Study** An explanatory study aims to find causal relationships or correlations between variables and explanations for how something works or functions [15]. This by studying a situation or problem to further on be able to explain the relationship between variables [33].
- **Problem-solving Study** A problem-solving study seeks to find a solution to an identified problem. It is a common method at Technical Universities, in combination with other study types [15].

Selected Study Type

The purpose of this study is to identify certain variables that are critical in vulnerability management and pair this information to certain roles or functions within a company. In this thesis a combination of exploratory and explanatory methods was used. The variables and roles that were considered critical were identified and described through an exploratory study. After that the relationship between them was explained with an explanatory method.

3.2 Research Approach

There are two main research approaches that can be used when conducting a study, the deductive approach and the inductive approach. Moreover, there is also a combination of the

two approaches, called the abductive approach. These research approaches are further explained below [33].

- **Deductive Approach** The deductive approach is when ideas or theories are identified through literature and later tested by using data. A theoretical framework is set up and a hypothesis is formulated, that later is tested to be proven wrong or right via a research strategy. The deductive approach is normally used in scientific research and is the dominant approach within natural sciences [33].
- **Inductive Approach** The inductive approach is not so much about testing already existing theory but to develop new theories. This by observing and collecting data through a research strategy and then from the findings derive a theoretical framework. This is a research approach common within social sciences that often involves qualitative data [33].
- **Abductive Approach** As stated earlier, the abductive approach is a combination of the deductive and the inductive approach. This is often an advantageous approach since both research approaches complement each other [33].

Selected Research Approach

In this thesis the abductive research approach was used to benefit both from the deductive and the inductive approach. An example of how the deductive research approach was used was when theoretical models of organisational structures were hypothesised to be applicable when categorising vulnerability information and then later were tested in order to determine if this was a correct hypothesis. An example of how the inductive research approach was used during this study was when theory about critical vulnerability information variables was built through studying different companies.

3.3 Research Strategy

For a master thesis within applied sciences, four different research strategies are considered relevant. These are survey, case study, experiment and action research [15]. However, it is not uncommon that more than one strategy is used in a study [33]. The four identified relevant research strategies are presented below.

- **Survey** If the aim of the study is to describe a phenomenon, a survey might be suitable. It is a compilation of the current state of a matter, also known as a status analysis. A survey is often linked to a descriptive or explanatory purpose. When a survey is conducted a certain selection group is investigated. If this group is very large, the survey is sent to a share of the whole group, commonly called a sample. Surveys are often conducted through standardised questionnaires, where the data collected is analysed through quantitative methods [15].
- **Case Study** A study that aims to explain a phenomenon or an object on a deeper level might use the strategy of a case study. A case study can for example be done in an organisation, to see what the work processes look like. Case studies can give more in

depth knowledge than a survey and are mainly done through interviews, observations or by analysing archives. Interviews can be structured, semi-structured or open. The data gathered through case studies is mainly qualitative. If the different case studies are not picked through random sampling, the result cannot be argued statistically significant. However certain patterns might be revealed if several studies are conducted [15].

- **Experiment** To be able to explain the occurrence of a certain phenomenon or to find causal relationships, a more structured research strategy than survey or case study is necessary. Such can be conducted through experiments. Experiments can compare technical solutions to each other, but also investigate people and behaviours. Usually the gathered data is quantitative and the experiment is performed from a fix design that includes a hypothesis, variables and subjects [15].

- **Action Research** When the purpose of a study is to improve something as well as studying a phenomenon, the action research strategy might be suitable. This approach is useful when a thesis is of a problem-solving character. When using the action research strategy the first step is observation, to get more clarity on the specific problem. The survey or case study method might be used here. After the observations are made, a solution is implemented and later the chosen approach is evaluated through an iterative process [15].

Selected Research Strategy

In this thesis the case study was chosen as the main research strategy, as well as a survey. More specifically the study was conducted through interviews. The case study strategy matches well with the purpose of the thesis, where the objective mainly is to derive qualitative information. The case study was conducted through two case interviews that were more open, to gain more knowledge and insight into the topic. The interview guide was then adjusted to the findings and used during a second round of interviews, the main interviews. The main interviews were more structured than the case interviews and thus more surveylike, with explicit questions being established and answered. As stated earlier the case study stretched over multiple cases, this to be enable comparison of the findings.

3.4 Case Selection

Since the study is based on previous research done within the HATCH-project, it was a natural first step to use two companies that respectively operates as supplier and integrator to investigate the roles and functions from a more generic perspective than done previously. Therefore, this was a criteria for the first two companies participating in the case interviews, together with the requirement that it is software containing open source code that they are supplying or integrating.

The selection of companies for the main interviews is ongoing throughout the study. Since the objective is to create a generic result, the requirements for the candidates at the main companies were that they in some way come across vulnerabilities in OSS in their

job. If this was the case, the company at which they were working were deemed as appropriate in regards of the study purpose. All company sizes, sectors and industries were of interest. The participants during the main interviews were contacted in various ways, some were reached out to through recommendations from previous interviewees, some were recommended through contacts at the main company and some candidates were found at the Software@LTH fair. The reason for the selection being ongoing was that was is time-consuming to find enough suitable individuals for the interviews to meet the target of ten main interviews. Secondly, keeping the selection ongoing was a way to adapt the interviews to the data that already had been collected and to choose roles or functions that were not as thoroughly covered in the data as other roles.

3.5 Data Collection

3.5.1 Qualitative vs. Quantitative Approach

There are two main types of data, quantitative and qualitative [33]. Both data types are further explained below.

- **Quantitative Approach** Quantitative data is data that can be quantified and expressed in numbers. It is standardised, numeric and the sample sizes need to be large to reach a statistically significant result in order to avoid the errors that might occur in smaller sample sizes. Quantitative data is commonly visualised through diagrams and tables, so the results are presented in a clear way and can easily be interpreted [33].
- **Qualitative Approach** Qualitative data is data that is usually detailed and profound, but not standardised nor numeric [15]. Qualitative data is usually presented in a text format, through descriptions. To be able to interpret and analyse the qualitative data, it might need to be grouped into different categories [33].

Selected Data Collection Approach

For this thesis the qualitative approach was most suitable. The objective was to get more in depth knowledge and insights through several case studies, thus it tended to fall naturally that the used method was of qualitative character. To be able to analyse the data collected in this thesis the data was grouped into categories that overlapped or answered the research questions, which in turn were based upon the variables that were identified as critical during the case interviews.

3.5.2 Interviews

During the study two rounds of interviews were conducted: the case interviews, referred to as C1 and C2, and the main interviews, referred to as M1-M10. In total 12 rounds of interviews were conducted. The first round of interviews was more open and focused on identifying critical variables when it came to communication of vulnerabilities to be able to adapt the interview guide for the second round of interviews to a more structured, survey-like approach. This to be able to a smaller extent quantify the identified roles and what type

of information each recipient need. All interviewees were anonymous throughout the study and only characteristics such as industry type, industry role and company size were mentioned. For the company sizes, according to the Swedish agency Upphandlingsmyndigheten, the Swedish definition of company size categories are the same as the ones stated by the European Commission in 2003 [40]. The defined interval for each size that is used to categorize the interview respondent companies, is presented in Table 3.1.

Table 3.1: Definition of company size [40].

Size	Number of employees
Micro	< 10
Small	< 50
Medium	< 250
Large	> 250

Case Interviews

During the initial literature study the areas of risk assessment, risk treatment and risk control were identified as suitable areas to divide the interviews into, to gain more knowledge about what the processes of handling vulnerabilities looked like within each company. Adding to that some questions about the background of the interviewee and company to get more context to the answers such as their own roles and functions, company size and if the company is primarily operating as a supplier or integrator. The next part of questions in the interview guide was derived directly from the research questions: The recipients of vulnerability information, what information each recipient needs and how information can be presented in an efficient way. How the organisation is handling vulnerabilities was asked as an open question early on during the case interviews, to see how the respondent themselves would explain their processes and policies regarding vulnerabilities in third party components. Once the two case interviews were conducted, the outcome was analysed and the interview guide was modified before starting the process of the main interviews.

Main Interviews

As for the main interviews, the format was slightly stricter and more standardised than during the case interviews, but still semi-structured. The interview guide was modified to better fit the purpose of the study and in order to address interesting findings from the case interviews. The modifications were the following: Adding questions regarding requirements, both what requirements the companies themselves have towards their suppliers, but also what their customers require. During the case interviews it appeared that it is not unusual for a customer to require full reports on how the supplier is working with vulnerability management, what vulnerabilities there are in the system right now and data on the last penetration test and latest updates. Also questions regarding the presentation of vulnerabilities were added. It occurred during the case interviews that the respondent explained how information was presented within the company. This was also added as a question, to give insight into how information currently is presented within organisations and what the respondents

think should be improved. After modification of the interview guide, the second round of interviews was conducted with the same approach as in the case interviews.

3.6 Analysis of Interview Results

The initial step after the interviews were conducted was to compile the data gathered. To do this, the word analysis program NVivo was used. With the help of NVivo the mentions of different roles that came across vulnerability were compiled, together with the information that was mentioned during the interviews. The vulnerability information collected was then sorted according to different steps in the HAVOSS model. The mentioned sources for vulnerability information and preferred presentation forms of vulnerability information was also compiled. This gave an overview and insight to dive deeper into in the analysis. The analysis section was primarily divided into four parts: Vulnerability Management in Practice, Recipients: Theory vs. Reality, Recipient Groups and Information Need and Prototype View Design. All areas are further explained below.

3.6.1 Vulnerability Management in Practice

The section of Vulnerability Management is a general comparison with what theory says and what the findings during the interviews are regarding vulnerability management in practice. In the analysis certain aspects that can affect how organisations and roles work with vulnerability management are raised and analysed. This section provides context to the further findings of vulnerability recipients and vulnerability information, along with a better understanding of what the gaps and similarities are between theory and work in practice.

3.6.2 Recipients: Theory vs. Reality/Recipient Groups and Information Need

The interview results were analysed in NVivo. Certain professional roles were identified and grouped based on the information they need according to the respondents. The grouping of recipients was evaluated and compared with theory, to make sure the differences and similarities between theoretical and empirical findings were highlighted in order to create a more transparent and credible study. The six recipient types that were identified were elaborated on separately, and also summarised in a table to give a better overview of the analysis results. The analysis findings lead to the construction of a new model for grouping vulnerability recipients: The Vulnerability Information Recipient Onion Model (VIROM). The model was derived mainly from the interview results, but also anchored in existing theory.

3.6.3 Prototype View Design

Many master theses' aim to develop a part of a product or methodology. What these kind of studies have in common is that the specifications are not known in the beginning of the study. Instead the specifications evolve during the study through different prototypes. The prototyping starts with an idea and a first specification of the prototype. Then the prototype

is evaluated, specifications are modified and a new prototype is created. This process is then repeated until a final prototype is developed. The specifications can initially be quite broad, getting more and more specific towards the end of the process [15]. To address the objective of conducting an interactive platform for information sharing together with Debricked, the chosen approach to do so was prototyping. From the analysis of the findings during the interviews and in theory, a first prototype was created. This first prototype consisted of six different prototype views, containing tailored information specifications for six different groups of recipients. The design of the prototype views was based on design templates from Debricked, to make the prototype views as realistic as possible.

3.7 Evaluation of Prototype Views

The prototype views designed after the analysis of the interview results were evaluated by the interview respondents. The prototype views were sent out to the respondents together with a form, through which their feedback was collected. The different prototype views were evaluated in terms of enough or not enough information, correct information and grouping of recipients. The respondents were given the possibility to also contribute with any specific feedback they would like to give. The feedback was then compiled.

3.8 Credibility

The credibility of a study is important for evaluating the conclusions. If the study has a high credibility, the chance of the answers being incorrect is lower [33]. There are different aspects that decide whether a study can be considered credible. The conclusions need to be drawn from a good set of data, the results need to be generic and the study addresses the phenomenon that is studied. The categorises that make up the credibility of a study are validity, reliability and representativeness [15]. All three aspects are further elaborated on below.

3.8.1 Validity

The validity of a study concerns the connection between what object or phenomena that is observed and what that is actually measured in the study. To generate a higher level of validity the method of triangulation can be used, where an object is studied with different methods and thus lowers the risk of the measurement being incorrect. Logging and documentation of decisions and activities during the working process is necessary for the validation of the study. During the study it might become necessary to look back into earlier steps of the study process. It is particularly necessary to save primary data, since this is the fundamental base of the finalised conclusions [15]. This is why all primary data from the interviews were attached to the report as appendices.

Another way to ensure that the study has a high validity, is to get feedback from, as in this case, the respondents to see if the prototype views and results reflect the information they provided. It is important to point out that it is the primary data that is being verified through feedback and not the analysis, which the respondents do not necessarily have to

agree on. To further create a study with high validity, a method for this is to let a third-party continuously review the study critically, so called third-party reviewing [15]. Third-party reviewing was done throughout the entire study process, mainly by the supervisor and examiner of the thesis.

To conduct a study during a longer period can both benefit and threaten the validity of the study. Longer studies can give a deeper understanding but may result in a very narrow view, while short studies might not capture the complexity of the issue [15]. This study was conducted during a period of more than 20 weeks and can therefore be considered a long-term study. To avoid narrowing down the study too much, the study was broadened with three main research questions that all were given time to investigate in depth. Along with this the delimitations were set clearly from the beginning of the study. To obtain a study with high validity, the different steps in the working process were carefully documented. Also, feedback was requested from the interview respondents to make sure that the data collected reflected their perceptions and opinions. The validity of a study can be divided into internal validity and external validity. These are further elaborated on below.

- **Internal Validity** Internal validity refers to how trustworthy the findings are [33]. The internal validity depends on the extent of systematic errors and often refers to experiments where errors can be clearly defined [15]. Using mostly qualitative research methods, the systematic errors are not as easy to define as in an experiment but may be explained by biases and errors presented below in section 3.8.2 Reliability.

- **External Validity** External validity addresses to what extent the findings and conclusions can transfer to other contexts, which in turn reveals if the findings can be generalised or not [33]. This is further described below in section 3.8.3 Representativeness.

3.8.2 Reliability

The reliability aspect of the total credibility of a study addresses the reliability of the data used to draw conclusions, as well as the method for analysing the data. For example, the data gathered is considered more reliable if the samples are chosen randomly. The reliability of a study also depends on how clearly and systematic the reader can follow the different steps taken in the study and evaluate them thoroughly [15]. There are some potential threats when looking at the reliability of a study. The object or person can be uncommitted to answer questions as good as possible, the subject or person can give an answer that is biased, or answer what they think the interviewer wants to hear. A study can be exposed to an observer error, especially if there are more than one person holding the interviews, asking questions in different ways that might affect the response of the contestant. Lastly there is also an observer bias, which means that the interviewers themselves are biased and can interpret the facts in various ways [33].

When it comes to the reliability of this thesis, it might have been exposed to some of the threats mentioned above. The interviews were always conducted by two people: one asking the questions and one taking the notes, but taking turns in doing so. Thereby the same two people were always present and might have set a similar environment for the respondents. Trying to avoid the observer bias, the one taking notes was fully focused on writing exactly what the interview object was saying. Then this data was interpreted and analysed, but in groups that was already decided in beforehand. It is thus likely the study was exposed to a bit

of observer bias, also since the interviews were semi-structured and could differ a bit from time to time.

3.8.3 Representativeness

The representativeness of a study concerns whether the conclusions can be generalised or not, which depends widely on the samples. If there is a bigger loss in the sample, it can affect the representativeness, especially if there is a certain group within the sample that falls short. One can argue that case studies are difficult to generalise. On the other hand, if several case studies are executed and the specific circumstances are described in detail, the larger the chance is that the results will be similar if put into a new context [15]. In this thesis the purpose was to find generic patterns, that could be applied when designing an interactive platform for many different business types. It is difficult to argue that the representativeness was high for all possible industries and that all possible industries were equally well represented in the sample. Case studies are in general not considered representative. However, since several case studies were conducted the chances are increasing that the results derived from the cases could be valid in other contexts as well. In this study certain parameters were prioritised when conducting the sample, to extend the representativeness. These were parameters such as company size, sector and professional roles of the interview respondents. With these variables fixed, the sample was then randomised.

Chapter 4
Interview Results

The Interview Results chapter contains the presentation of the empirically collected data from the case and main interviews with participating companies.

4.1 Participating Companies

A total of 12 companies have participated in the study. Two companies, named C1 and C2, participated in the case interviews. Of the 32 contacted companies for the main interviews, ten accepted the invitation to participate in the study. The participators in the main interviews are named M1-M10. Below in Table 4.1 follows an account of the participating companies, which industries they are primarily active in, the professional roles of the respondents and the size of each company.

Table 4.1: Participating companies in case and main interviews.

Company	Industry	Professional role	Size
C1	Security	Technology and operations manager	Large
C2	Video surveillance	Security expert	Large
M1	IT services and consulting	Senior enterprise architect	Large
M2	Telecommunications	Head of engineering	Medium
M3	Software	Development manager	Medium
M4	Electronics	Manager of software security	Large
M5	Retail	Manager of software security	Large
M6	Software	Technical director	Large
M7	Financial services	CTO	Small
M8	Web development	Project leader	Micro
M9	IT services and consulting	Information security expert	Large
M10	IT services and consulting	Developer	Large

4.2 Recipients of Vulnerability Information

In Table 4.2 follows an account of which professional roles that are mentioned by the respondents as recipients of vulnerability information and somehow being involved in handling vulnerabilities. A total of 39 different professional roles are mentioned and a majority of them are mentioned by only one company. Those mentioned more than once are developer, product owner, customer support, board, CEO, CTO, project manager, IT team, software security operations, security expert and key account manager. This means that 72% of the mentioned professional roles are mentioned by only one company.

Table 4.2: Recipients of vulnerability information according to respondents.

Professional roles	# mentions
Developer	11
Product owner	7
Customer support	6
Board	5
CEO	4
CTO	4
Project manager	4
Software security operations	3
IT team	3
Marketing and sales	2
Key account manager	2
Security expert	2
Technical experts	1
Head of engineering	1
Senior enterprise architect	1
Security analysts	1
Technology and operations manager	1
Development manager	1
Operations manager	1
Operations technician	1
IT architect	1
Central product quality organisation	1
Incident manager	1
PR	1
Other IT security branches	1
Managers of bug bounty program	1
SPOC	1
Security team	1
Operations	1
Technical director	1
Line manager	1
Tester	1
Technical writer	1

Technical support	1
COO	1
Communication function	1
System architect	1
Management	1
Information security team	1

4.3 Vulnerability Information

4.3.1 HAVOSS Relations

In Table 4.3 the results regarding relevant vulnerability information for recipients are summarised and sorted according to which HAVOSS step the information can be considered being related to. The represented company sizes are also compiled for each information type.

Table 4.3: Vulnerability information sorted by HAVOSS step belonging.

Product knowledge	**Represented company sizes**
Technical information about the affected products	Large
How vital the affected functionalities are	Large
If the functionalities are active or default	Large
If the functionalities can be turned off	Large
Customers' business situation	Large
Security requirements of the customers	Large/Medium
Own business impact	Large
Identification	**Represented company sizes**
Scanning tools	Large/Medium/Small/Micro
Penetration test reports	Large/Micro
Supplier reports	Large/Medium
Customer reports	Large/Medium
Bug bounty program reports	Large
Evaluation	**Represented company sizes**
Evaluation of the vulnerability	Large/Medium/Micro
What type of vulnerability it is	Large
What customers are affected by the vulnerability	Large
What systems are affected by the vulnerability	Large/Medium/Small
What component version the vulnerability is present in	Large/Medium/Small
How the component is affected	Large
What functionalities are affected by the vulnerability	Large/Medium/Small
If it is possible to quantify the risks of the vulnerability	Medium
Quantification of the risks	Large/Medium
How exploitable the vulnerability is	Large
If the vulnerability can be exploited remotely	Large
Generic vulnerability score such as CVSS	Large/Small

Internal vulnerability score	Large/Micro
Remedy	**Represented company sizes**
What actions are needed	Large
What actions are being taken	Large
How to fix the vulnerability	Large/Small/Micro
Other fix alternatives	Small
Consequences if the vulnerability is not fixed	Large/Medium/Micro
When a fix will be available	Large
If it is better to await a fix from the OSS originator or create an own solution	Large
If it is possible to downgrade the component to an earlier version while awaiting a fix	Large
Company specific fix recommendations	Large
How to protect the customers while awaiting a fix	Large/Small
Deployment	**Represented company sizes**
When a fix will be deployed	Large
Patch advisory	Large
Communication	**Represented company sizes**
Technical and development near information for employees involved in the SDLC	Large
Less technically detailed information for other company employees	Large/Medium
Less technically detailed information for customers	Large/Medium

4.3.2 Vulnerability Information Sources

In Table 4.4 the sources from which vulnerability information is usually gathered from are accounted for. These sources are all mentioned in the interviews with participating companies. The represented company sizes are also compiled for each source type.

Table 4.4: Vulnerability information sources according to participating companies.

Sources	Represented company sizes
Suppliers	Large/Medium/Micro
Customers	Large
Reports from scanning tools	Large/Medium/Small/Micro
Reports from penetration tests	Large/Small/Micro
Bug bounty programs	Large/Medium
Internal communication	Large/Medium/Small
Vulnerability list subscriptions	Small
Manual monitoring of OSS websites	Large/Medium/Micro
Manual monitoring of media	Large/Medium
Manual monitoring of communities	Large/Medium
Manual monitoring of Facebook groups	Large

4.4 Presentation of Vulnerability Information

The majority of participating companies do not have any input to give on how vulnerability information currently is or should be presented to different recipients. The suggestions on presentation forms that are mentioned in the interviews are accounted for and described in Table 4.5.

Table 4.5: Ways of presenting vulnerability information.

Presentation form	Applications
Trend lines	Bringing attention to security issues within company management.
Dashboard	Summarising, making technical information easy to understand.
Speedometer	Depicting vulnerability severity to the rest of the organisation.
Graphic presentation	Enhancing security knowledge within the whole organisation.
Component overview	Summarising vulnerabilities on a component as well as system level.

Chapter 5
Analysis

The Analysis of Interview Results chapter contains the analysis of the presented data from the Interview Results chapter. The foundation of the analysis is the theoretical framework presented in the Theory chapter.

5.1 Vulnerability Management in Practice

There are several ways to work with vulnerability management in practice. How much the interviewed companies are working with vulnerability management and how conscious they are about the management of vulnerabilities varies. Several respondents acknowledge the fact that the knowledge of software security and vulnerabilities is in general not widespread within the industry. Also the level of knowledge varies quite much between different professional roles within a company and cannot always be tied to a certain function.

From the interview results it becomes clear that security practices and ways of handling vulnerabilities varies to a great extent between companies. The variation of approaches to software development amongst the participating companies conforms with the theory of SDLC and security risk management. As shown in theory, there exist a various amount of SDLC models. Some of these, such as DevSecOps, are more security oriented and thus more inclined to have formal policies for handling vulnerabilities. However, it is not uncommon that companies create their own way of working, either through tweaking some of the theoretical models or just making up their own rules. How much the different companies are working with vulnerability management might also correlate with what requirements their customers have and what the companies are offering and selling to their clients. Some companies have security and protection of data as their core business and thus the process of vulnerability management is more thorough and theory-like than companies that do not have security and protection of data as their main selling point.

Business growth and maturity of the participating companies' security organisations varies, as all respondents state that their companies are engaged in some type of vulnerability han-

dling and assessment of OSS components, but not all of them have specific policies for it. Larger companies seem to have more structured security organisations and outspoken strategies for handling OSS vulnerabilities. In the Theory chapter it is implied that factors such as company size, business growth and customer requirements have an impact on how a company works with vulnerability management. With so many factors affecting the organisation of software development and security practices, it becomes difficult to identify specific commonalities in regards of recipients of vulnerability information. Just because an employee has a certain role at a company does not mean that the employee has the same responsibilities and knowledge as other similar roles at other companies. The knowledge of vulnerabilities can differ widely along with the actual responsibilities of a particular role. It can also be assumed that the way of structuring SDLC and corporate security risk management does directly affect what recipient that need what information.

5.2 Recipients: Theory vs. Reality

The organisational value chain model, Figure 2.3, specifies the general functions involved with creating the value of a company's products. The roles that are mentioned by respondents are such connected to activities within firm infrastructure, marketing and sales, operations and service. All though some of the functions and roles mentioned by respondents do not naturally come to mind when considering software security and development, e.g. board, chief officers, customer support and communication functions, they are in many interviews presented as more or less needed liaisons in the processes surrounding successful creation of secure products and handling of vulnerabilities. Vulnerability information is thereby to be considered as a cross-functional concept and not isolated to only professional roles and functions connected directly to the technical aspects of software security and development.

In Table 2.3 the professional roles typically included in traditional SDLC models are presented, in Table 2.4 and Table 2.5 the roles of the agile models. Neither of the participating companies' responsibility division between professional roles in software development and vulnerability handling conforms completely with the roles presented in the theory models. Even though all of the mentioned roles are regarded as more or less active partakers by the respondents, some of them are mentioned more often. Examples of such roles are developers, product owners and different manager types. But despite some of these roles being mentioned often, the extent of the involvement and scope of responsibilities within vulnerability handling for the same professional roles varies greatly between companies. This makes it difficult to identify commonalities solely on the basis of professional roles that are suitable to design platform views after. Regarding the professional roles within security organisation this diversity is also prominently present, as well as the diversity of the structures of the security organisation.

From the interview results it is thus clear that the types of professional roles that are mentioned as partakers in the handling of vulnerabilities vary greatly between the different companies. As previously stated, 72% of the mentioned roles are in fact only mentioned by one company. This result is not completely unexpected, since the theoretical framework shows that the ways of organising and structuring SDLC are almost endless, hinting at the assumption that the diversity of organisation of software development is reflected in the organisation of handling vulnerabilities. The above leads to the conclusion that trying to

identify individual professional roles to design platform views for is indeed to be considered futile and some kind of grouping of recipients is necessary for the final results of this thesis to be contributing any kind of value.

5.3 Recipient Groups and Information Need

The analysis of the interview results with NVivo shows that some recipients are possible to identify as needing certain information. Through this analysis it is therefore possible to group the different professional roles into more prominent recipient groups, that are deemed appropriate for tailoring platform views to. To further specify what information each recipient group need, the vulnerability information categorised through the HAVOSS model is analysed and connected with the identified recipient groups.

Support Functions

Functions and professional roles such as marketing and sales, key account managers, customer support, technical writers and PR are reported to be involved in the process of communication of vulnerabilities within and between companies, especially to customers. At some companies these functions are only informed if a severe vulnerability occurs in order to be able to inform customers about the situation. At other companies these functions are always informed about vulnerabilities causing operational disruptions. In such cases they are responsible of informing customers about what actions they are required to take, e.g. give instructions if extra login authentication is required due to security reasons caused by a vulnerability. The vulnerability information these functions are said to typically require is the seriousness of the incident, when a fix will be available for customers and instructions of what the customer should do while awaiting the fix. In the HAVOSS model this role typically needs the information that fits under the Communication section, which can be summarised as less technically detailed information for customers or other areas of the company. The support functions also need some information from the remedy step, namely when a fix will be available.

According to some respondents, in order to bridge the technical knowledge gap between the originators of the vulnerability information and the final recipients of it, which can be internal stakeholders such as employees or external stakeholders such as customers and media, communication functions are often involved with producing statements fitted for the recipients. This by translating and filtering vulnerability information so it is not so technically advanced and thus more appropriate and easier for the recipients to interpret. These statements are communicated via various information sources such as emails directly to the customers, common sharepoints, meetings or published on the company website. The above implies that it might be necessary with division of the support function recipients into two kinds of recipient groups. The first recipient group address the need of filtering and forwarding vulnerability information, so it is appropriately communicated both internally and externally. This recipient group is further on referred to as Communication function and its prototype view as the Communication function view. The second recipient group address the need of informing support functions such as customer support and key account managers of what to say to the customers if asked questions regarding vulnerabilities and other incidents

by them. This recipient group is further referred to as Support function and its prototype view as the Support function view.

Management and Board

The CTO does at some companies take active part in the handling of vulnerabilities, contributing to evaluation and being part of the triage team. At other companies the CTO is only informed together with the rest of management and board if something highly critical occurs. CEO, board and other kind of chief officers are often only informed if an actual exploit occurs, or if a vulnerability is regarded important or severe enough so that they ought to be informed. The employees responsible for handling the incident can in such cases escalate critical information to higher instances if necessary. For management and board, the information should in general not be too technical and rather as straight forward as possible, giving an overview of the issue and the biggest risks. Management and board rarely interacts with customers, but it is mentioned by some companies that it is standard to always keep management and board informed about disruptions, so they can be prepared for answering questions from customers. The management and board can therefore benefit from similar information as the support function in aspects of less technically detailed information, the step of communication in the HAVOSS model. Sometimes when the CTO has more technical knowledge it might also be of interest to receive information from the evaluation and remedy parts of the HAVOSS model.

Recipients being part of the operational company level reports to rarely get vulnerability information from higher ranks, indicating that upstream communication of vulnerabilities is more common than downstream. Only one company mentioned having a CSO with the responsibility to advice and support the triage responsible if necessary. These results imply that management and board ought to be considered as prominent enough to be an appropriate recipient group. They are thus further referred to as Management and Board and their prototype view as the Management and Board view.

Triage Responsible

The primary responsibility of a triage responsible is in general to evaluate and assess vulnerabilities, as well as to give recommendations on how they should be handled. Some triage responsible only evaluate on the basis of intuition or experience, some use internal scoring systems and some use generic scoring systems such as CVSS. Together with risk, information such as what it is a vulnerability affects, how it affects, what the consequences are if it is not fixed and likelihood of these consequences happening are information that is of relevance to a triage responsible. It is common for vulnerabilities to be prioritised on the basis of risk. The triage responsible is primarily in need of information derived from the step of evaluation in the HAVOSS model. However, to actually assess and evaluate the vulnerability, information from the categories of product knowledge and remedy in the HAVOSS model are also necessary for the triage responsible.

Not all companies have a dedicated triage team, at some companies the development teams, operations teams, system architects, SPOC or security experts are the ones that are separately responsible for the handling of vulnerabilities. By companies with dedicated triage teams, professional roles such as security analysts, CTO, developers, product owners and

marketing and sales representatives are mentioned as triage team members. Some companies have both local and global triage teams so decisions regarding specific products also are made on a basis of the local knowledge of the products, not only the global perspective. The triage role is to be considered as a vital recipient group no matter if it is carried by a team or an individual. This recipient group is further on referred to as the Triage responsible and its prototype view as the Triage view.

Product Owner

At some companies the product owners are a part of the triage team, contributing with the business perspective and technical information about the products during evaluation of vulnerabilities and deciding how they should be prioritised and handled. Other times they are not involved in the triage process at all and similar to other manager roles only informed about severe vulnerabilities, important operational disruptions, if a certain problem needs to be addressed, temporary solutions and other similar operational information. The product owners' knowledge in development and security is mentioned to affect the security of the product, with some prioritising quick releases of new features and others prioritising keeping core features secure by updating them regularly. Product owners sometimes gets information about required patches and are responsible for making sure these are applied. As a recipient group, product owners are therefore considered an intermediate between triage, development, management and board and not prominent enough to qualify as an own recipient group. Instead their prototype view can be a hybrid version of the previously mentioned recipient groups' prototype views. Designing a prototype view for product owners is therefore considered as redundant.

Development Team

Development teams are sometimes the only ones responsible for handling vulnerabilities, both when it comes to evaluation and remediation. According to some companies, developers do not need to know so much about the risk but more the technical aspects such as what the problem is, where in the codebase it is found and remedy recommendations. In such cases the development teams usually only implement the solutions the triage team decides to go with. Yet at some companies the opposite is a fact, with developers or at least representatives from the development teams being involved with assessing risks associated to vulnerabilities. The information identified as the most relevant for the development team is the step of product knowledge in the HAVOSS model, together with remedy of the vulnerability. The development team can also receive deployment information, more specifically patch advisory.

In some cases, the development teams are sometimes solely responsible for monitoring OSS webpages and other information sources in order to discover newly discovered vulnerabilities. Some companies assign incident managers within each development team, who are responsible for synchronising information with the rest of the company, log decisions and implemented solutions, evaluate the outcomes, root causes and how to avoid similar incidents in the future. Developers and development teams are mentioned in almost every interview, making them vital as a recipient group and are therefore necessary to create a specific prototype view for. This recipient group is further on referred to as Development team and the view type as the Development view.

Operations and IT

According to some recipients, operations and IT are inclined to focus on vulnerabilities regarding IT security, and not so much when it comes to OSS in development. At other companies, operations are heavily involved in the continuous monitoring of vulnerabilities, responsible of keeping track of OSS websites, developer communities and flagging to the development or triage teams if new vulnerabilities are discovered and further actions are required. From a DevSecOps point of view, the operations and IT departments should be more involved in the handling of vulnerabilities, but in general these departments are not mentioned in interviews as vital parties regarding this area. The results in this study are not lifting operations and IT as prominent enough recipient groups, for these to currently be eligible for their own prototype views. Instead theirs can be a hybrid of the Triage view and Development view.

Customer

Supplier relationships matter when it comes to how customers relate to vulnerability handling. Some rely to a great extent on the suppliers to take the responsibility for vulnerabilities, meaning they do not have a systematic approach towards scanning or scrutinising third party components. Some customers pay for certain services through which the suppliers are responsible to search for vulnerabilities. Companies often rely on other users in the OSS community to report issues and thus do not conduct any scans on their own. Some suppliers encourage their customers to update without following up whether this is done or not, while others are keen to make sure their customers apply updates and therefore do not publish any vulnerability information publicly at all, but instead focus on emphasising the importance for customers to apply patches.

Customers with stricter security requirements for their products have high demands for their suppliers to keep secure environments. Sometimes weekly regular reporting of the suppliers' vulnerability handling of the products is contractually agreed upon, so the customers can be assured that the suppliers are doing vulnerability scans and penetration tests. Some suppliers provide lists of vulnerabilities that their customers can access anytime. If the customers do not have any strict security requirements, they normally only want to know when an issue will be fixed and what is required of them to do in order to fix it. According to the HAVOSS communication practices, CI4 and CI5, communication with customers is indeed of importance when handling vulnerabilities. External and internal communication of vulnerability and security information has been stated to contribute to security awareness and more secure products being developed. A customer recipient group is thus to be considered as important and is further on referred to as Customer and its prototype view as the Customer view.

Summary of Final Recipient Groups

From the analysis above it can be concluded that the recipient groups in Table 5.1 are prominent enough and thus appropriate to design prototype views for.

Table 5.1: Summary of final recipient groups.

Recipient group	Information
Communication function	Less technically detailed information, but enough details so it is possible to redirect information to the appropriate recipients.
Support function	Less technically detailed information, should be suitable for communication with customers.
Management and Board	Less technically detailed information, more focus on relating vulnerabilities to a business context.
Triage responsible	Technically detailed information, product knowledge and severity scoring to be used as basis for reevaluation.
Development team	Technically detailed information, more focus on remedies than actual triage.
Customer	Not too technically detailed information, straight forward instructions on how to address the issue.

5.3.1 The Vulnerability Information Onion Model

After the above analysis, vulnerability information with its transfunctional character is identified as a denominator for the recipient groups. In TKOM, Figure 2.2, tacit knowledge is presented as multilayer phenomenon. By regarding vulnerability information as a multilayer phenomenon as well, a version of TKOM adapted for segmenting vulnerability information recipients is derived. By acknowledging the fact that amongst the different professional roles there are different application areas for vulnerability information, it is used as a differentiator.

A Vulnerability Information Recipient Onion Model (VIROM) is constructed as follows: The more technically detailed information related to vulnerabilities the recipients need in order to carry out their work duties, the further into to the core of the onion the recipients belong. Moving towards the outer layers of the onion, the degree of necessary software security knowledge for interpreting and making use of the information declines. Such segmentation of recipients based on their knowledge in software security and need of technical information creates a way of sorting them in a information hierarchical way.

As a suggestion VIROM consists of three main layers for a company handling vulnerabilities: The Technical layer, the Organisation layer and the Client layer. To the Technical layer belongs recipients of vulnerability information within the company that are heavily involved with the technical aspects of development and vulnerabilities, i.e. triage responsible and development team. The triage responsible needs to have a greater and broader technical understanding of the vulnerability than the development team, to be able to evaluate the vulnerability. This skill also consist of a tacit component, where intuition and experience are important factors and therefore triage responsible is placed in the layer before development team. To the Organisation layer belongs recipients of vulnerability information within the company that have responsibilities within business, communication and customer relations, i.e. management and board, communication function and support function. Finally, external recipients of vulnerability information, i.e. customer, belongs to the Client layer. The final recipient groups presented in Table 5.1 are sorted according VIROM in Table 5.2. A graphic

presentation of VIROM with the recipient groups is depicted in Figure 5.1.

Table 5.2: Final recipient groups sorted according to VIROM.

Recipient group	Information
1. Triage responsible	Technically detailed information, product knowledge and severity scoring to be used as basis for reevaluation.
2. Development team	Technically detailed information, more focus on remedies than actual triage.
3. Communication function	Less technically detailed information, but enough details so it is possible to redirect information to the appropriate recipients.
4. Support function	Less technically detailed information, should be suitable for communication with customers.
5. Management and Board	Less technically detailed information, more focus on relating vulnerabilities to a business context.
6. Customer	Not too technically detailed information, straight forward instructions on how to address the issue.

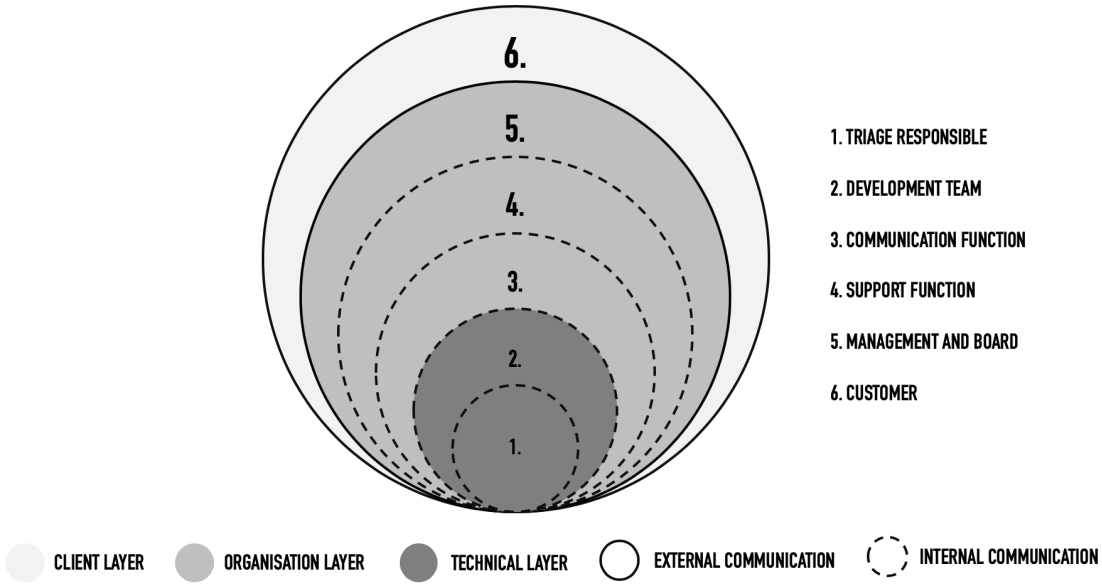


Figure 5.1: Illustration of final recipient groups sorted according to VIROM.

5.4 Prototype View Design

5.4.1 Prototype View Requirements

In order for the prototype views to be of high quality, the subject of SR is of essence. As stated in the theory, SR are essential in order to ensure creation of high quality products while developing new software as they describe the capabilities the suggested software product should have to fit its users' needs. Since the scope of this thesis is centered in the design of and not actual code implementation of the prototype views, the most relevant types of SR are such related to design constraints. In this case, examples of interesting design constraints are standards or policies that must be followed and other security requirements.

The quality perspective leads back to the concept of software quality, which according to the SQuaRE model, Figure 2.12, consists of a total of eight dimensions. One of these is security, i.e. that users only should have access to information that is appropriate to their types and authorisation levels. According to one of the security design principles, no communication or activities should be possible to execute unless there is a specific need for that transaction or access. Another security design principle states that due to the exploit risk, it is especially important for authority regarding features for creating and handling sensitive information to not be reserved for a single person or team.

The above poses an interesting contradiction to one of the main principles of DevSecOps that promotes open collaboration, Table 2.6. It also contradicts previously mentioned research that states the total cybersecurity of a product depends on cooperation between several actors and that sharing of sensitive information increases the performance of actors in a network. The degrees of openness regarding sensitive information such as vulnerability information and sharing of such, vary between the different companies. Some companies have strict policies that mandate no external publication of vulnerabilities at all and restrict the access to vulnerability information to so called "circle of trust", consisting of chosen employees. Others have no problems with the idea of publishing vulnerabilities openly on their public websites, as long as the issue is already fixed or there is a remedy to publish as well. This indicates the hardly surprising conclusion that the extent and types of requirements between different companies most likely varies. It is thus clear that varying degrees of vulnerability information openness is to be taken into consideration when designing prototype views, but especially during the construction of a final product that might interpret them.

The concluded variation of SR can also be linked to the concluded difference between professional roles involved with vulnerability handling at different companies, since such differences also contributes to the variation of SR for each company. It is therefore to be considered difficult to successfully design one specific solution that aims to fit all company types, without stating that some flexibility in terms of features is necessary in order for the eventual usage of the prototype views in practice to be successful. In practice such flexibility is expressed by making it available to the companies that are supposed to be the end users of the view, to contribute with their own requirements on who should have access to what and be able to request hybrids of different view types. An example of when a hybrid comes to questions, is for a company at which the triage responsible for a specific product category is the development team of that product. In such a case it is necessary for the development team to have access to both types of views, preferably from the same sidebar.

Another reason to why frame flexibility for the prototype views is advocated, is that the-

ory suggests that the security organisation of a company often adapts to the business growth and maturity of the company. This is likely to be true for other company functions as well. The enabling of e.g. hybrid constellations of the prototype views are a solution to enhance their adaptability excessively. If standardisation of such feature tailoring is possible, it can contribute to a higher quality of the end product that these prototype views might become a part of. Such flexibility is to consider as going hand in hand with two other software quality dimensions, namely usability and maintainability. These dimensions address how much a product can be used by specified users and achieve goals in a successful way, and how much a product can be modified due to changes in requirements or its environment. One requirement related to these dimensions, that is specifically expressed by some of the interview respondents, is that an eventual end product should be possible to integrate with other software tools. This to streamline the working processes surrounding the usage of an extensive arsenal of tools, which often is the case for some companies.

5.4.2 Description of Prototype View Content

The HAVOSS model for handling vulnerabilities, Figure 2.7, is kept in mind during the design of the prototype views, to ensure that the vital parts of vulnerability handling are addressed in a wholesome way. All prototype views show an entry named "Overview" in their left sidebars, yet no such view page is included in the prototype views. This is because the prototype views all are based on templates from previously designed views by Debricked and repeating the overview design does not add any extra value. Instead the focus is to design prototype views with content that is of somewhat new character.

Triage View

Since the triage responsible recipient group is heavily involved with the technology aspects of handling vulnerabilities, i.e. making evaluations and assessments of vulnerabilities based on factors such as CVSS details and technical product knowledge, the Triage view concept mirrors this in its layout. This results in five view pages being designed in order to fulfil the information need of this recipient group: Vulnerabilities, Vulnerabilities > ID, Product portfolio, Product portfolio > Prod ID and Information sources.

- **Vulnerabilities** In order to give the triage responsible easy access to an overview of all vulnerabilities, a view page dedicated to this subject is of value. The page contains such information that summarises the present vulnerability handling situation for the company in question. To begin with, the identifiers for the different vulnerabilities are shown so they are easy to search for, find and possible to tell apart. Regarding these identifiers, they are either CVE IDs if the vulnerabilities in question are published, or company specific IDs if the vulnerabilities are detected internally and not published due to company disclosure policies or likewise. If the vulnerability has a specific name, e.g. as CVE-2014-0160 is generally referred to as the Heartbleed bug, this can perhaps also be used as an identifier. According to both theory and the interview results, risk is often used as prioritisation basis both via internal scoring systems as well as generic ones such as CVSS. In the vulnerability qualification workflow, Figure 2.8, the process of sifting amongst and prioritising vulnerabilities begins with filtering out moderates

and lows. Then the qualification workflow of vulnerabilities continues with determining if the vulnerabilities actually are affecting the company's environment and re-rating according to eventual company specific scales. It is thus of value to show the severity of the vulnerabilities on this page, both before and after reevaluation. The status of the vulnerabilities is presented as well and tells whether the vulnerabilities are yet to be examined, affecting the company environment, vulnerable or fixed, so an overlook of what is left to be done is gained from a first sight. Filtering of the vulnerabilities depending on their status is also possible, since such a categorisation provides additional clarity of the vulnerability handling situation to the user. When the vulnerabilities are detected, who within or outside the company that detected them and what customers and products that are affected by the vulnerabilities is also presented on this page, as it gives the context of each vulnerability and are factors that might be of interest during prioritisation and evaluation.

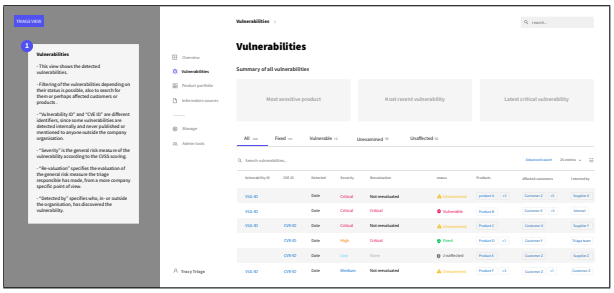


Figure 5.2: Vulnerabilities view page.

- **Vulnerabilities > ID** When choosing one of the listed vulnerabilities on the Vulnerabilities view page, the user is directed to more detailed information about the chosen vulnerability. The information simply consists of the extended versions of what is given in the previous page. Actions that are needed to carry out in order to evaluate and assess the vulnerabilities are possible to take here, e.g. a space dedicated to reevaluation is where the reevaluation of the vulnerability according to company specific criterias is carried out and logged. Another space is dedicated for logging the decided fix, also called remedy. It is possible to set the status of the vulnerability to the appropriate value. A suggested solution in order facilitate communication of vulnerability information in a structured way is communication requests. By creating a request, vulnerability information to relevant recipients of both internal and external character is requested by the triage responsible to be sent out to chosen recipients. The request contains such vulnerability information that is relevant for the various recipients, e.g. remedy recommendations for the responsible development team, direct information about vulnerabilities to customers, eventual information updates for board and management, etc. The technical writers, PR or other similar communication functions then adapt the information to each recipient type. This communication solution is elaborated on under the Communication function view.

- **Product portfolio** This view summarises vulnerabilities and other security related events from a product or perhaps even component perspective, as this is a way to give an

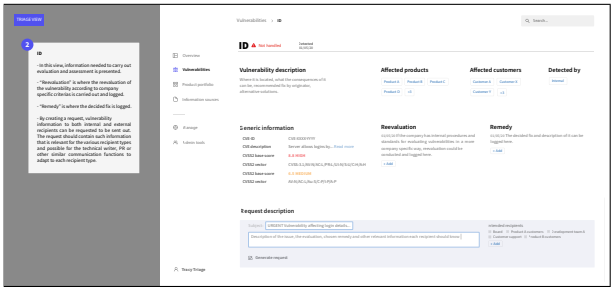


Figure 5.3: Vulnerabilities > ID view page.

overview on such a level. The products are referred to by their specific product IDs or name descriptions. The status of products shows if any new events regarding vulnerabilities have been detected. Associated products are presented here, to show which other products or product groups the product in question are related to and thus also might be affected by the same vulnerabilities. The status of the products, i.e. if new events have been checked or not, is a possible basis for filtering amongst the products.

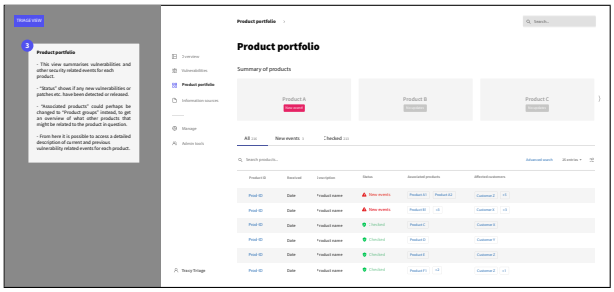


Figure 5.4: Product portfolio view page.

- **Product portfolio > Prod ID** By choosing a product on the previous view page, it is possible to access a detailed description of current and previous vulnerability related events for the product in question. This view thus elaborates on what product group the product belongs to, which customers that are connected to the product and a log with all current and previous vulnerability related events. Events are in this case the detection of new vulnerabilities, recommended or released remedies and likewise.

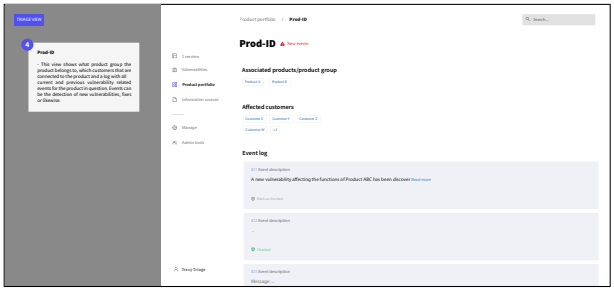


Figure 5.5: Product portfolio > Prod ID view page.

- **Information sources** This view page shows categorisation of vulnerability information sources, from which vulnerability information is collected by the user. In Table 4.4 it is shown that information about vulnerabilities comes from different types of sources, ranging from internal sources such as development teams to external sources such as vulnerability subscription lists and developer communities. Therefore, it is of interest to the user to access and filter vulnerability information on the basis of commonly used sources.

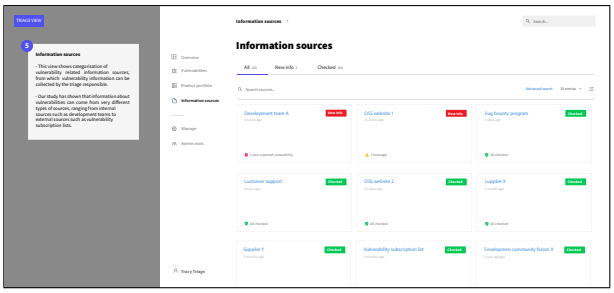


Figure 5.6: Information sources view page.

Development View

The layout and content of the Development view pages are similar to those of the Triage view. The biggest difference between these views is less focus on evaluation and more on creation of the recommended remedy. Another suggested content difference from the corresponding Triage view is that development teams are not able to make communication requests, as this is restricted to the triage responsible in order for the development teams to solely focus on carrying out the decided measures. This conforms with the theory of the SDLC models scrum and XP. Both models emphasise that development teams should not waste valuable time on other tasks than writing and developing code. As much else, this kind of division of duties does of course depends on the requirements and work policies of the company in question. For some companies, some type of two-way communication between development teams and triage responsible carried out from this view is likely of interest. The view has four pages, Vulnerabilities, Vulnerabilities > ID, Product portfolio and Product portfolio > Prod ID.

- **Vulnerabilities** This view shows the detected vulnerabilities and is more or less identical with the one in the Triage view. The biggest difference between the Triage view and the Development view is the information shown in the view page called ID.

- **Vulnerabilities > ID** In this view, focus lies on presenting information that is needed for the developers to create the remedy decided by the triage responsible. Product information can be collected about each affected product in order to ensure that the implementation of remedies is carried out in correct ways for all products. The implementation of the remedies is logged and when a vulnerability is fixed, the status of the vulnerability is changed from here.

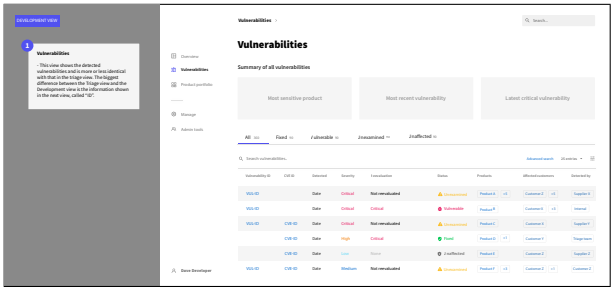


Figure 5.7: Vulnerabilities view page.

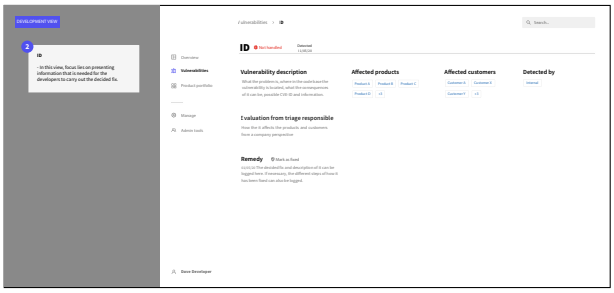


Figure 5.8: Vulnerabilities > ID view page.

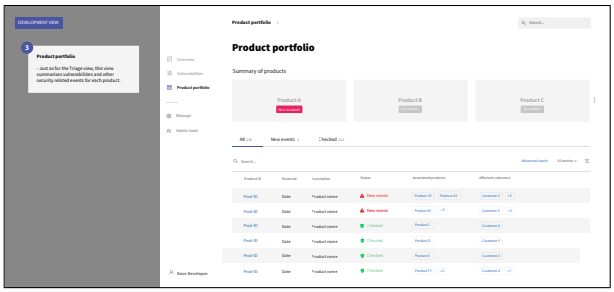


Figure 5.9: Product portfolio view page.

- **Product portfolio** Just as for the Triage view, this view summarises vulnerabilities and other security related events on product level, giving the user an idea of the software context.

- **Product portfolio > Prod ID** Just as for the Triage view, this view shows what product group the product belongs to, which customers that are connected to the product and a log with all current and previous vulnerability related events for the product in question.

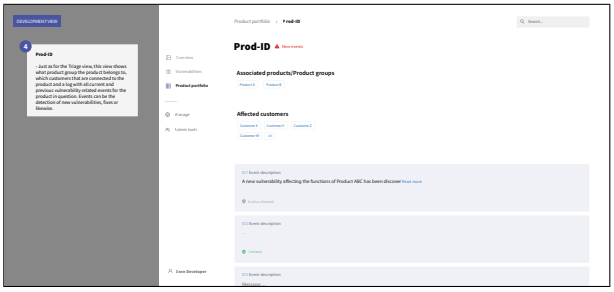


Figure 5.10: Product portfolio > Prod ID view page.

Communication Function View

The purpose of this view is to address the need of filtering and forwarding vulnerability information, so it is appropriately communicated both internally and externally. This by adding the function of sending communication requests within the organisation. Depending on organisation structure, e.g. who it is that evaluates and fixes vulnerabilities, the originator of the request is likely a development team or the triage responsible, as is shown in this view. By requesting the communication, describing what information should be sent out and who the recipients should be, the information is adjusted and formulated by the receiver of the request and passed on. Receivers of the request are likely PR or technical writers. Requests are sent from company internal stakeholders, but the communication that is requested might also be intended for both internal and external stakeholders, e.g. emails for customers or publications on the website. The view consists of four view pages, Requests, Requests > REQ-ID, Recipients and Recipients > Board.

- **Requests** In this view page an overview is given of incoming requests from fellow employees at the company. Requests are identified via specific IDs. Besides this, the overview gives information such as the date of request receipt, who it is that is ordering the request, the subject of the request i.e. what it is that has happened or need to be informed about, the status of whether editing has begun or not and if it is still active or not. Finally, the intended recipients are given. Such information makes it easy for the request handler to estimate the workload or urgency of the request.

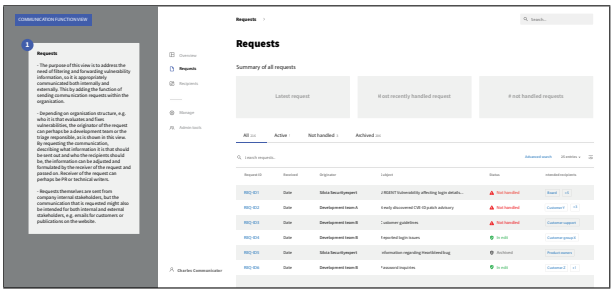


Figure 5.11: Requests view page.

- **Requests > REQ-ID** By choosing one of the requests, more detailed information of the chosen request is shown to the user. The information is simply the extended versions

of what is given on the previous view page. Actions that are needed to carry out in order adapt the information so it fits the recipient of it are possible to take here. This page contains a description of the vulnerability or whatever the issue is and its consequences. What information each recipient should get and technical details that the more technically inclined recipients, e.g. developers, need to know in order to carry out the fix or likewise. Information to the various recipients of the different information adaptions is sent from here, communicated directly via the publication feature meanwhile presenting an overview of what is communicated to each recipient.

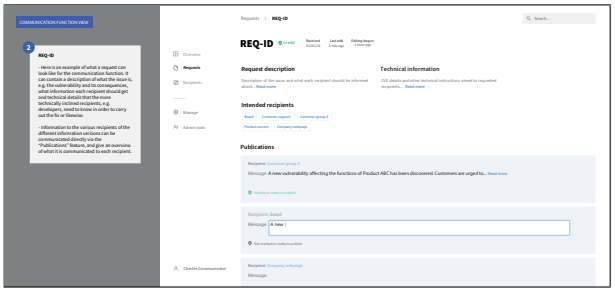


Figure 5.12: Requests > REQ-ID view page.

- **Recipients** The information on this view page is kind of similar to the one in Requests, but the summary of the requests is based on the recipient type. It shows every request for each recipient type and is filtered depending on who it is that will receive the information. The author of the publication shows who it is that has handled the request and written the information.

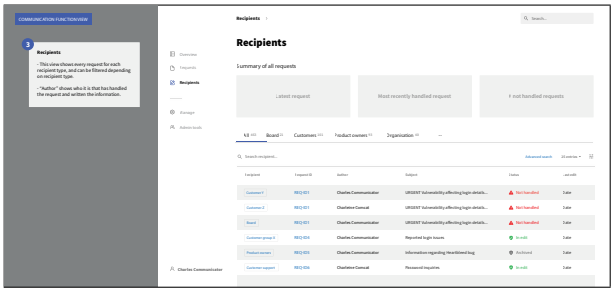


Figure 5.13: Recipients view page.

- **Recipients > Board** By choosing one of the recipient types in the previous view, a summary of all publications aimed for the specific recipient is shown. In this example, the chosen recipient is board. The content characteristics describe the kind of information this specific recipient type should receive, e.g. on which technical level the information should be presented, if there is a specific type of format in which it should be presented in, etc. A log showing each publication gives the historical context of previously published information for the specific recipient.

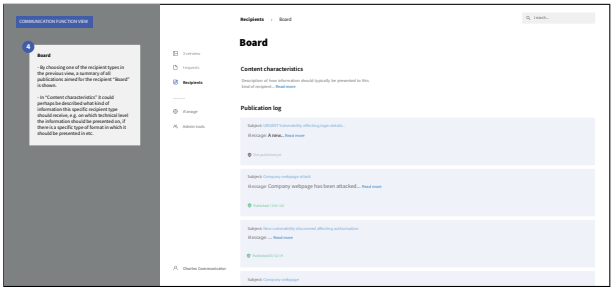


Figure 5.14: Recipients > Board view page.

Support Function View

This view shows information that support functions such as customer support or key account managers should know in order to assist customers that contact them regarding vulnerability related events. The view pages designed are called Events, Events > EVT ID and Submit report.

- **Events** This view shows information that support functions such as customer support or key account managers should know in order to assist customers that contact them regarding vulnerability related events. The information does thereby conform a lot with that of the Customer view. Event identifier, date of receipt, subject of the event and which components or products that are affected are examples of information that are described in this page.

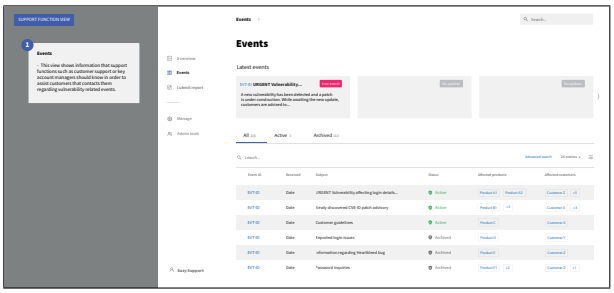


Figure 5.15: Events view page.

- **Events > EVT-ID** Description of each event and details on how it should be handled is presented here, specifying information the support functions pass on when in contact with customers. The given information is not too detailed as the final recipients, the customers, are usually not inclined to wanting to know more than what is necessary for them. Information about expected solution releases or instructions for handling the issue meanwhile awaiting a remedy is given here.

- **Submit report** This view is a suggestion on how customer support or other support functions can handle incoming vulnerability tips from customers, by submitting reports to perhaps triage responsible. This feature can be integrated with existing tools if similar features already exists within in the company.

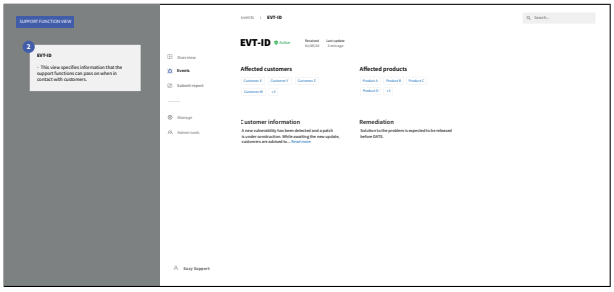


Figure 5.16: Events > EVT-ID view page.

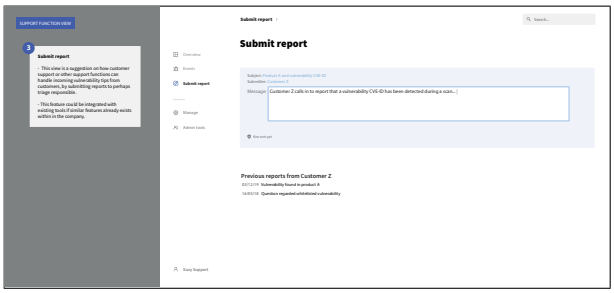


Figure 5.17: Submit report view page.

Management and Board View

The purpose of this view is to give management and board an overview of vulnerabilities on a product or product group level, since the interview results show that this recipient group often do not need to know so much technical details and only want to be made aware of severe incidents. As mentioned before, this varies of course depending on the organisation structure and division of responsibilities between different companies. If the management or board of a specific company is interested in being informed of more technical details, such view pages are adapted for these recipient groups from perhaps the Triage and Development views. Thus, two view pages are designed for management and board, Product portfolio and Product portfolio > PRGR-ID.

- **Product portfolio** This view page contains such information that is regarded as summarising on a product group level. The product group ID is specified, together with which date the information about the latest event is received, description of the occurring events and which products and customers that are affected.

- **Product portfolio > PRGR-ID** By choosing a certain product group, this view shows more general information than technical for the user. For some companies it is of interest to have a more business inclined perspective presented here as well, e.g. description of how a certain vulnerability can damage the brand or likewise. To enhance the vulnerability interest of users that normally might discard vulnerabilities' due to indifference or lack of knowledge, illustrating the amount of vulnerabilities over time in trend lines or the severity of detected vulnerabilities via speedometer are solutions aimed at enhancing the understanding of vulnerabilities and their consequences. According to the interview results, Table 4.5, such graphical presentation is already used

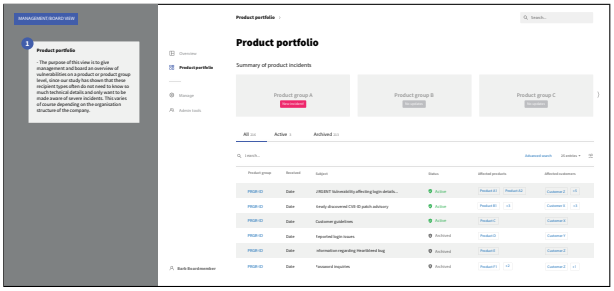


Figure 5.18: Product portfolio view page.

in practice as a tool in order to communicate vulnerability information. A log showing all the information related to vulnerabilities and other security issues is of interest to give the vulnerability related events a historical context.

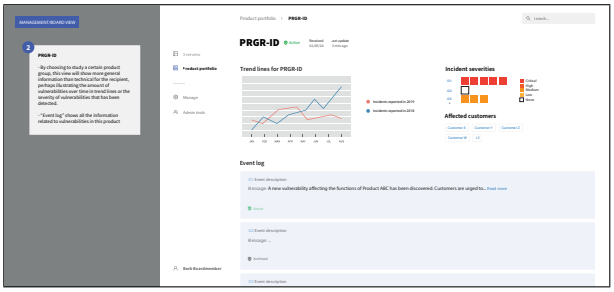


Figure 5.19: Product portfolio > PRGR-ID view page.

Customer View

This view is the only view intended for a company external recipient, a customer. The interview results show that customers in general want as straight forward information as possible, only be given brief information about what has occurred and what they need to do to fix it. Notice that customers often also are suppliers to other companies, meaning that they might have internal views such as the other described prototype views. One idea is that the type of information shown in this view can perhaps appear when the customer enters their equivalent to the page Information sources from the Triage view and choosing to enter one of their suppliers information flows. According to the analysis, it is likely that the first to receive this kind of information at the customer company are the recipient groups Triage responsible and Development team. Some customers might want to receive the vulnerability information at an independent feature, which is how the view pages Events, Events > EVT-ID, Component portfolio and Component portfolio > COMP-ID are currently more designed for.

- **Events** In this view page an overview of the communicated events surrounding vulnerabilities is presented. Events are the announcement of a newly discovered vulnerability, a new update or patch advisory, weekly vulnerability reports from suppliers and reminders to update. The purpose of this view is for the customer to be able to access relevant vulnerability information surrounding components or products from all of its

suppliers, gathered and manageable at one place. Event identifier, date of receipt, subject of the event, which components or products that are affected and from what supplier the information is coming from are examples of information that are described in this page. A view page like this where suppliers have the opportunity push important information, updates and update reminders to their customers contributes to more secure products for the users of the view. Such handling of vulnerability related updates is also seen in the interview results.

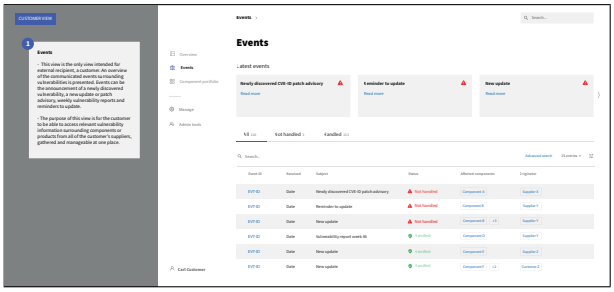


Figure 5.20: Events view page.

- **Events > EVT-ID** Description of each event and details in how it should be handled are presented on this view page, to make it easy for the customer to follow the instructions or advice from the suppliers. The given information is at glance not too complicated or technically advanced, but can be more specified if this would be a requirement of the customer.

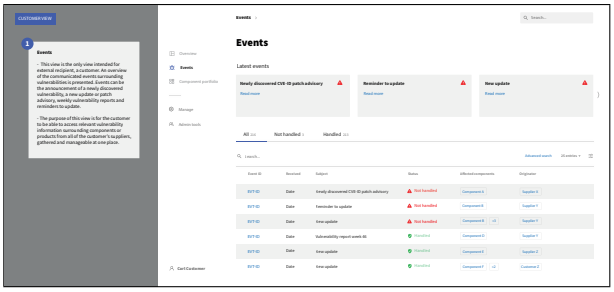


Figure 5.21: Events > EVT-ID view page.

- **Component portfolio** This view page structures the vulnerabilities as what component or product they affect, in order to be able to study them from that perspective. Affected stakeholders are specified, stating which of the customer company's stakeholders that are affected by the event. Stakeholders are likely of both internal and external character, since it depends on the area of use of the affected components. In this example all of the stakeholders are regarded as customers of the customer.

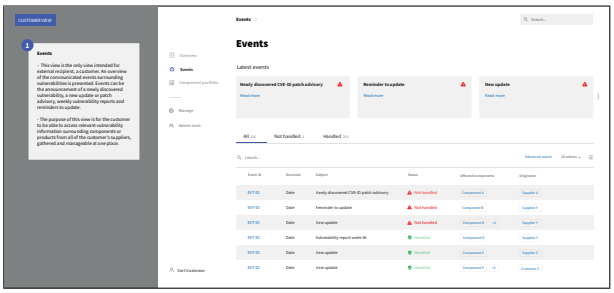


Figure 5.22: Component portfolio view page.

- **Component portfolio > COMP-ID** By selecting one of the components in the previous view, a summary of all current and previous events regarding the component is presented, together with associated products and affected stakeholders.

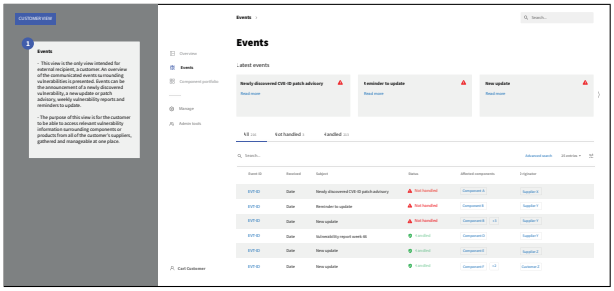


Figure 5.23: Component portfolio > COMP-ID view page.

Chapter 6
Prototype Views

The Prototype Views contains the resulting prototype views that are based on the information obtained after the analysis. The chapter also contains feedback from the interview respondents.

6.1 Design Results

In Table 6.1 the location of the design result in Appendices is given for each prototype view.

Table 6.1: Location direction for the resulting prototype views in Appendices.

View type	Location
1. Triage view	p. 93
2. Development view	p. 99
3. Communication function view	p. 105
4. Support function view	p. 111
5. Management and Board view	p. 115
6. Customer view	p. 119

6.2 Prototype Views Feedback

Four of the twelve interview respondents contributed with feedback on the prototype views. The overall verdict is that the prototype views contains enough information and are easy to process for the user. The recipient groups are considered as appropriate and relevant. Examples of positive feedback that is given is that the view page showing latest events for Support function and Customer is valuable, as well as Triage responsible and Development team are able to get the status of vulnerabilities through a quick glance at their views. Severity ranking

and filtering are also mentioned as valuable input. Mentioned improvement suggestions that should be taken into consideration for further adjustments of the prototype views are:

- Add information such as statistics and trends to the Development view.
- Developers should be able to add vulnerabilities.
- A column called "Assigned to" should be added to see who it is within the company that owns each vulnerability.
- An "Audit log" should be added to see who is handling each CVE, what and when measures have been taken in the system.
- Severity scoring should be added to the overall view page in Customer view, even if it is only the suppliers' recommendations and does not match how serious the vulnerability is internally.
- Depending on who the customer is, the Customer view can be redundant since some customers are not even interested in the information that is presented in this prototype view.
- Depending on developer and development project the column "Affected customers" can be redundant, since some projects only involve one customer.

Chapter 7
Discussion

The Discussion chapter contains the discussion of the conducted work and results. The contribution of the thesis is discussed, thereafter the collected data and analysis are critically evaluated and discussed in terms of credibility.

7.1 Contribution

This study contributes to an expansion of the existing academic knowledge base regarding tailored communication of vulnerabilities. It provides insights that can be useful for a wide range of companies using OSS in their products and solutions. Numerous studies explore vulnerability management, software development models and even communication between customers and suppliers. However, no other study appears to have the goal of grouping vulnerability information recipients and present the information that they require on an online-sharing platform.

One of the main contributions of this study is the initial exploration of tailoring vulnerability information to different recipients - providing insight into what typical recipients there are within software development and how these recipients can be grouped and what vulnerability information that each recipient group need. The study is conducted within a topic that is very much on the agenda in organisations worldwide and thus the results anchored both in existing theory and new collected data can generate value and insight for organisations across industries. The views for the different recipients are clear results from the study and are easily adapted and used directly without much modification in many organisations, since the views are generic and applicable to many contexts and industry types.

A specific theoretical contribution to research is the creation of the Vulnerability Information Recipient Onion Model, VIROM. The model addresses software development and vulnerability management in practice and the findings of factors such as technical knowledge, organisation and contact with clients being more relevant to group recipients than professional roles or titles. Lastly the findings from the study contribute with input to both

Debricked's SecT project and the HATCH project.

7.2 Credibility

As described in the Methodology chapter of the report, there are certain factors on which credibility in terms of reliability, validity and representativeness depends on. Below are these aspects discussed and thus the total credibility of the study evaluated.

Regarding the theory and the conducted literature view, search motors such as LUBSearch are used together with certain books on methodology to conduct a credible study. A thorough research is done in order to find the most relevant and up to date sources that are well renowned and commonly cited, to ensure that the desired initial exploration of the topic is established in a successful way.

The choice of conducting several case studies to gather qualitative data can be evaluated as a suitable approach to the purpose of the study and a good foundation for the exploratory approach chosen. In terms of representativeness, the interviewees are chosen randomly and also through the snowball method, with the interviewees recommending someone that they find suitable to participate in the study. The vulnerability knowledge of the participants varies and not all participants have full insight into what the process of vulnerability management looks like at their company and what the scope of responsibility looked like at other roles or functions. Furthermore some interviewees tend to answer in more general how vulnerabilities are treated at their company rather than specifically how vulnerabilities in third party components are treated. In total twelve interviews are conducted with representatives from a wide range of industries, most of them having different professional roles. The final result is thus considered as generic. However, one can argue that the sample size is rather small and would benefit from being complemented by a larger survey to ensure that the results are generic and that the results are not threatened by being biased due to the sample size. As it is always great to have more data, the conducted case studies are considered to be sufficient to make the desired analysis.

To ensure that the validity of the study is kept at a high level, feedback from the interviewees is collected to make sure that the information is reflecting the situation and not misleading. In addition to this continuous third party reviewing is conducted with the help from the supervisors of this thesis. This to also ensure that the study is not exposed to extensive observer bias and thus enable a reliable study. The analysis is based on the collected data and anchored in existing theory. Fact checks are performed to make sure no subjective conclusions or intuition based facts are used in the study and that the conclusions are based on facts discovered during the study.

To summarise, the authors find that sufficient actions are taken in order to achieve a study with high credibility and the results being applicable in the context of the study.

Chapter 8
Conclusions

The Conclusions chapter contains the answers to the research questions and is where the final thesis conclusions are drawn, by reconnecting the conducted work with its stated purpose. Future research recommendations are also given.

8.1 Fulfilling the Purpose and Answering the Research Questions

Since the research questions in this thesis are possible to answer with the conducted work as a basis, the intended purpose of the thesis is considered as fulfilled. The participating companies represents various industries, contributing to the basis of analysis being of sufficient general character. Distinctive recipient groups of vulnerability information are identified as well as what vulnerability information each group need, without the results being limited to only be applicable for a specific industry. Different platform views are designed for each recipient group. Therefore, the intended purpose of the thesis is achieved.

RQ1 Who are the typical recipients of vulnerability information within a company?

In this thesis it is discovered that the diversity of professional roles and their responsibility scopes between different companies is vast. When trying to identify specific recipient types, this diversity makes it complicated to succeed in doing so. In total 39 different professional roles are mentioned as recipients of vulnerability information and in general the division of responsibilities between similar roles varies between different companies. Yet the analysis shows that grouping of some professional roles and functions into different recipient groups is possible. The most prominent recipient groups that are identified are triage responsible, development team, communication function, support function, management and board and

finally customer. All though product owner, operations and IT at first are identified as recipient groups, the interview results indicate that there is either not enough data to make assumptions on what type of information these recipient groups need or that their part in handling vulnerabilities is too similar to those of triage responsible, development team, management and board.

RQ2 What kind of vulnerability information does each type of recipient need within the frame of their profession?

Since vulnerability information is identified as a transfunctional concept and the recipient groups as needing different types of information, the detail level of required vulnerability information is regarded as a differentiator. This leads to the development of the framework VIROM, in order to sort the recipient groups according to their required detail level of vulnerability information. VIROM consists of three main layers, see Figure 5.1. The recipient groups triage responsible and development team belong to the inner Technical layer, because of their deep involvement with the technical aspects of handling vulnerabilities. Communication function, support function and management and board belong to the middle Organisation layer, since their main vulnerability responsibilities are more business, communication and customer inclined. The final recipient group, customer, belongs to the outer Client layer.

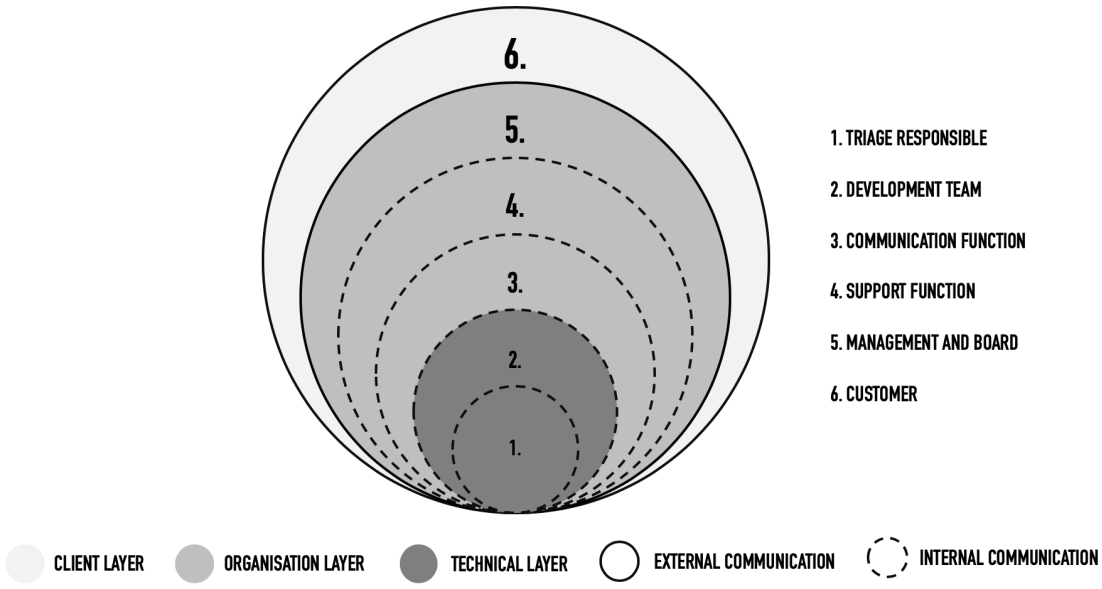


Figure 5.1: Illustration of final recipient groups sorted according to VIROM.

RQ3 How can vulnerability information be tailored and presented on an online-sharing platform for each type of recipient?

Six prototype views are identified as relevant when tailoring and presenting vulnerability information on an online-sharing platform to different recipients. These are the Triage view, the Development view, the Communication function view, the Support function view, the

Management and Board view and finally the Customer view. The evaluation of these prototype views shows that the content of each view is in general suitable for different company and industry types. In Figure 8.1 the problem formulation figure from the Introduction chapter, Figure 1.2, is adjusted so the question marks are replaced with the results of this thesis.

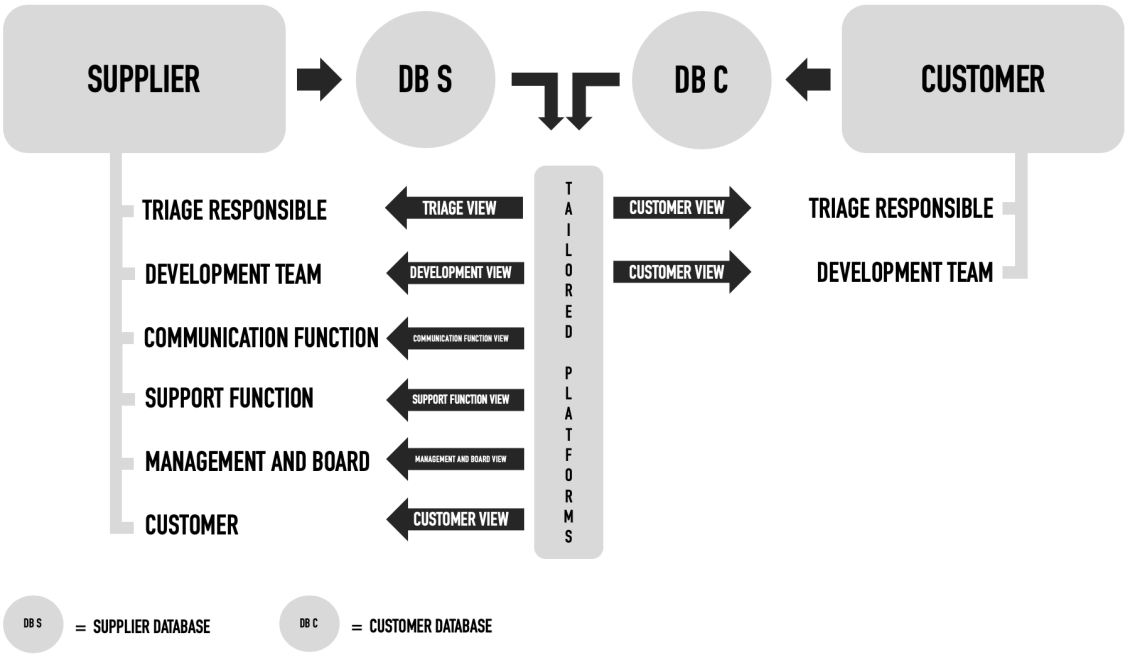


Figure 8.1: Addressing the problem formulation with answers to the research questions.

8.2 Further Research Recommendations

Further research recommendations are to test the prototype views in practice and investigate whether they contribute to more secure software products and if they actually do facilitate vulnerability handling. A possible adjustment of the prototype views is to further differentiate them and adapt them to potential discrepancies in information need between different industry types. The layers of VIROM can perhaps also be the subject of further research, as there might exist additional layers that are yet to be identified.

References

[1] B.B. Agarwal, S.P. Tayal, and M. Gupta. *Software engineering and testing*. Jones and Bartlett Publishers, Sudbury, Massachusetts, 2010.

[2] D. Asher and M. Popper. Tacit knowledge as a multilayer phenomenon: the "onion" model. *The Learning Organization*, Vol. 26:264–275, 2019.

[3] K. Beck and C. Andres. *Extreme programming explained: Embrace change*. Addison-Wesley, Boston, Massachusetts, 2004.

[4] City Business Solutions. Types of Cyber Security Breaches – What are the Most Common. `https://www.cbsit.co.uk/2016/09/23/types-cyber-security-breaches-common/` Gathered 2020-02-05.

[5] CNBC. A serious shortage of cybersecurity experts could cost companies hundreds of millions of dollars. `https://www.cnbc.com/2019/03/06/cybersecurity-expert-shortage-may-cost-companies-hundreds-of-millions.html` Gathered 2020-02-05.

[6] CVE Details. Browse Vulnerabilities By Date. `https://www.cvedetails.com/browse-by-date.php` Gathered 2020-02-04.

[7] Diverse. *Nordisk kriminalkrönika 2001*. Egmont SAGA, Copenhagen, 2001.

[8] P. Foreman. *Vulnerability Management*. Taylor and Francis Group, Boca Raton, Florida, 2010.

[9] L. Fraccascia and D.M. Yazan. The role of online information-sharing platforms on the performance of industrial symbiosis networks. *Resources, Conservation and Recycling*, Vol. 136:473–485, 2018.

[10] Heartbleed. The Heartbleed Bug. `http://heartbleed.com/` Gathered 2020-02-06.

[11] L. E. Hecht and L. Clark. Survey: Open Source Programs Are a Best Practice Among Large Companies. `https://thenewstack.io/`

survey-open-source-programs-are-a-best-practice-among-large-companies/ Gathered 2020-02-06.

[12] T. Hsiang-Chih Hsu. *Hands-on security in DevOps: Ensure continuous security, deployment, and delivery with DevSecOps*. Packt Publishing Ltd, Birmingham, United Kingdom, 2018.

[13] D. W. Hubbard. *The failure of risk management: Why it's broken and how to fix it*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2009.

[14] M. Höst. Personal communication, 1 December 2019.

[15] M. Höst, B. Regnell, and P. Runeson. *Att Genomföra Examensarbete*. Studentlitteratur AB, Lund, Sweden, 2006.

[16] M. Höst, J. Sönnerup, M. Hell, and T. Olsson. Industrial practices in security vulnerability management for IoT systems – an interview study. In *Proceedings of the 2018 International Conference on Software Engineering Research & Practice*, pages 61–67, 2018.

[17] IBM. IBM X-Force Threat Intelligence Index 2020. https://www.ibm.com/security/data-breach/threat-intelligence Gathered 2020-04-14.

[18] IBM Security. IBM Cost of a Data Breach 2019. https://www.ibm.com/security/data-breach Gathered 2020-02-05.

[19] International Organization for Standardization. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE, 2014.

[20] P. Jalote. *A concise introduction to software engineering*. Springer Science+Business Media, Berlin, Germany, 2008.

[21] G. Johnson, K. Scholes, and R. Whittingtion. *Fundamentals of Strategy*. Pearson Education Limited, Harlow, England, 2009.

[22] P.A. Laplante. *Encyclopedia of software engineering*. Taylor and Francis Group, Boca Raton, Florida, 2011.

[23] T. Lee. Value chain for software product delivery. https://www.linkedin.com/pulse/value-chain-software-product-delivery-thomas-lee/ Gathered 2020-04-13.

[24] G. Maayan. Devsecops: Security and DevOps working together, 2020. https://developer.ibm.com/recipes/tutorials/devsecops-security-and-devops-working-together/ Gathered 2020-04-04.

[25] National Vulnerability Database. CVE-2014-0160 Detail. https://nvd.nist.gov/vuln/detail/CVE-2014-0160#vulnCurrentDescriptionTitle Gathered 2020-04-14.

[26] N. Ngah, S. Fahmy, W. Othman, N. Sukinah, Z. Fariha, N. S. Sulaiman, A. Yacob, and N. Shiratuddin. Evaluation of e-book applications using iso 25010. https://www.researchgate.net/publication/286439574_Evaluation_of_e-Book_Applications_Using_ISO_25010 Gathered 2020-04-20.

[27] P. Nikbakht Bideh, M. Höst, and M. Hell. HAVOSS: A Maturity Model for Handling Vulnerabilities in Third Party OSS Components. In *Product-Focused Software Process Improvement*, pages 81–97, 2018.

[28] NIST. Vulnerability. https://csrc.nist.gov/glossary/term/vulnerability Gathered 2020-02-05.

[29] NIST. Vulnerability Metrics. https://nvd.nist.gov/vuln-metrics/cvss Gathered 2020-02-05.

[30] T. Olsson, M. Hell, M. Höst, U. Franke, and M. Borg. Sharing of vulnerability information among companies – a survey of swedish companies. In *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 284–291, 2019.

[31] Opensource. What is open source software? https://opensource.com/resources/what-open-sourceGathered2020-02-06.

[32] M. Robinson. DevSecOps: A complete guide to what, why, and how. https://www.plutora.com/blog/devsecops-guide Gathered 2020-04-11.

[33] M. Saunders, P. Lewis, and A. Thornhill. *Research methods for business students*. Pearson Education Limited, Harlow, England, 2009.

[34] K. Schwaber and M. Beedle. *Agile software development with scrum*. Prentice Hall, Upper Saddle River, 2001.

[35] S. Sharma and B. Coyne. *DevOps for dummies*. John Wiley and Sons, Hooboken, New Jersey, 2004.

[36] M. Stoica, M. Mircea, and B. Ghilic-Micu. Software development: Agile vs. traditional. http://www.revistaie.ase.ro/content/68/06%20-%20Stoica,%20Mircea,%20Ghilic.pdf Gathered 2020-04-02.

[37] Synopsys Cybersecurity Research Center. 2019 open source security and risk analysis. https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/rep-ossra-19.pdf Gathered 2020-04-14.

[38] The Guardian. Two Swedes charged with hacking into NASA computers. https://www.theguardian.com/science/1999/aug/17/spaceexploration.aeronautics Gathered 2020-02-04.

[39] The White House. Fighting Cybersecurity Threats to the Growing Economy. https://www.whitehouse.gov/articles/fighting-cybersecurity-threats-to-the-growing-economy/ Gathered 2020-02-04.

[40] Upphandlingsmyndigheten. Små och medelstora företag. `https://www.upphandlingsmyndigheten.se/leverantor/SME/` Gathered 2020-04-14.

[41] M. Warner. Cybersecurity: A Pre-History. *Intelligence and National Security*, Vol. 27:781–799, 2012.

[42] E. Wheeler. *Security risk management - Building an information security risk management program from the ground up*. Elsevier, Waltham, Massachussetts, 2011.

[43] M. Yilmaz, R. O'Connor, and P. Clarke. A Systematic Approach to the Comparison of Roles in the Software Development Processes. In *Software Process Improvement and Capability Determination: 12th International Conference*, pages 198–209, 2012.

Appendices

Appendix A
Case Interview Guide

Bakgrund
Vad är din yrkesroll?
Hur använder du dig av information om sårbarheter i din yrkesroll?

Sårbarhetshantering
Har ni någon särskild policy för att hantera sårbarheter? Beskriv processen.
Vad är du nöjd med i dagsläget?
Finns det något du tycker kan förbättras, isåfall vad?
Vilka befattningar är ansvariga för säkerhet?
Vilka befattningar har triage-ansvar/bedömer hur riskfylld en sårbarhet är?
Hur bedöms risken, har ni något eget score system eller används redan befintliga?
Vilka åtgärdalternativ finns det efter att en sårbarhet har blivit utvärderad?
Vilka befattningar avgör vad nästa åtgärd bör vara?
Vilka befattningar utför dessa åtgärder?
Vilka befattningar avgör vilka metrics som ska användas vid sårbarhetsövervakning?
Vilka metrics använder ni er av idag?
Vilka befattningar tar del av informationen som fås via övervakning?

Mottagare av sårbarhetsinformation
Vilka befattningar är inblandade i sårbarhetshantering och vilken information behöver man på den strategiska nivån?
Vilka befattningar är inblandade i sårbarhetshantering och vilken information behöver man på den taktiska nivån?
Vilka befattningar är inblandade i sårbarhetshantering och vilken information behöver man på den operativa nivån?
Vilka befattningar brukar du komma i kontakt med då det gäller sårbarheter?
Vilka befattningar skulle man kunna gruppera och därefter skräddarsy information åt?

Hur ska man identifiera dessa befattningar, vilka parametrar är viktiga?

Sårbarhetsinformation

Om du skulle få information om en ny sårbarhet, vad är kritiskt för dig att veta?
Vad skulle du säga är kritiskt för andra befattningar att veta? Rangordna.
Räcker det med endast viss information för att du ska kunna utföra ditt jobb, vilken?
Från vilka befattningar får du information om eventuella sårbarheter?
Rapporterar du om sårbarheter till någon?

Presentation av information

Ska informationen anpassas beroende på om man är leverantör eller kund?
Hur föredrar du att få information presenterad för dig?

Appendix B
Main Interview Guide

Bakgrund
Vad är din yrkesroll?
Hur använder du dig av information om sårbarheter i din yrkesroll?

Sårbarhetshantering
Har ni någon särskild policy för att hantera sårbarheter? Beskriv processen.
Vad är du nöjd med i dagsläget?
Finns det något du tycker kan förbättras, isåfall vad?
Vilka befattningar är ansvariga för säkerhet?
Vilka befattningar har triage-ansvar/bedömer hur riskfylld en sårbarhet är?
Hur bedöms risken, har ni något eget score system eller används redan befintliga?
Vilka åtgärdalternativ finns det efter att en sårbarhet har blivit utvärderad?
Vilka befattningar avgör vad nästa åtgärd bör vara?
Vilka befattningar utför dessa åtgärder?
Vilka befattningar avgör vilka metrics som ska användas vid sårbarhetsövervakning?
Vilka metrics använder ni er av idag?
Vilka befattningar tar del av informationen som fås via övervakning?

Mottagare av sårbarhetsinformation
Vilka befattningar brukar du komma i kontakt med då det gäller sårbarheter?
Vilka befattningar skulle man kunna gruppera och därefter skräddarsy information åt?

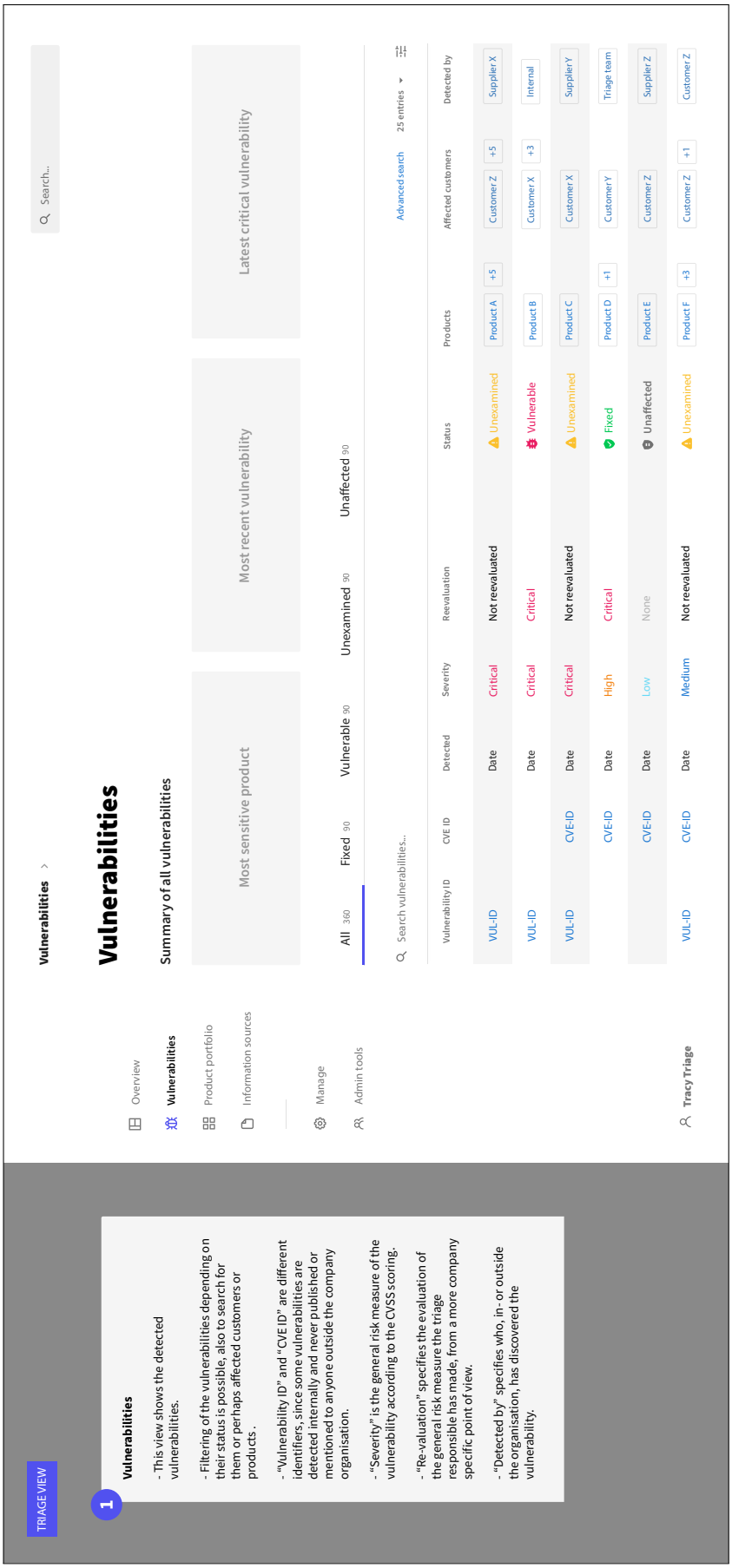
Sårbarhetsinformation
Om du skulle få information om en ny sårbarhet, vad är kritiskt för dig att veta?
Vad skulle du säga är kritiskt för andra befattningar att veta?
Räcker det med endast viss information för att du ska kunna utföra ditt jobb, vilken?
Från vilka befattningar får du information om eventuella sårbarheter?

Rapporterar du om sårbarheter till någon?

Presentation av information

Hur presenteras data inom företaget idag?
Hur presenteras information om sårbarheter inom företaget idag?
Hur föredrar du att få information presenterad för dig?

Appendix C
Triage View


TRIAGE VIEW
1
Vulnerabilities
- This view shows the detected vulnerabilities.
- Filtering of the vulnerabilities depending on their status is possible, also to search for them or perhaps affected customers or products .
- "Vulnerability ID" and "CVE ID" are different identifiers, since some vulnerabilities are detected internally and never published or mentioned to anyone outside the company organisation.
- "Severity" is the general risk measure of the vulnerability according to the CVSS scoring.
- "Re-valuation" specifies the evaluation of the general risk measure the triage responsible has made, from a more company specific point of view.
- "Detected by" specifies who, in- or outside the organisation, has discovered the vulnerability.
Vulnerabilities
Vulnerabilities
Summary of all vulnerabilities
Most sensitive product
Most recent vulnerability
Latest critical vulnerability
Search...
Overview
Vulnerabilities
Product portfolio
Information sources
Manage
Admin tools
Tracy Triage
All 360
Fixed 90
Vulnerable 90
Unexamined 90
Unaffected 90
Search vulnerabilities...
Advanced search
25 entries
Vulnerability ID
CVE ID
Detected
Severity
Reevaluation
Status
Products
Affected customers
Detected by
VUL-ID
Date
Critical
Not reevaluated
Unexamined
Product A +5
Customer Z +5
Supplier X
VUL-ID
Date
Critical
Critical
Vulnerable
Product B
Customer X +3
Internal
VUL-ID
CVE-ID
Date
Critical
Not reevaluated
Unexamined
Product C
Customer X
Supplier Y
CVE-ID
Date
High
Critical
Fixed
Product D +1
Customer Y
Triage team
CVE-ID
Date
Low
None
Unaffected
Product E
Customer Z
Supplier Z
VUL-ID
CVE-ID
Date
Medium
Not reevaluated
Unexamined
Product F +3
Customer Z +1
Customer Z

TRIAGE VIEW

2

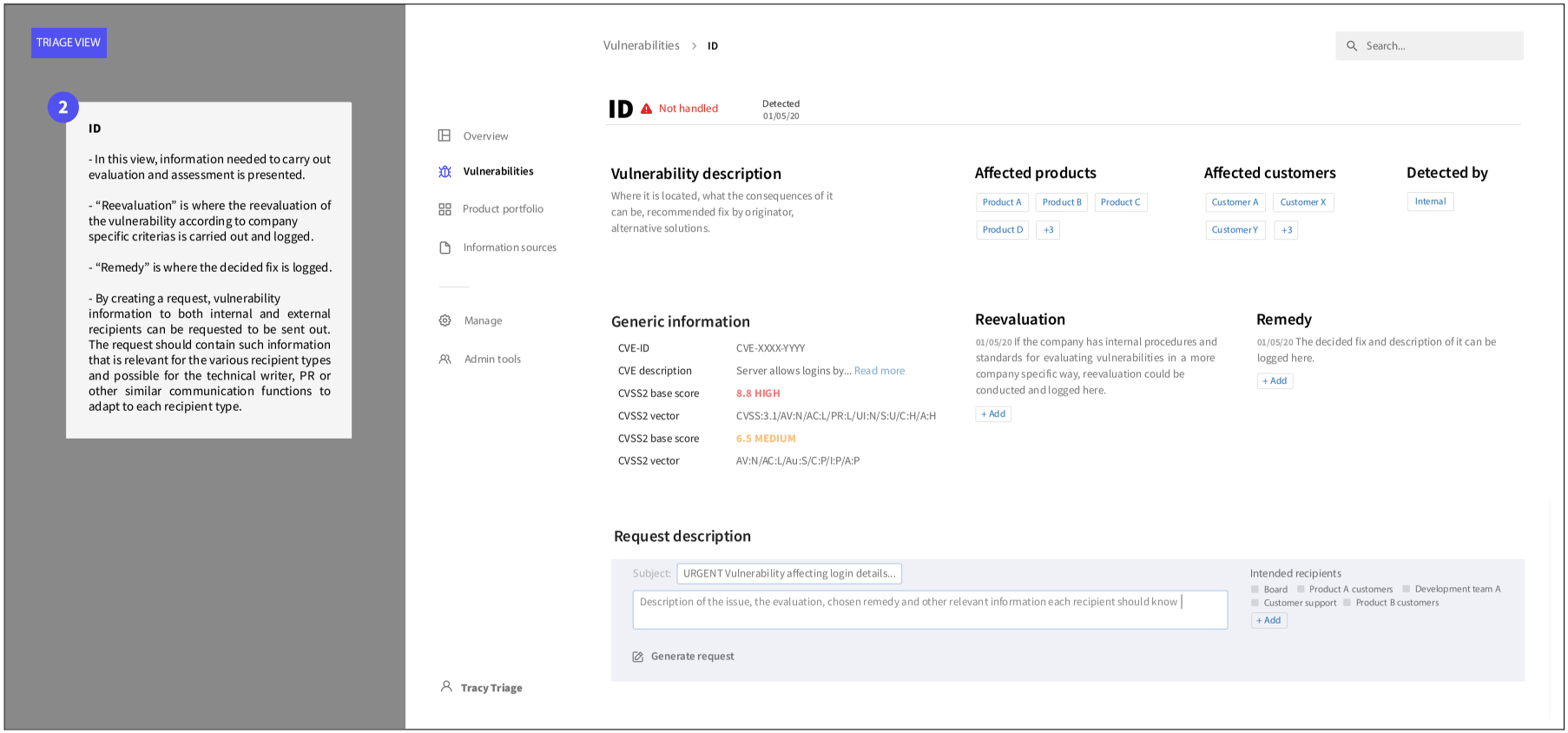
ID

- In this view, information needed to carry out evaluation and assessment is presented.

- “Reevaluation” is where the reevaluation of the vulnerability according to company specific criterias is carried out and logged.

- “Remedy” is where the decided fix is logged.

- By creating a request, vulnerability information to both internal and external recipients can be requested to be sent out. The request should contain such information that is relevant for the various recipient types and possible for the technical writer, PR or other similar communication functions to adapt to each recipient type.



TRIAGE VIEW

3

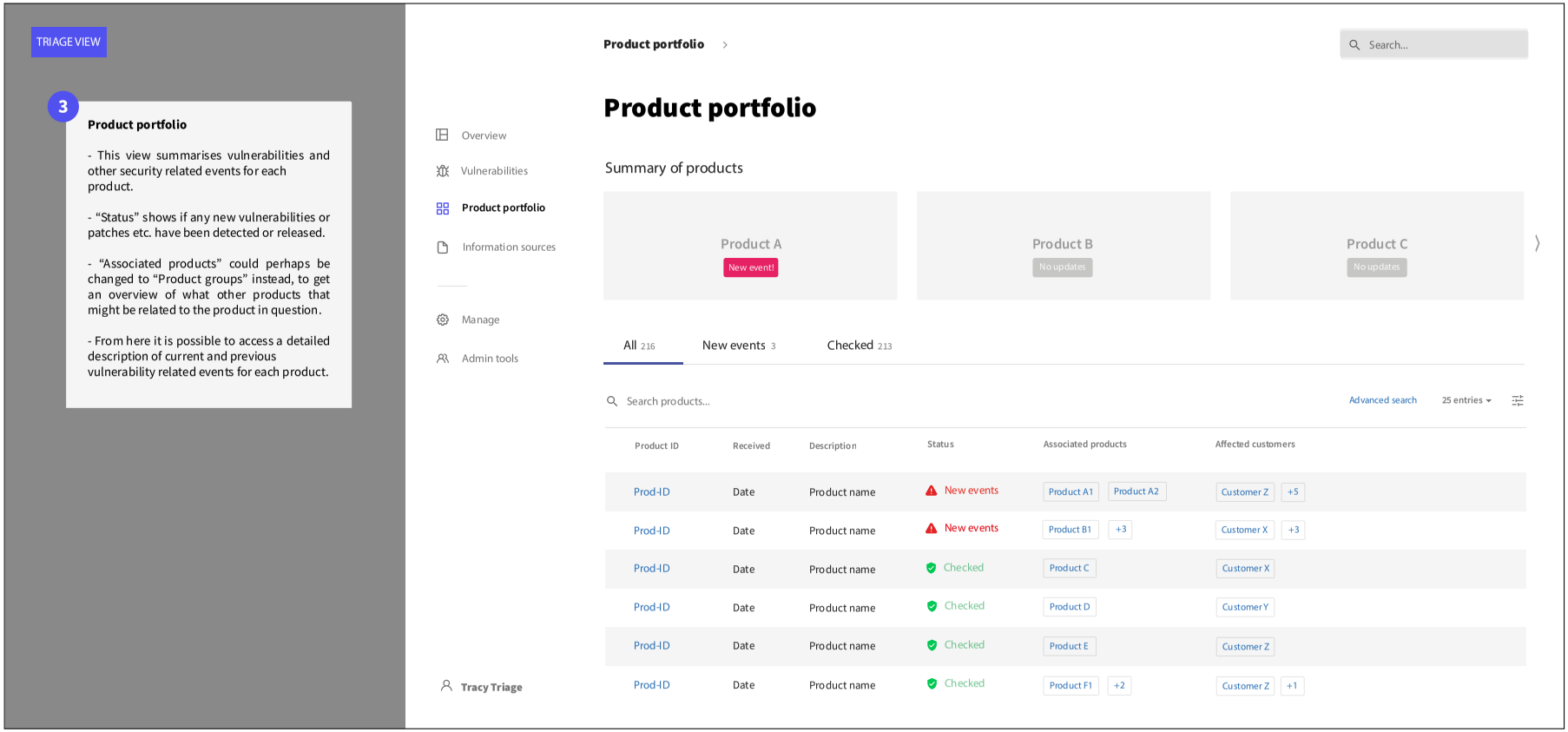
Product portfolio

- This view summarises vulnerabilities and other security related events for each product.

- “Status” shows if any new vulnerabilities or patches etc. have been detected or released.

- “Associated products” could perhaps be changed to “Product groups” instead, to get an overview of what other products that might be related to the product in question.

- From here it is possible to access a detailed description of current and previous vulnerability related events for each product.

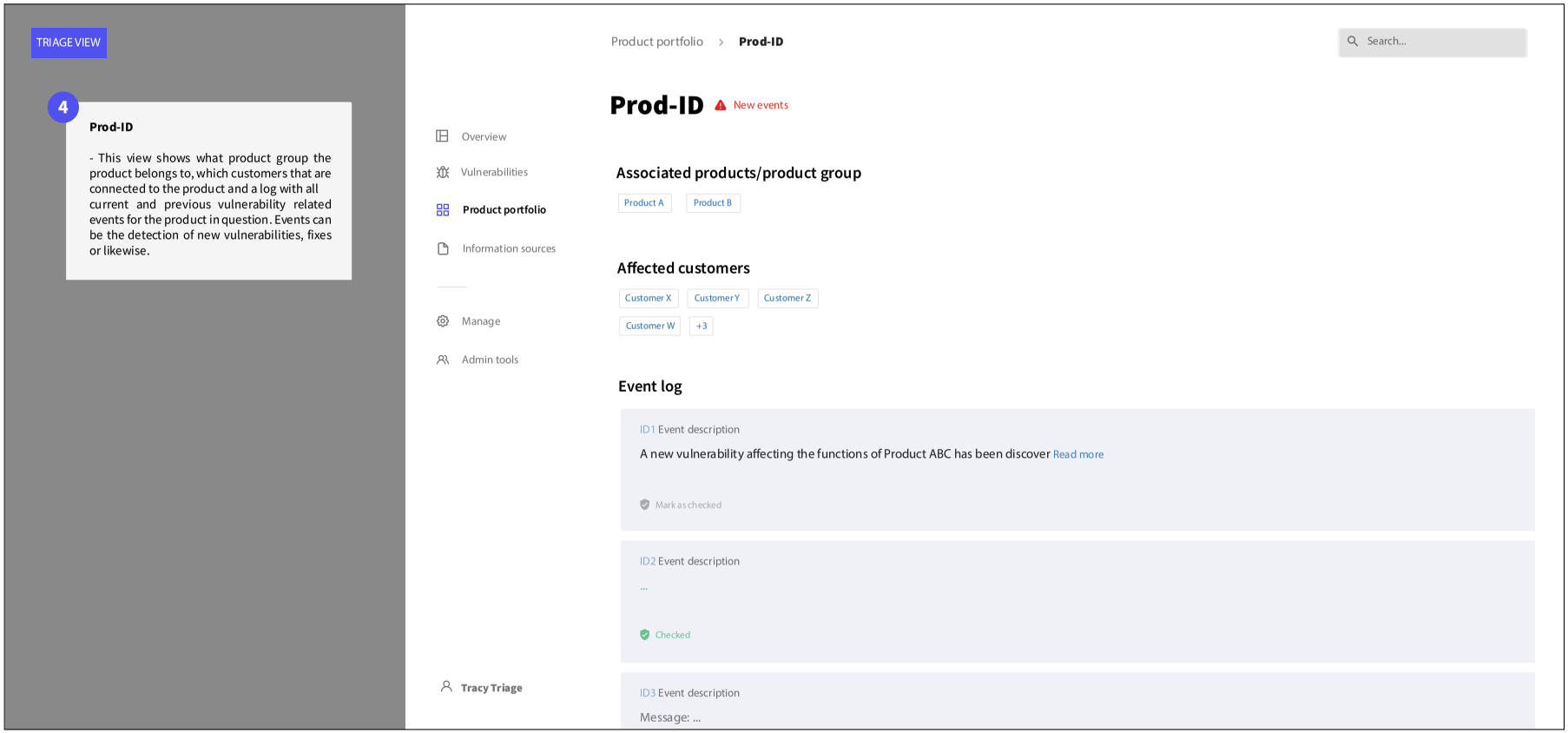


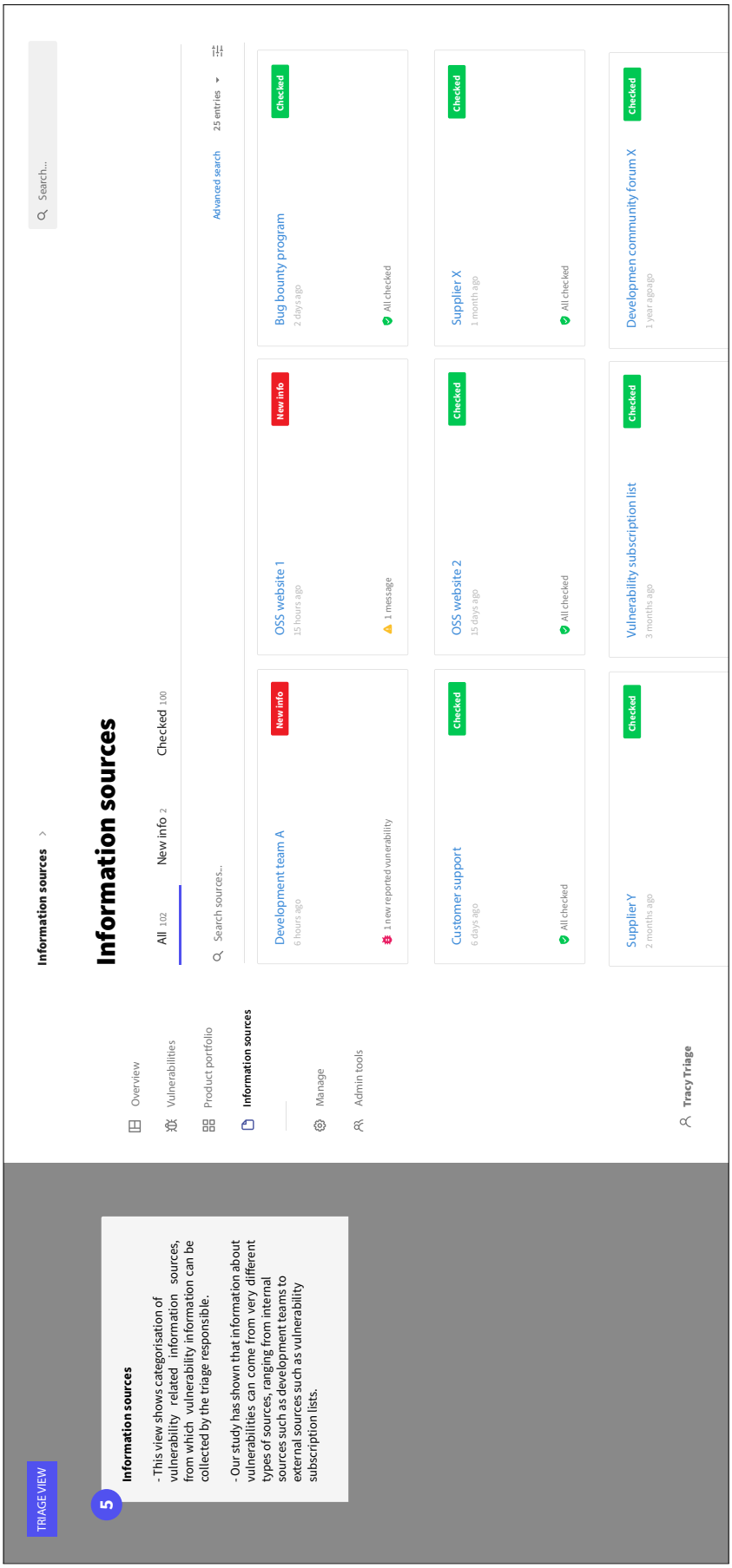
TRIAGE VIEW

4

Prod-ID

- This view shows what product group the product belongs to, which customers that are connected to the product and a log with all current and previous vulnerability related events for the product in question. Events can be the detection of new vulnerabilities, fixes or likewise.




TRIAGE VIEW
5
Information sources
- This view shows categorisation of vulnerability related information sources, from which vulnerability information can be collected by the triage responsible.
- Our study has shown that information about vulnerabilities can come from very different types of sources, ranging from internal sources such as development teams to external sources such as vulnerability subscription lists.
Information sources
Search...
Overview
Vulnerabilities
Product portfolio
Information sources
Manage
Admin tools
Tracy Triage
Information sources
Information sources
All 102
New info 2
Checked 100
Search sources...
Advanced search
25 entries
Development team A
6 hours ago
New info
1 new reported vulnerability
OSS website 1
15 hours ago
New info
1 message
Bug bounty program
2 days ago
Checked
All checked
Customer support
6 days ago
Checked
All checked
OSS website 2
15 days ago
Checked
All checked
Supplier X
1 month ago
Checked
All checked
Supplier Y
2 months ago
Checked
Vulnerability subscription list
3 months ago
Checked
Developmen community forum X
1 year agoago
Checked

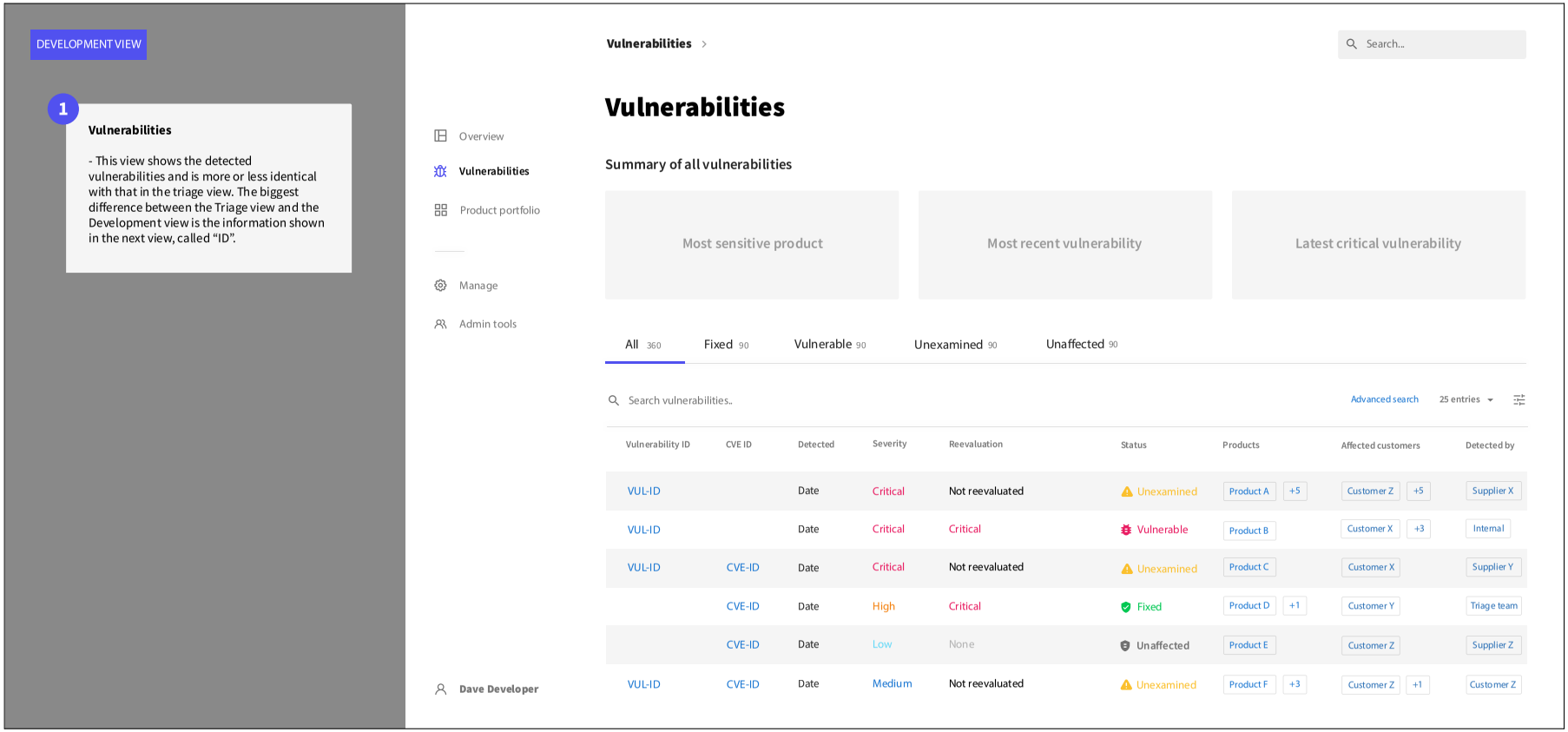
Appendix D
Development View

DEVELOPMENT VIEW

1

Vulnerabilities

- This view shows the detected vulnerabilities and is more or less identical with that in the triage view. The biggest difference between the Triage view and the Development view is the information shown in the next view, called “ID”.

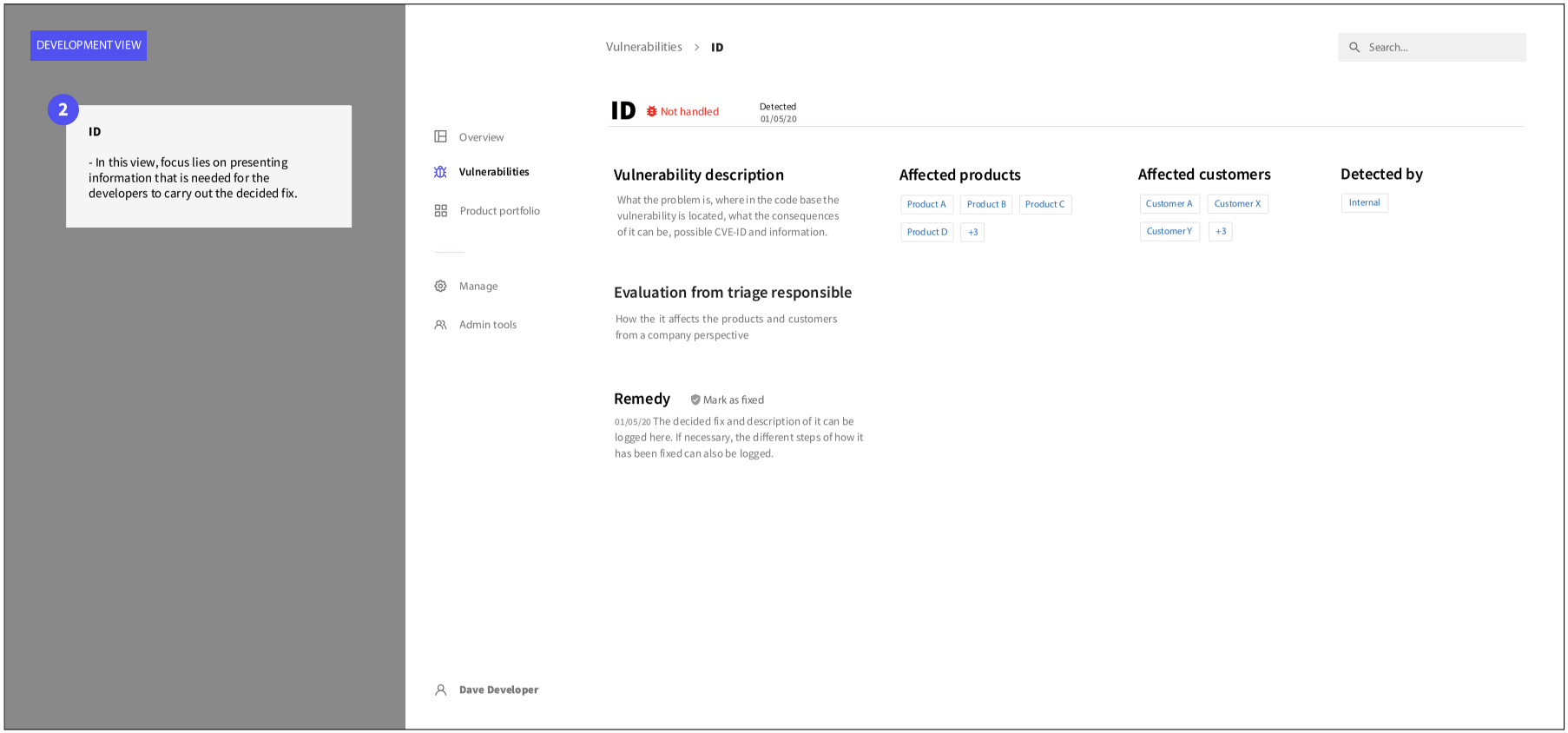


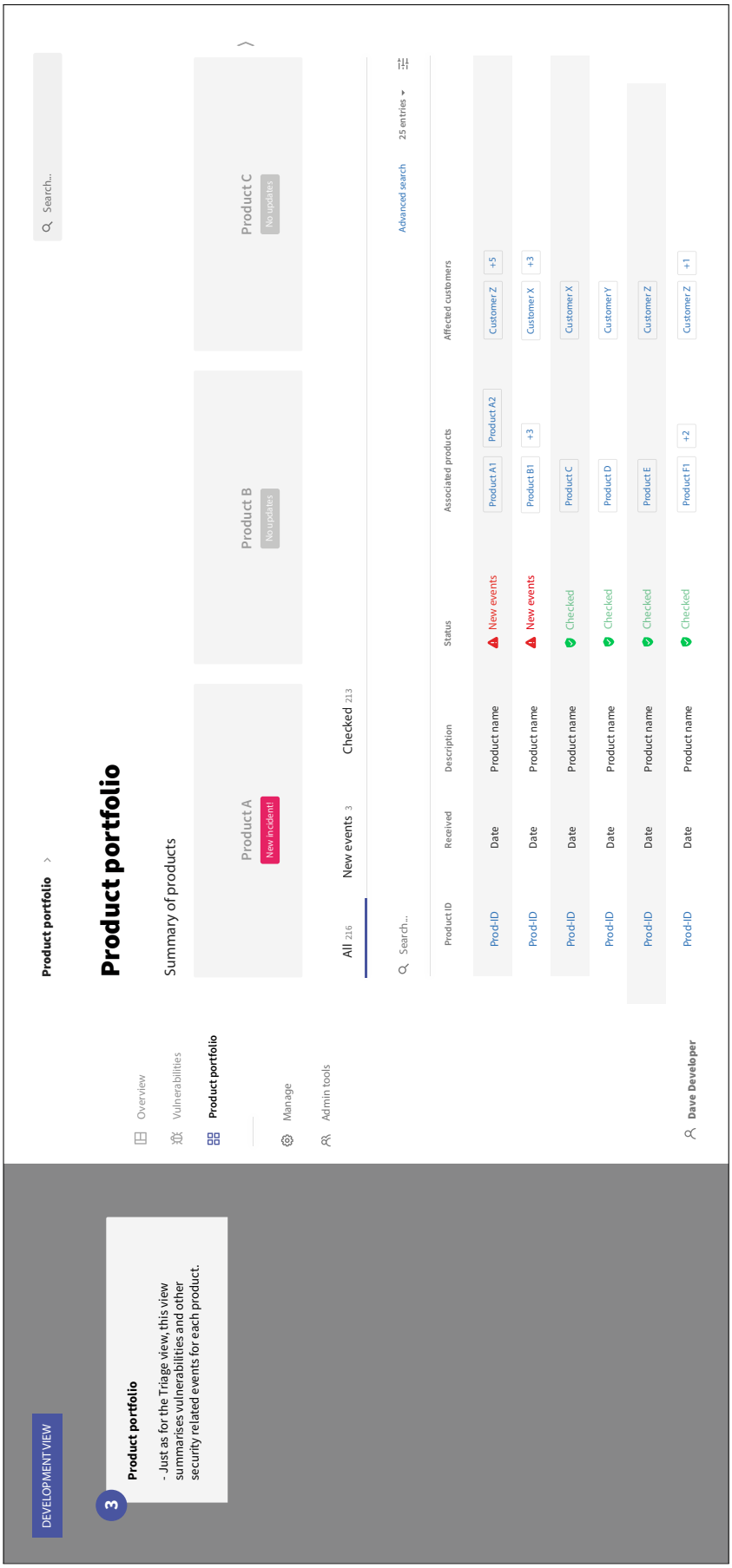
DEVELOPMENT VIEW

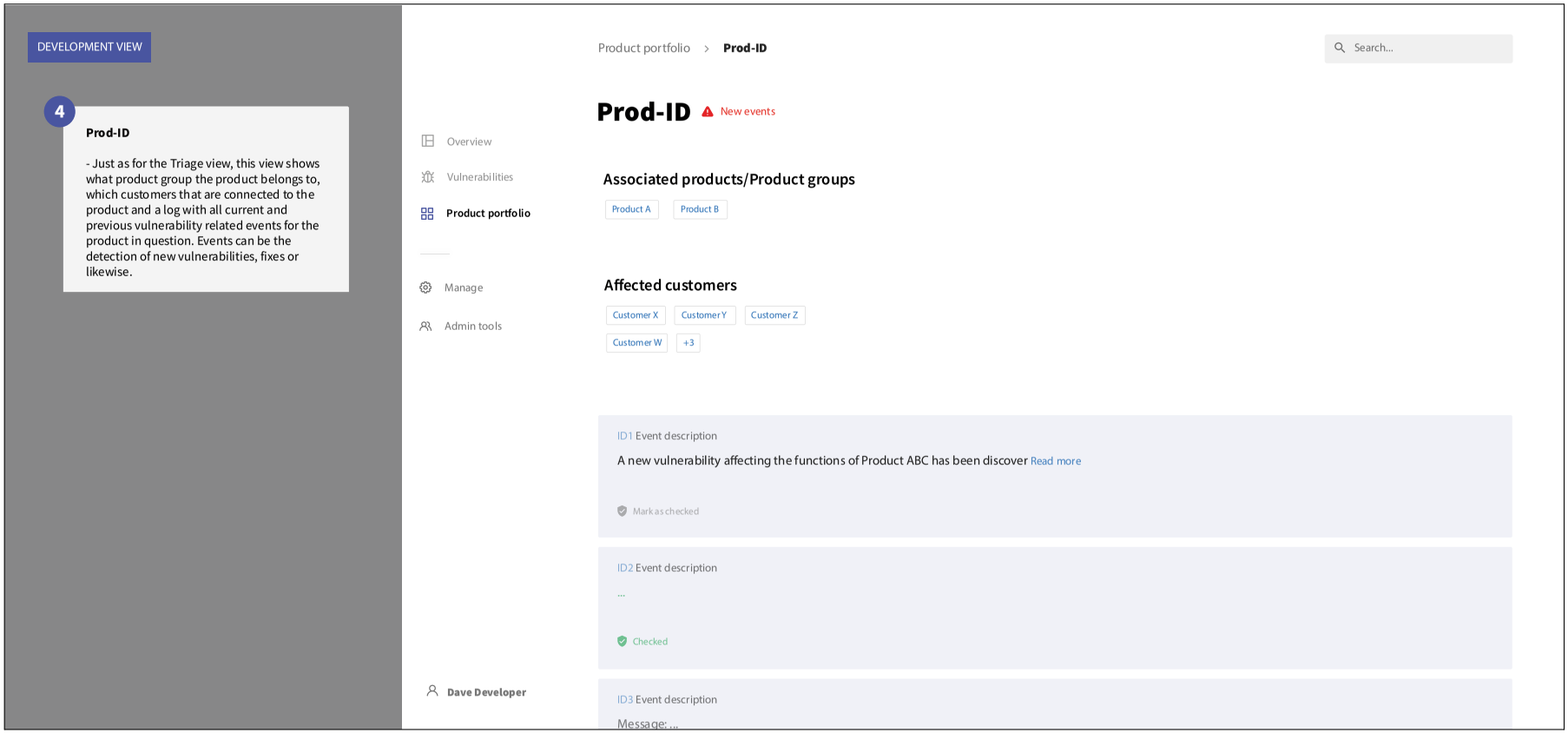
2

ID

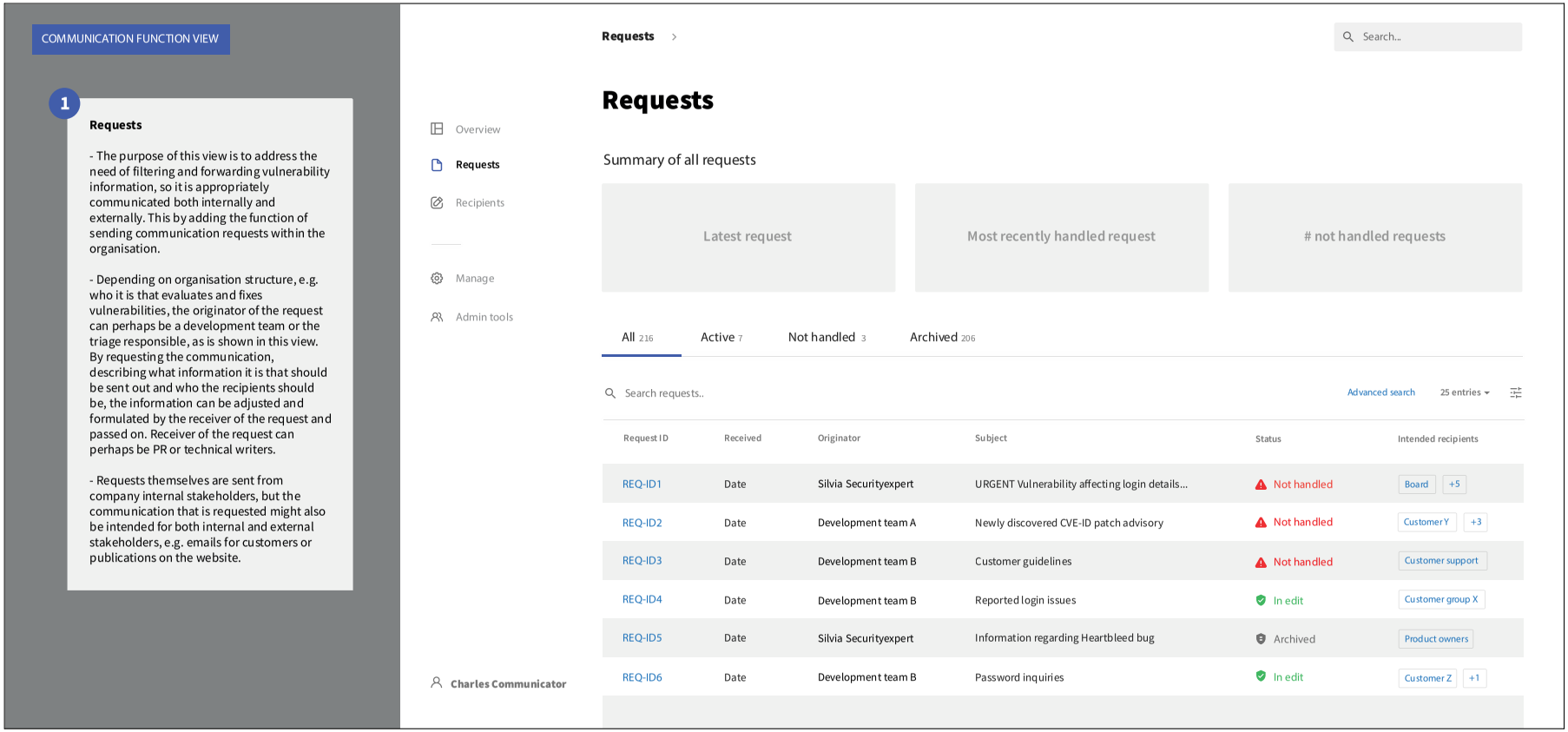
- In this view, focus lies on presenting information that is needed for the developers to carry out the decided fix.




DEVELOPMENT VIEW
3
Product portfolio
- Just as for the Triage view, this view summarises vulnerabilities and other security related events for each product.
Product portfolio
Search...
Overview
Vulnerabilities
Product portfolio
Manage
Admin tools
Dave Developer
Product portfolio
Product portfolio
Summary of products
Product A
New incident!
Product B
No updates
Product C
No updates
All 216
New events 3
Checked 213
Search...
Advanced search
25 entries
Product ID
Received
Description
Status
Associated products
Affected customers
Prod-ID
Date
Product name
New events
Product A1
Product A2
Customer Z
+5
Prod-ID
Date
Product name
New events
Product B1
+3
Customer X
+3
Prod-ID
Date
Product name
Checked
Product C
Customer X
Prod-ID
Date
Product name
Checked
Product D
Customer Y
Prod-ID
Date
Product name
Checked
Product E
Customer Z
Prod-ID
Date
Product name
Checked
Product F1
+2
Customer Z
+1


DEVELOPMENT VIEW
4
Prod-ID
- Just as for the Triage view, this view shows what product group the product belongs to, which customers that are connected to the product and a log with all current and previous vulnerability related events for the product in question. Events can be the detection of new vulnerabilities, fixes or likewise.
Overview
Vulnerabilities
Product portfolio
Manage
Admin tools
Dave Developer
Product portfolio > Prod-ID
Search...
Prod-ID
New events
Associated products/Product groups
Product A
Product B
Affected customers
Customer X
Customer Y
Customer Z
Customer W
+3
ID1 Event description
A new vulnerability affecting the functions of Product ABC has been discover Read more
Mark as checked
ID2 Event description
...
Checked
ID3 Event description
Message: ...

Appendix E
Communication Function View


COMMUNICATION FUNCTION VIEW
1
Requests
- The purpose of this view is to address the need of filtering and forwarding vulnerability information, so it is appropriately communicated both internally and externally. This by adding the function of sending communication requests within the organisation.
- Depending on organisation structure, e.g. who it is that evaluates and fixes vulnerabilities, the originator of the request can perhaps be a development team or the triage responsible, as is shown in this view. By requesting the communication, describing what information it is that should be sent out and who the recipients should be, the information can be adjusted and formulated by the receiver of the request and passed on. Receiver of the request can perhaps be PR or technical writers.
- Requests themselves are sent from company internal stakeholders, but the communication that is requested might also be intended for both internal and external stakeholders, e.g. emails for customers or publications on the website.
Overview
Requests
Recipients
Manage
Admin tools
Charles Communicator
Requests
Search...
Requests
Summary of all requests
Latest request
Most recently handled request
not handled requests
All 216
Active 7
Not handled 3
Archived 206
Search requests..
Advanced search
25 entries
Request ID
Received
Originator
Subject
Status
Intended recipients
REQ-ID1 Date Silvia Securityexpert URGENT Vulnerability affecting login details... Not handled Board +5
REQ-ID2 Date Development team A Newly discovered CVE-ID patch advisory Not handled Customer Y +3
REQ-ID3 Date Development team B Customer guidelines Not handled Customer support
REQ-ID4 Date Development team B Reported login issues In edit Customer group X
REQ-ID5 Date Silvia Securityexpert Information regarding Heartbleed bug Archived Product owners
REQ-ID6 Date Development team B Password inquiries In edit Customer Z +1

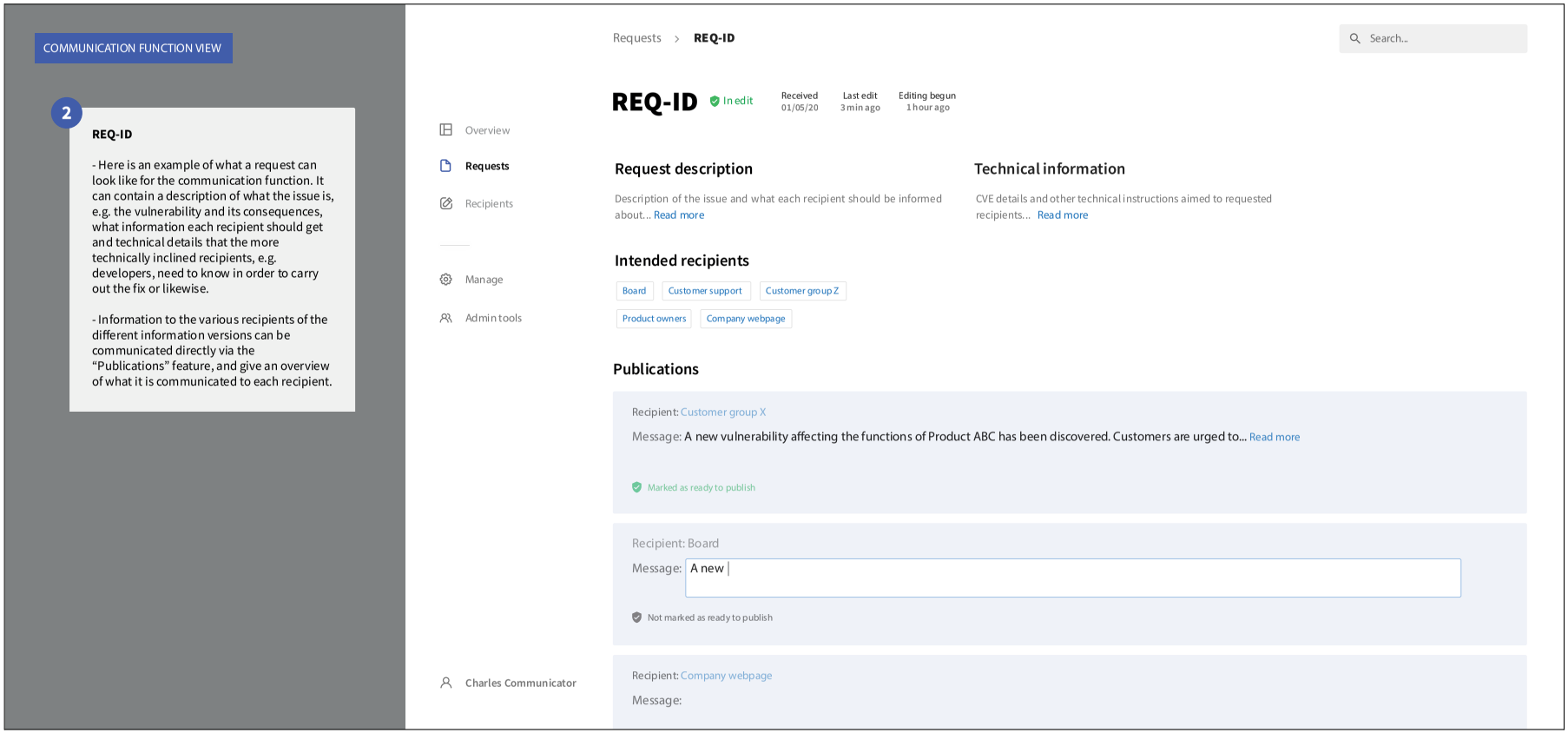
COMMUNICATION FUNCTION VIEW

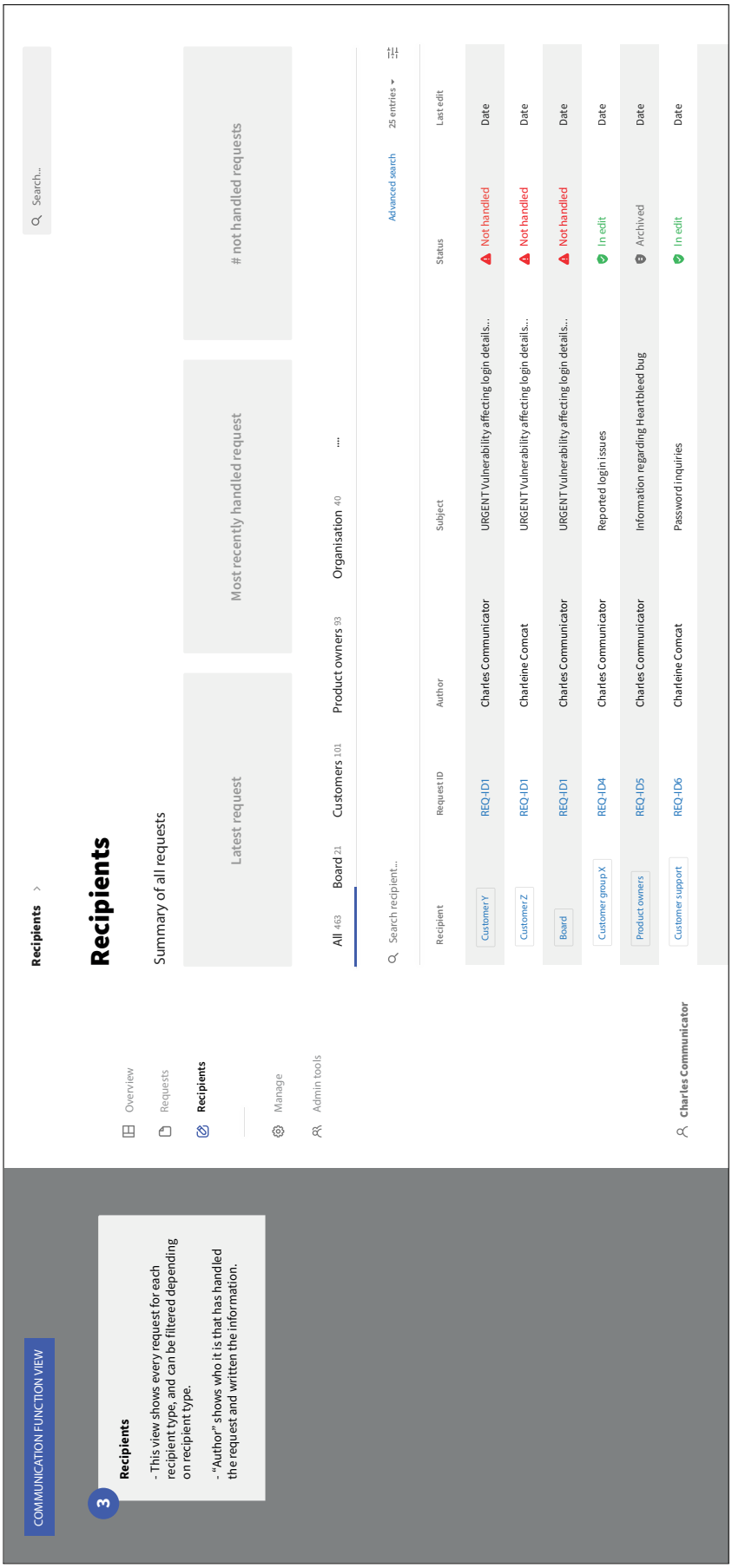
2

REQ-ID

- Here is an example of what a request can look like for the communication function. It can contain a description of what the issue is, e.g. the vulnerability and its consequences, what information each recipient should get and technical details that the more technically inclined recipients, e.g. developers, need to know in order to carry out the fix or likewise.

- Information to the various recipients of the different information versions can be communicated directly via the "Publications" feature, and give an overview of what it is communicated to each recipient.




COMMUNICATION FUNCTION VIEW
3
Recipients
- This view shows every request for each recipient type, and can be filtered depending on recipient type.
- "Author" shows who it is that has handled the request and written the information.
Recipients
Overview
Requests
Recipients
Manage
Admin tools
Charles Communicator
Search...
Recipients
Recipients
Summary of all requests
Latest request
Most recently handled request
not handled requests
All 463
Board 21
Customers 101
Product owners 93
Organisation 40
Search recipient...
Advanced search
25 entries
Recipient
Request ID
Author
Subject
Status
Last edit
Customer Y
REQ-ID1
Charles Communicator
URGENT Vulnerability affecting login details...
Not handled
Date
Customer Z
REQ-ID1
Charleine Comcat
URGENT Vulnerability affecting login details...
Not handled
Date
Board
REQ-ID1
Charles Communicator
URGENT Vulnerability affecting login details...
Not handled
Date
Customer group X
REQ-ID4
Charles Communicator
Reported login issues
In edit
Date
Product owners
REQ-ID5
Charles Communicator
Information regarding Heartbleed bug
Archived
Date
Customer support
REQ-ID6
Charleine Comcat
Password inquiries
In edit
Date

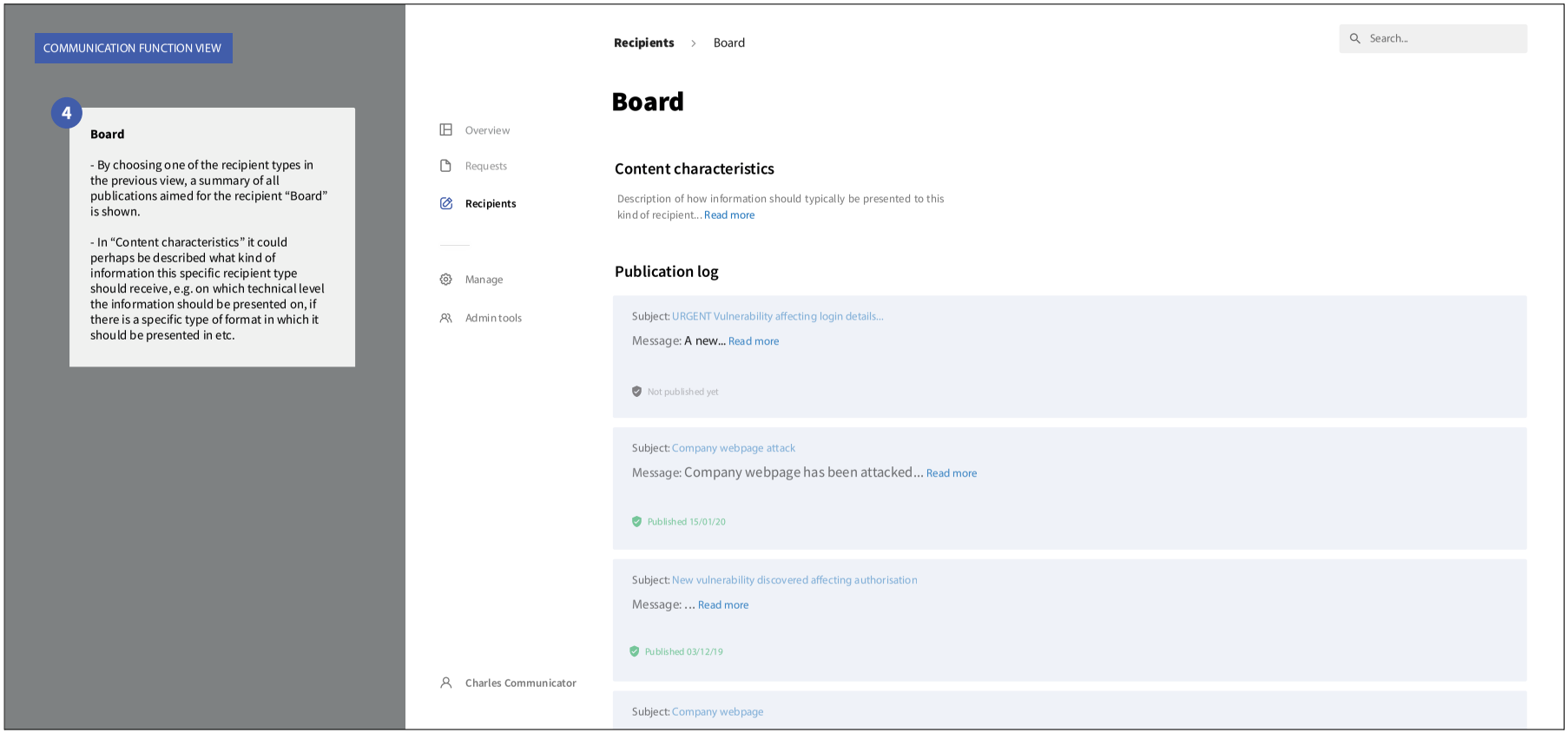
COMMUNICATION FUNCTION VIEW

4

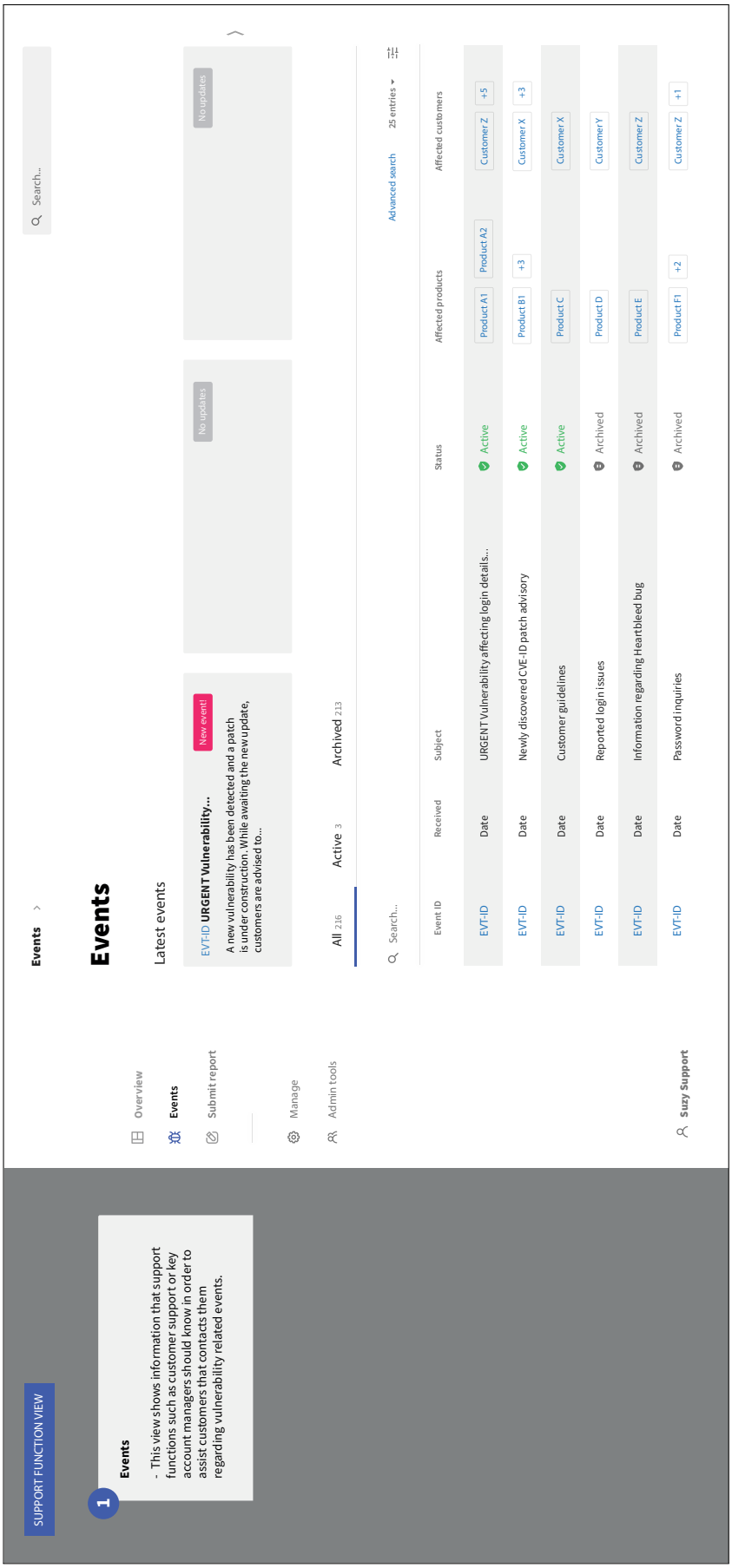
Board

- By choosing one of the recipient types in the previous view, a summary of all publications aimed for the recipient “Board” is shown.

- In “Content characteristics” it could perhaps be described what kind of information this specific recipient type should receive, e.g. on which technical level the information should be presented on, if there is a specific type of format in which it should be presented in etc.



Appendix F
Support Function View

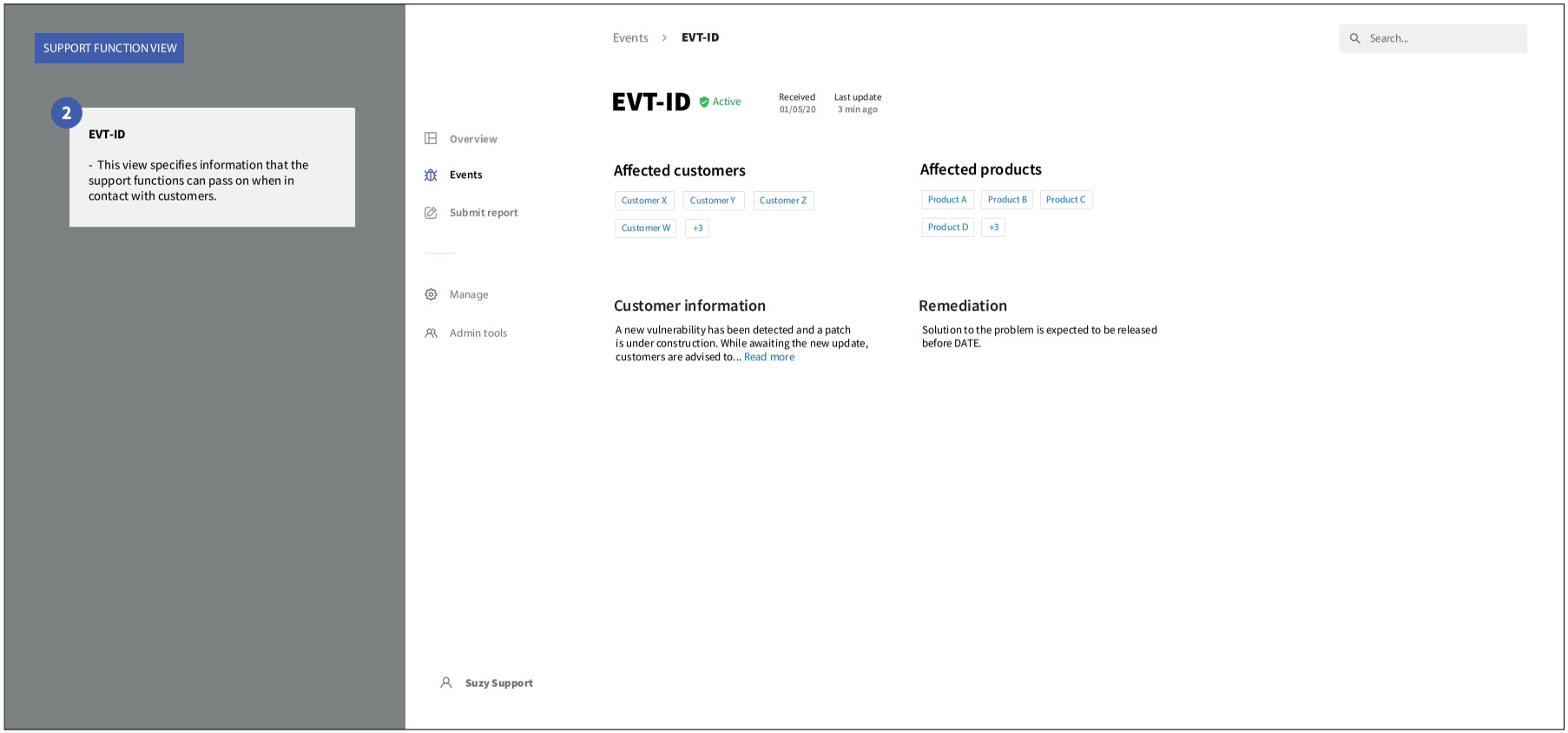

SUPPORT FUNCTION VIEW
1
Events
- This view shows information that support functions such as customer support or key account managers should know in order to assist customers that contacts them regarding vulnerability related events.
Events
Search...
Overview
Events
Submit report
Manage
Admin tools
Suzy Support
Events
Latest events
EVT-ID URGENT Vulnerability...
New event!
A new vulnerability has been detected and a patch is under construction. While awaiting the new update, customers are advised to...
No updates
No updates
All 216
Active 3
Archived 213
Search...
Advanced search
25 entries
Event ID
Received
Subject
Status
Affected products
Affected customers
EVT-ID
Date
URGENT Vulnerability affecting login details...
Active
Product A1
Product A2
Customer Z
+5
EVT-ID
Date
Newly discovered CVE-ID patch advisory
Active
Product B1
+3
Customer X
+3
EVT-ID
Date
Customer guidelines
Active
Product C
Customer X
EVT-ID
Date
Reported login issues
Archived
Product D
Customer Y
EVT-ID
Date
Information regarding Heartbleed bug
Archived
Product E
Customer Z
EVT-ID
Date
Password inquiries
Archived
Product F1
+2
Customer Z
+1

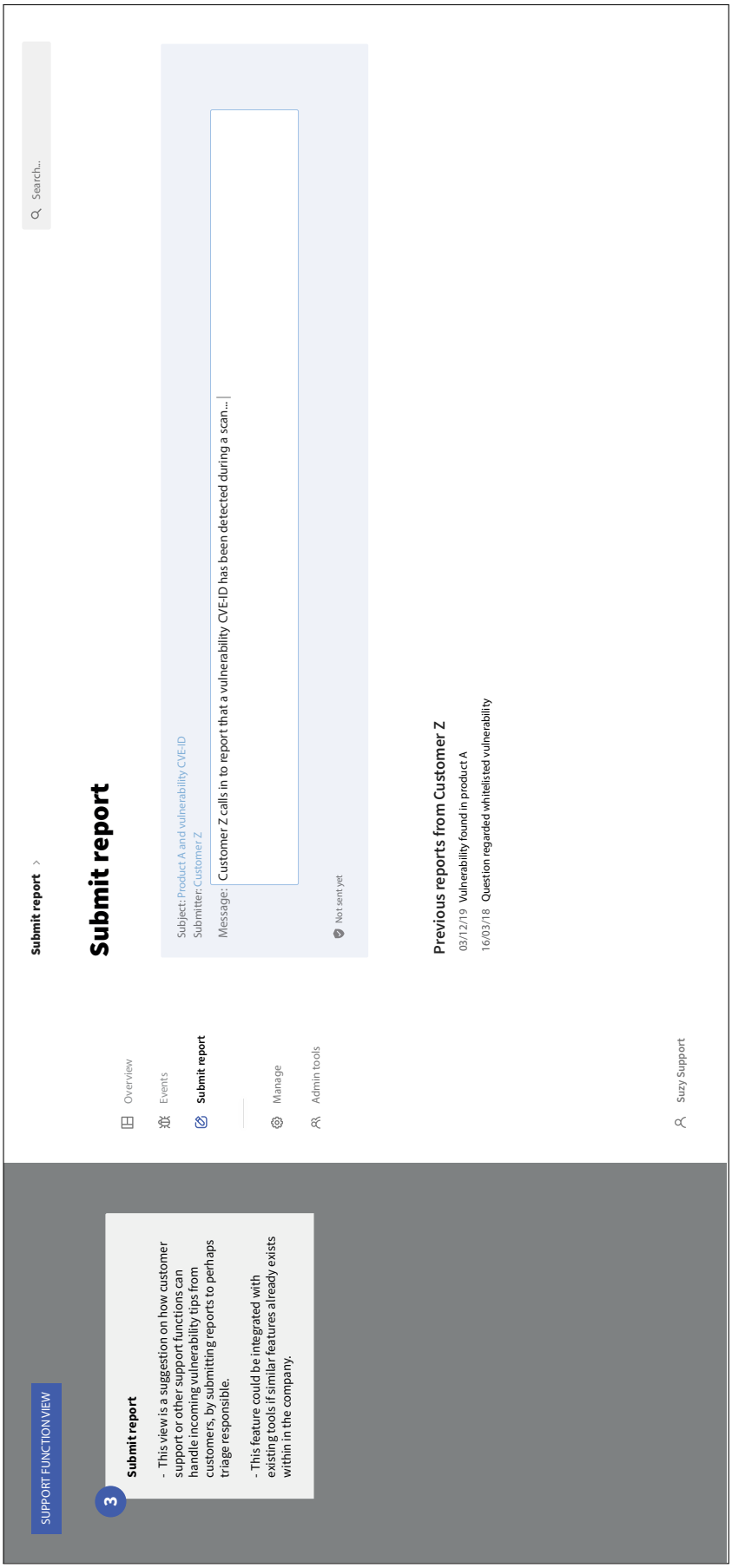
SUPPORT FUNCTION VIEW

2

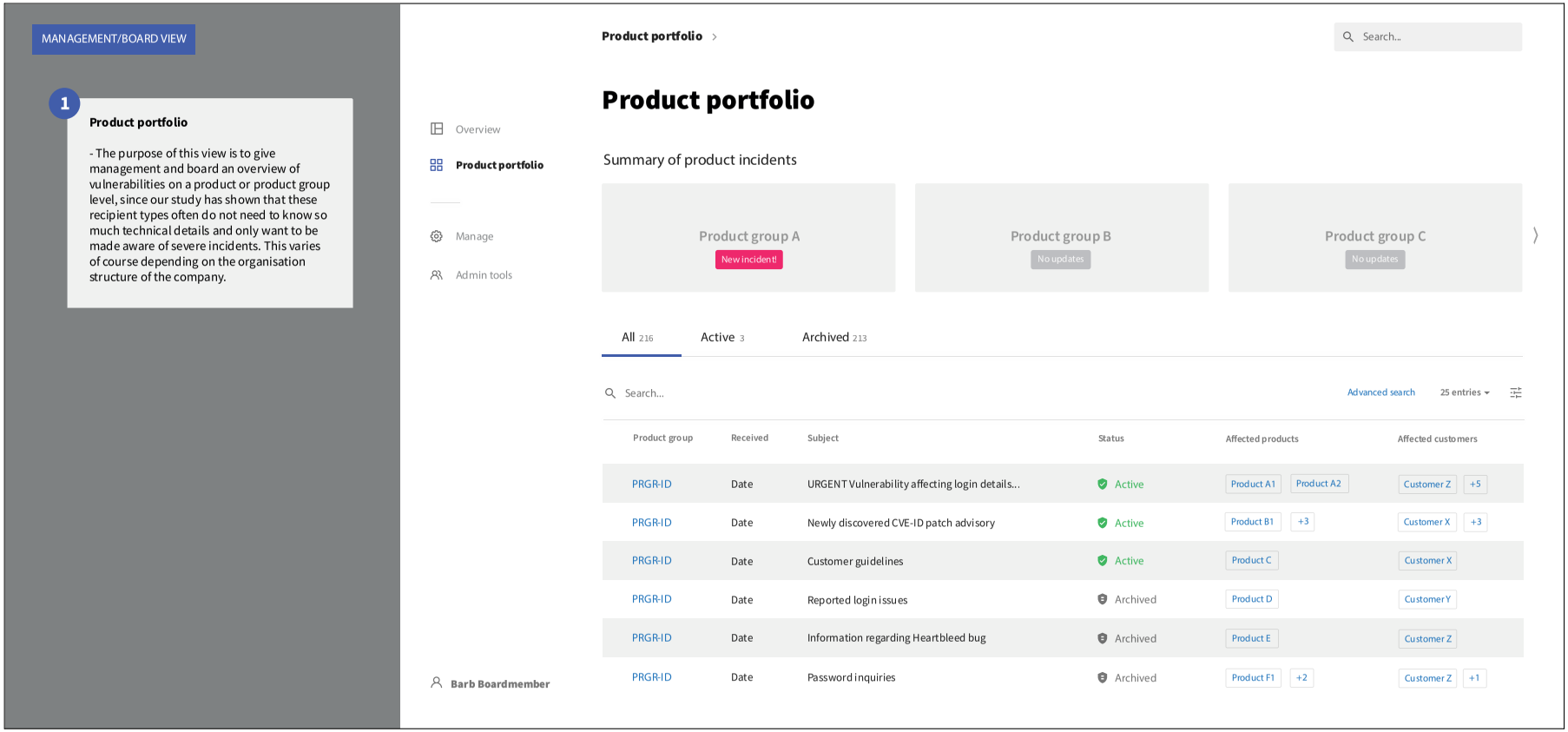
EVT-ID

- This view specifies information that the support functions can pass on when in contact with customers.




SUPPORT FUNCTION VIEW
3
Submit report
- This view is a suggestion on how customer support or other support functions can handle incoming vulnerability tips from customers, by submitting reports to perhaps triage responsible.
- This feature could be integrated with existing tools if similar features already exists within in the company.
Search...
Submit report
Submit report
Overview
Events
Submit report
Manage
Admin tools
Suzy Support
Subject: Product A and vulnerability CVE-ID
Submitter: Customer Z
Message: Customer Z calls in to report that a vulnerability CVE-ID has been detected during a scan...
Not sent yet
Previous reports from Customer Z
03/12/19 Vulnerability found in product A
16/03/18 Question regarded whitelisted vulnerability

Appendix G
Management and Board View


MANAGEMENT/BOARD VIEW
1
Product portfolio
- The purpose of this view is to give management and board an overview of vulnerabilities on a product or product group level, since our study has shown that these recipient types often do not need to know so much technical details and only want to be made aware of severe incidents. This varies of course depending on the organisation structure of the company.
Overview
Product portfolio
Manage
Admin tools
Barb Boardmember
Product portfolio
Search...
Product portfolio
Summary of product incidents
Product group A
New incident!
Product group B
No updates
Product group C
No updates
All 216
Active 3
Archived 213
Search...
Advanced search
25 entries
Product group
Received
Subject
Status
Affected products
Affected customers
PRGR-ID
Date
URGENT Vulnerability affecting login details...
Active
Product A1
Product A2
Customer Z
+5
PRGR-ID
Date
Newly discovered CVE-ID patch advisory
Active
Product B1
+3
Customer X
+3
PRGR-ID
Date
Customer guidelines
Active
Product C
Customer X
PRGR-ID
Date
Reported login issues
Archived
Product D
Customer Y
PRGR-ID
Date
Information regarding Heartbleed bug
Archived
Product E
Customer Z
PRGR-ID
Date
Password inquiries
Archived
Product F1
+2
Customer Z
+1

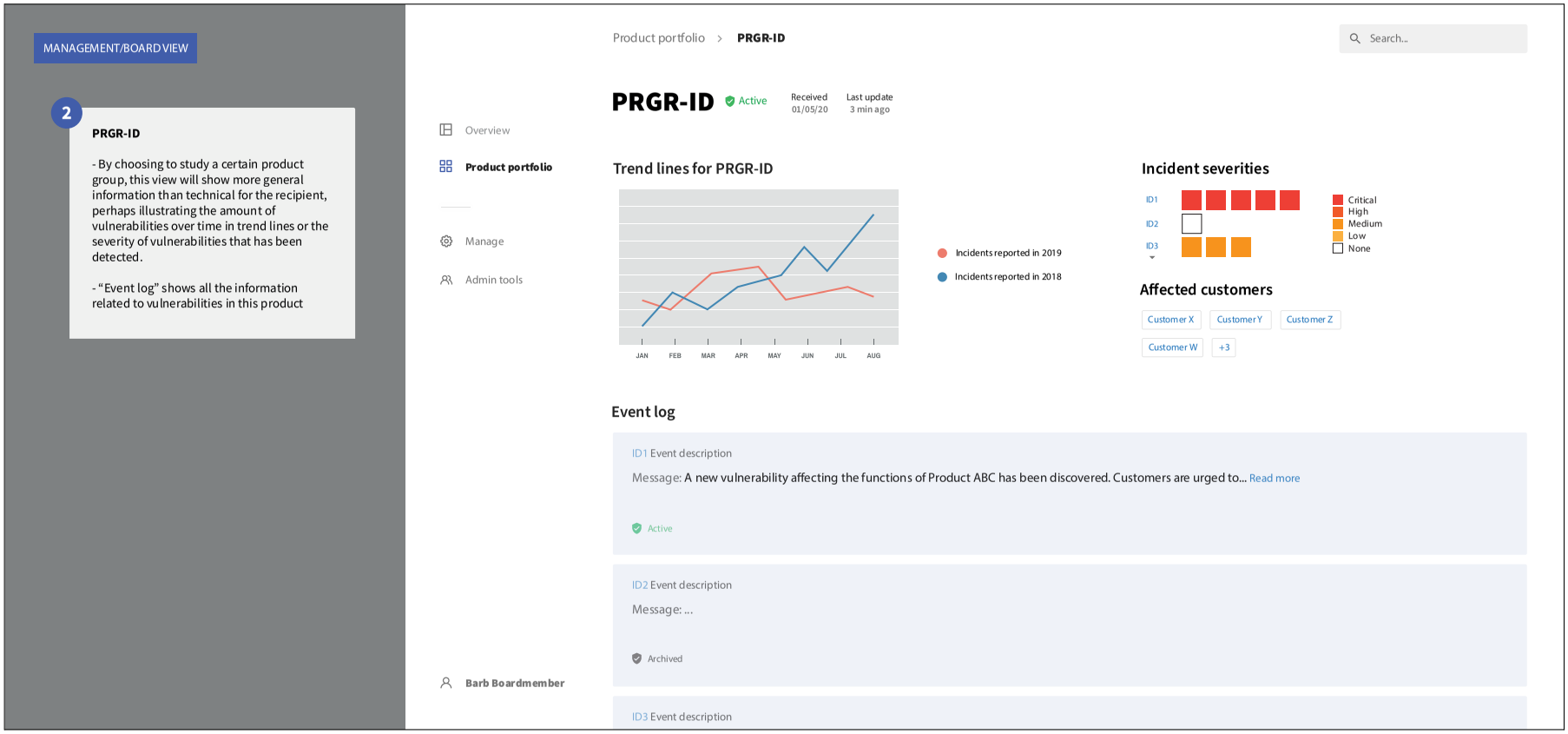
MANAGEMENT/BOARD VIEW

2

PRGR-ID

- By choosing to study a certain product group, this view will show more general information than technical for the recipient, perhaps illustrating the amount of vulnerabilities over time in trend lines or the severity of vulnerabilities that has been detected.

- "Event log" shows all the information related to vulnerabilities in this product



Appendix H
Customer View

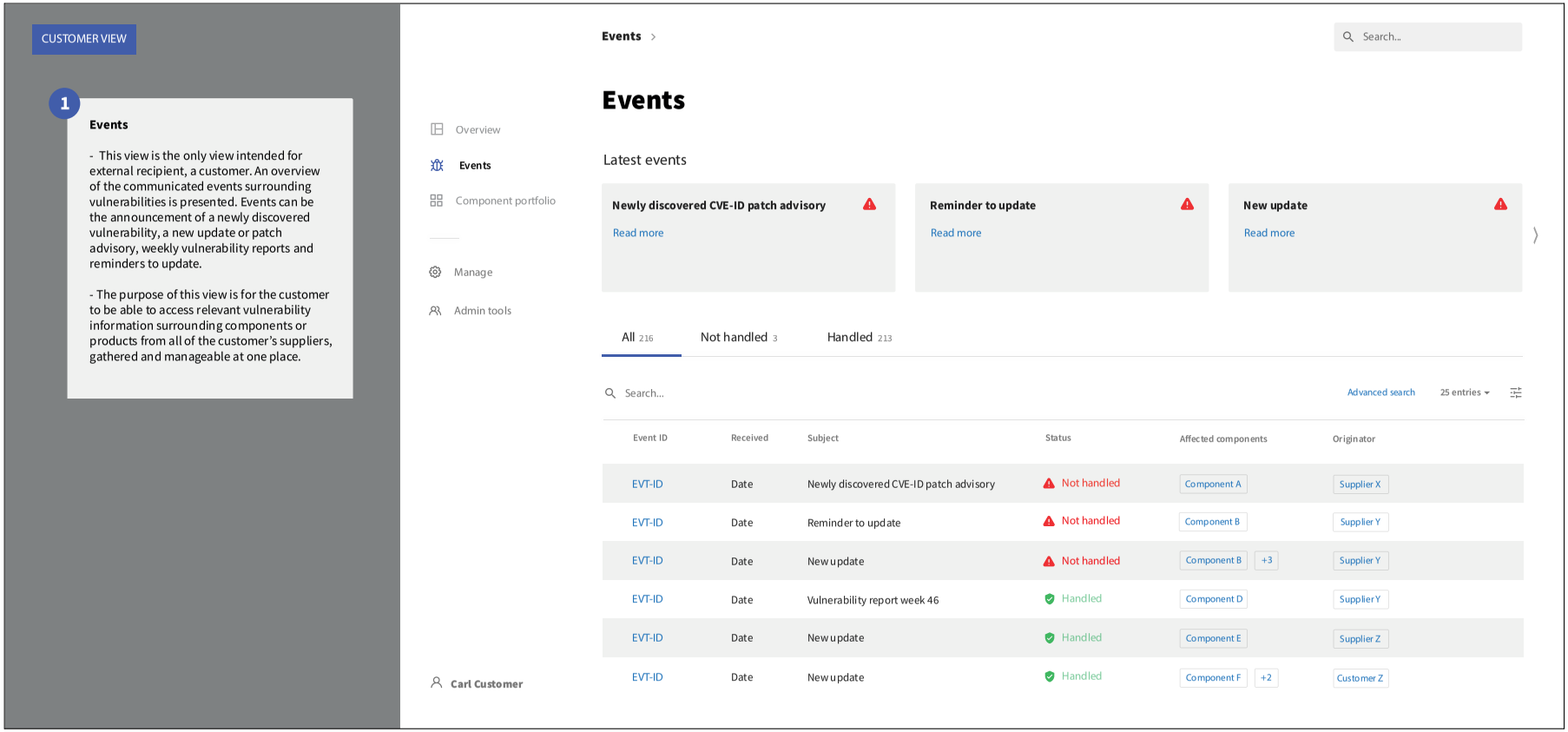
CUSTOMER VIEW

1

Events

- This view is the only view intended for external recipient, a customer. An overview of the communicated events surrounding vulnerabilities is presented. Events can be the announcement of a newly discovered vulnerability, a new update or patch advisory, weekly vulnerability reports and reminders to update.

- The purpose of this view is for the customer to be able to access relevant vulnerability information surrounding components or products from all of the customer's suppliers, gathered and manageable at one place.

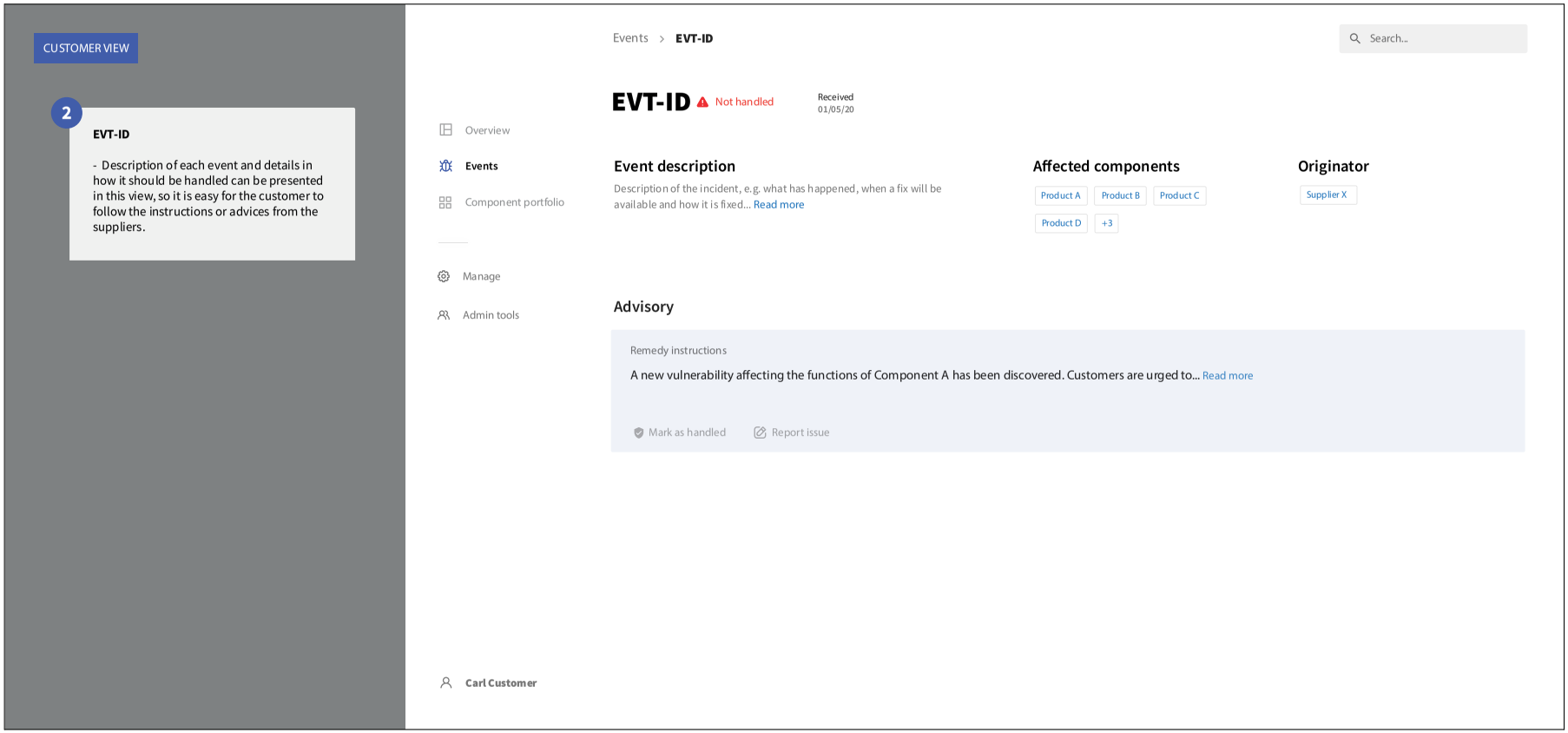


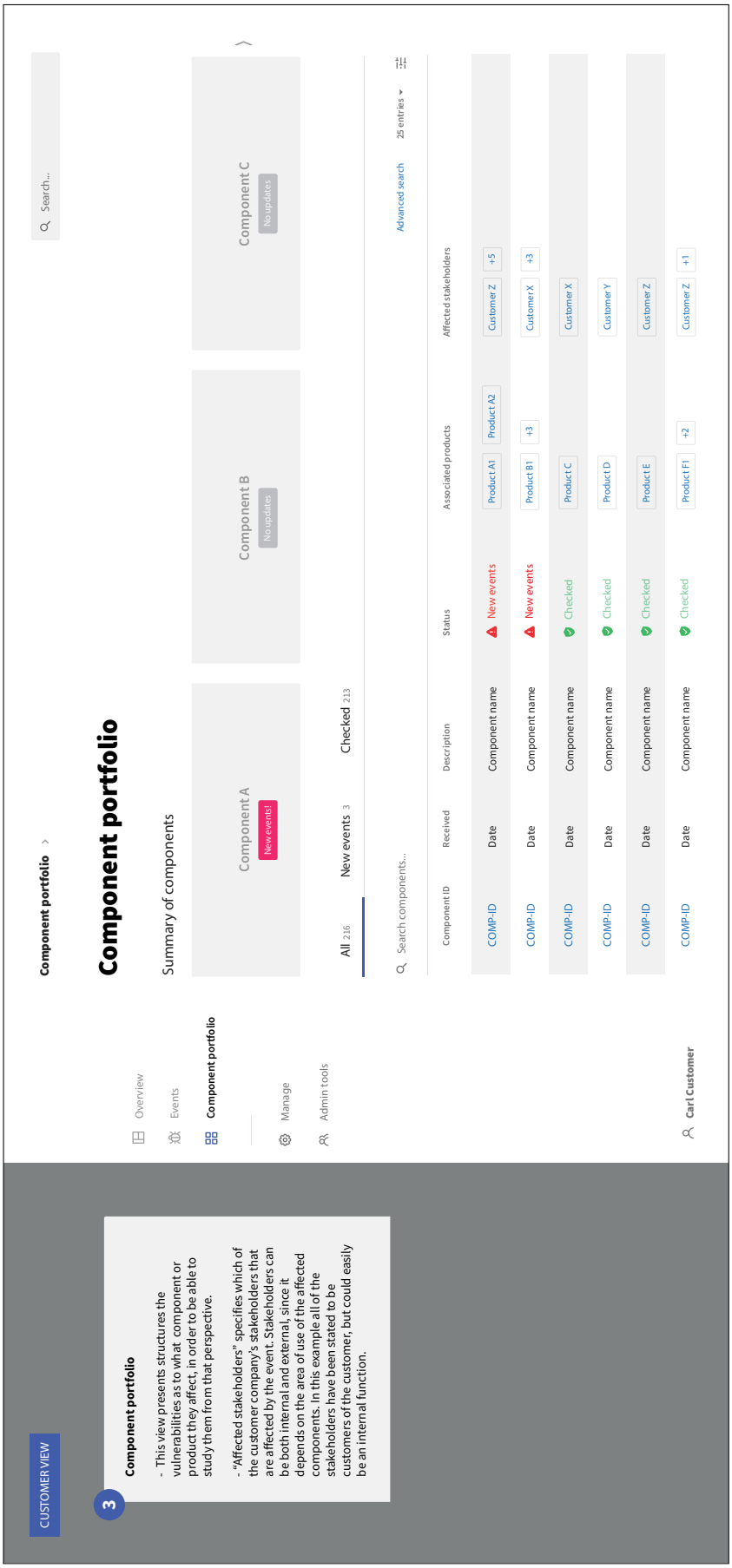
CUSTOMER VIEW

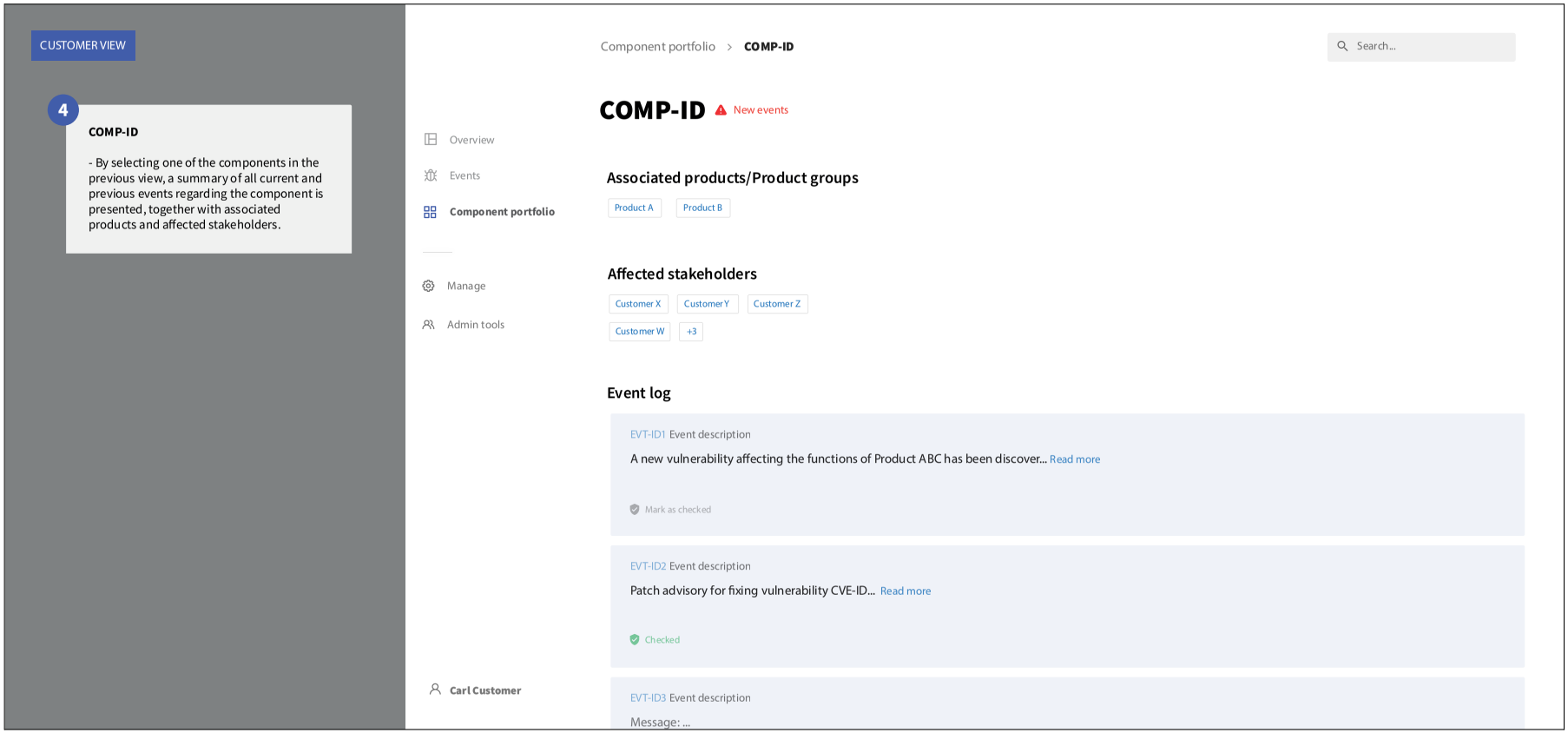
2

EVT-ID

- Description of each event and details in how it should be handled can be presented in this view, so it is easy for the customer to follow the instructions or advices from the suppliers.




CUSTOMER VIEW
3
Component portfolio
- This view presents structures the vulnerabilities as to what component or product they affect, in order to be able to study them from that perspective.
- "Affected stakeholders" specifies which of the customer company's stakeholders that are affected by the event. Stakeholders can be both internal and external, since it depends on the area of use of the affected components. In this example all of the stakeholders have been stated to be customers of the customer, but could easily be an internal function.
Component portfolio
Search...
Overview
Events
Component portfolio
Manage
Admin tools
Carl Customer
Component portfolio
Component portfolio
Summary of components
Component A
New events!
Component B
No updates
Component C
No updates
All 216
New events 3
Checked 213
Search components...
Advanced search
25 entries
Component ID
Received
Description
Status
Associated products
Affected stakeholders
COMP-ID
Date
Component name
New events
Product A1
Product A2
Customer Z
+5
COMP-ID
Date
Component name
New events
Product B1
+3
Customer X
+3
COMP-ID
Date
Component name
Checked
Product C
Customer X
COMP-ID
Date
Component name
Checked
Product D
Customer Y
COMP-ID
Date
Component name
Checked
Product E
Customer Z
COMP-ID
Date
Component name
Checked
Product F1
+2
Customer Z
+1


CUSTOMER VIEW
4
COMP-ID
- By selecting one of the components in the previous view, a summary of all current and previous events regarding the component is presented, together with associated products and affected stakeholders.
Component portfolio > COMP-ID
Search...
COMP-ID
New events
Overview
Events
Component portfolio
Manage
Admin tools
Carl Customer
Associated products/Product groups
Product A
Product B
Affected stakeholders
Customer X
Customer Y
Customer Z
Customer W
+3
Event log
EVT-ID1 Event description
A new vulnerability affecting the functions of Product ABC has been discover... Read more
Mark as checked
EVT-ID2 Event description
Patch advisory for fixing vulnerability CVE-ID... Read more
Checked
EVT-ID3 Event description
Message: ...

DEPARTMENT OF COMPUTER SCIENCE | LUND UNIVERSITY | PRESENTED 2020-06-11

MASTER'S THESIS Sharing is caring: Communicating recipient tailored OSS vulnerability information on an online platform
STUDENTS Emmy Dahl, Michaela Karlsson
SUPERVISOR Martin Hell (LTH)
EXAMINER Martin Höst (LTH)

Sharing is caring

POPULAR SCIENCE PAPER **Emmy Dahl, Michaela Karlsson**

To tackle the increasing number of vulnerabilities in open source software, more and more extensive vulnerability management is needed. Proactive sharing of vulnerability information has never been more important, thus it is of essence for organisations to establish a structured way of communicating vulnerabilities.

Introduction

With IoT and digitisation in general comes initiatives for malicious actors to exploit possible vulnerabilities in the software used. Both the usage of open source software and detected vulnerabilities have increased in the last decade and it is becoming important for companies to know what weaknesses they have in their software systems, to ensure secure products. Presently the communication regarding vulnerabilities in and between organisations is done reactively instead of proactively. This even though research has shown that the total cybersecurity of a product often depends on cooperation between several actors and sensitive information sharing increases the performance of the actors in a network. This insinuates an industrial need for structured communication between organisations regarding software vulnerability management.

Our qualitative case study into the area renders a differentiation of information recipient groups within companies, as well as suggestions on how communication can be handled and in what way suggested information can be presented to various professional roles. Companies often consist of several employees that contribute to the organisational activities in different ways, depending on what their professional roles are. Different professional roles imply varying employee knowledge of software, vulnerabilities and their importance for the companies, therefore each recipient group needs specific vulnerability information. A possible solution that can improve vulnerability management between industry actors is illustrated in Figure 1.

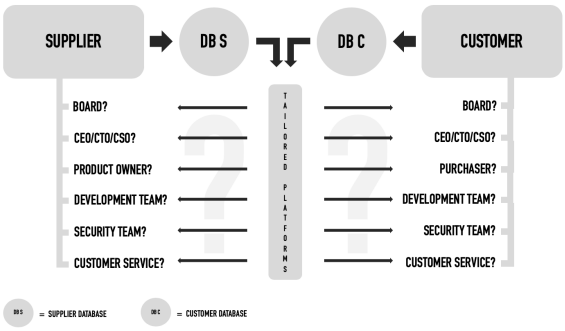


Figure 1: Illustration of the problem formulation.

The Vulnerability Information Recipient Groups

From the study results it is clear that the type of professional roles that are mentioned as partakers in the handling of vulnerabilities vary greatly between different companies. 72% of the mentioned

MASTER'S THESIS Sharing is caring: Communicating recipient tailored OSS vulnerability information on an online platform
STUDENTS Emmy Dahl, Michaela Karlsson
SUPERVISOR Martin Hell (LTH)
EXAMINER Martin Höst (LTH)

roles are in fact only mentioned by one company. This result is not completely unexpected, since the theory shows that the ways of organising and structuring software development are almost endless. The above leads to the conclusion that trying to identify individual professional roles to design platform views to is difficult and some kind of grouping of recipients is necessary to conduct a feasible solution. The analysis of the results show that some recipients are possible to identify as needing certain information. Through this analysis it is therefore possible to group the different professional roles into more prominent recipient groups, that are deemed appropriate for tailoring platform views to. These recipient groups are triage responsible, development team, communication function, support function, management and board and customer.

The Vulnerability Information Recipient Onion Model

In an attempt to illustrate the recipient groups' different information needs, a Vulnerability Information Recipient Onion Model (VIROM) is derived. VIROM is constructed as follows: The more technically detailed information related to vulnerabilities the recipients need in order to carry out their work duties, the further into to the core of the onion the recipients belongs. Moving towards the outer layers of the onion, the degree of necessary software security knowledge for interpreting and making use of the information declines. Such segmentation of recipients based on their knowledge in software security and need of technical information creates a way of sorting them in a information hierarchical way. As a suggestion VIROM consists of three main layers for a company handling vulnerabilities: The Technical layer, the Organisation layer and the Client layer. To the Technical layer belong recipients of vulnerability information within the company that are heavily involved with the technical aspects of development and vulnerabilities, i.e. triage responsible and development team. To the Organisation layer belong recipients of vulnerability information within the company that have responsibilities within business, communication and customer relations, i.e. management and board, communication function and support function. Finally, external recipients of vulnerability information, i.e. customer, belongs to the Client layer. A graphic presentation of VIROM with the recipient groups is depicted in Figure 2.

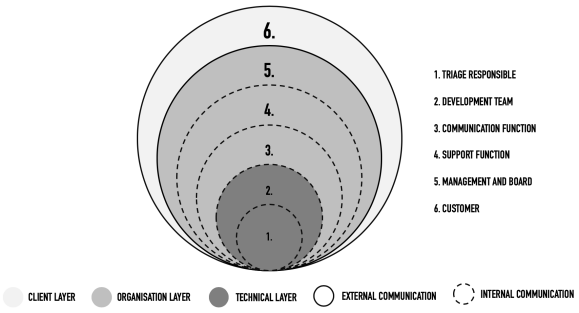


Figure 2: Illustration of final recipient groups sorted according to VIROM.

The results from the analysis of recipient groups and what information they need, result in six different prototype views with suggestions of how vulnerability information can be tailored and presented on an online platform for each recipient group.

Conclusion

The most prominent recipient groups that are identified are triage responsible, development team, communication function, support function, management and board and finally customer. The recipient groups triage responsible and development team need more technically detailed information, because of their deep involvement with the technical aspects of handling vulnerabilities. Communication function, support function and management and board need vulnerability information that is related to their more business, communication and customer inclined responsibilities. Customer needs information that is as straight forward and solution oriented as possible. Finally, prototype views are designed for each prominent recipient group.