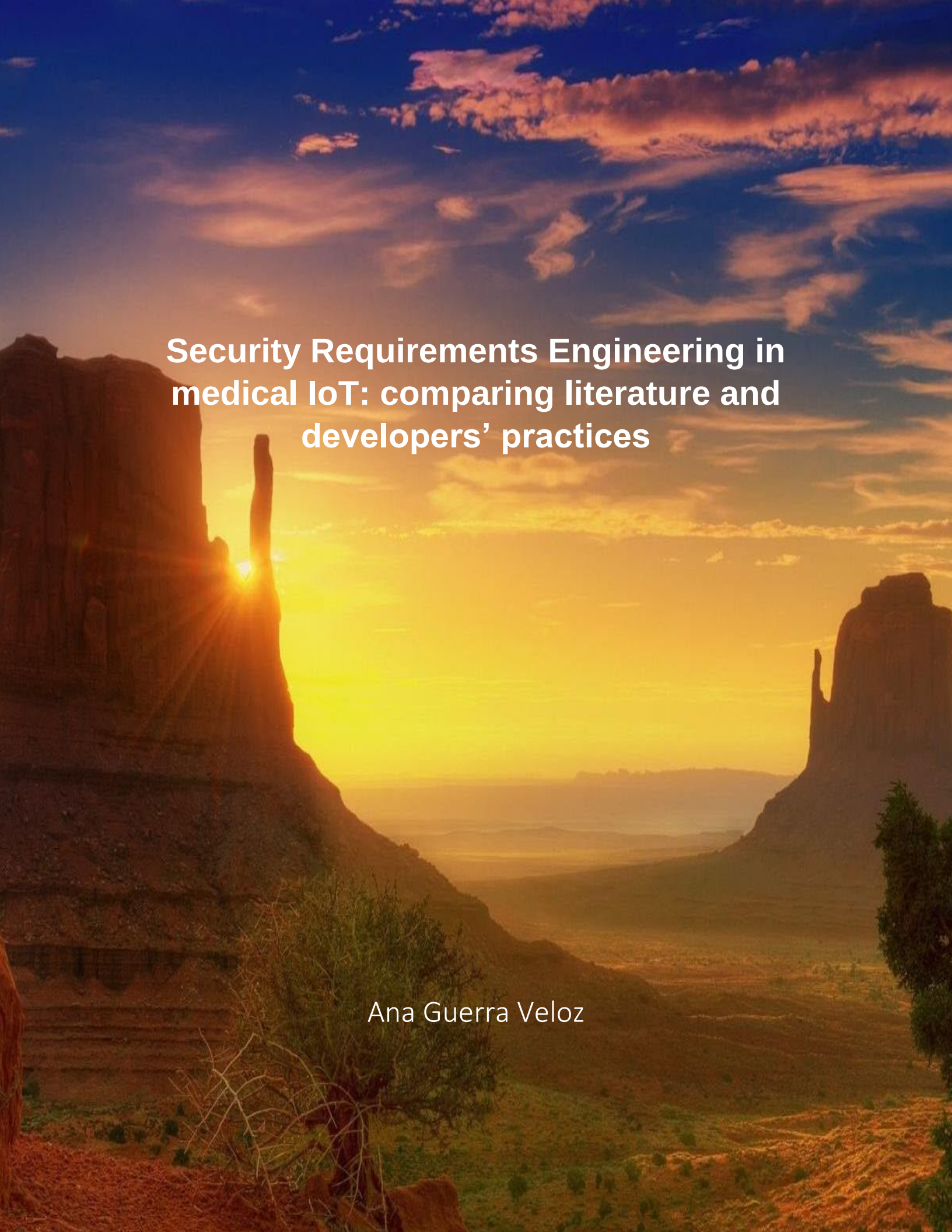

Security Requirements Engineering in medical IoT: comparing literature and developers' practices

Ana Guerra Veloz

Cover illustration taken from http://bsnscb.com

Security Requirements Engineering in medical IoT: comparing literature and developers' practices

Master thesis submitted to Delft University of Technology

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in **Management of Technology**

Faculty of Technology, Policy and Management

by

Ana Alexandra Guerra Veloz

Student number: 4520084

To be defended in public on August 24th, 2017

Graduation committee

Chairperson	: Prof.dr. S. Roeser, Ethics/Philosophy of technology
First Supervisor	: Dr.ir. W. Pieters, Safety and Security Science
Second Supervisor	: Dr.ir. M. de Reuver, ICT section
External Supervisor	: L.V.E. L. Fichtner, ICT section

If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.

Bruce Schneier

Acknowledgment

We don’t know that what we are capable of until we challenge our own limits. I got an opportunity to challenge my limits when I ventured my way to the Netherlands for my higher studies. This thesis project is the final product of my studies at TU Delft. I can say that it has been a wonderful and exciting experience of my life. During this two years, I thoroughly enjoyed my time in the faculty of TPM. Not just the classes but also the exams were as exciting and challenging to test my potential as well as improve my knowledge in total.

First and foremost, I would like to deeply express my gratitude to all the members of my graduation committee. All of the work wouldn’t have been accomplished if it was not for my first supervisor Dr.ir. Wolter Pieters. I would like to thank him for his constructive feedback so as to help me think out of box and improve my research. Secondly, I would deeply like to thank my external supervisor Laura Fichtner. Her constant support and commitment towards the research work was quite commendable. From bottom of my heart, I would like to thank you for guiding me through my thoughts and always giving me a positive comment which constantly motivated me throughout my research work. I am pretty certain that your research work that you’re working on is going be flawless and world-class. Thirdly, I would like to thank my second supervisor Dr.ir. Mark de Reuver for always being willing to offer me his guidance and opinion throughout the research work. Last but not the least, I would like to thank my chairperson Prof.dr. S. Roeser. Thank you for accepting to be my chair and for providing me with valuable feedback.

The research work would not have been successful if it wasn’t for innovative developers and managers who participate in the study. Their willingness to share with me their experiences, views, challenges, hopes, and dreams helped me a lot for research work. I hope you the best in the challenging world of technology development.

Next, I would like to thank the scholarship program by the Secretariat of Higher Education, Science, Technology and Innovation of the Republic of Ecuador who has provided me with financial support for conducting my master studies at the University of TU Delft. I hope that the knowledge I acquired during my studies will be beneficial to the future of my beautiful country.

Finally, I would like to thanks to my loved ones especially to my lovely husband, *Fabricio*. Without your unconditional support, I would not have been able to undertake this journey. I am sure we have a great future together. Moreover, I would like to thanks to my family *Alicia*, *Juan Manuel*, *Maria Fernanda*, y *Juan David*. Thank you for believing in me!! And to my friends at the MOT. I enjoyed so much our time together.

Delft, August 2017

Ana Guerra

Summary

The connection of smart objects to the Internet provides opportunities for innovative services and applications in almost every field. However, this network of objects brings serious ***security issues*** to users, society, and even to the internet. Billions of things connected to the internet turn out to be an attractive target for hackers and a doorway into the information technologies' infrastructure and personal data. In the last years, malicious actors have been able to exploit vulnerabilities of IoT-enabled devices and network. Baby monitors and internet-enabled cameras that expose consumers' private lives on the internet or internet-connected devices that perform a DDoS attack shows that cyberattacks to the IoT are a reality. Nevertheless, the number of IoT devices and applications is growing, and it will continue to grow by offering inexpensive devices with a lot of features.

During the applications' development, functional requirements – i.e. what the system does – are fundamental to attract customers and satisfy their needs. But **security**, which is a non-functional requirement, is not a basic application's requirement. Besides, security and security requirements are not always well defined or understood by producers. Security experts argue that an effective way of guaranteeing security in information systems is to include focused security practices in the development cycle by considering security requirements early in the development process. The security requirements engineering (SRE) field provides frameworks, techniques, and methods for addressing security issues during the early phases of development. However, before providing a method to address security requirements of IoT, the SRE field needs to understand developers of IoT applications, their needs, and motivations to address security. Thus, developers' practices to address security during the development process should be brought into the field of SRE. This challenge is the main goal of this thesis and is formalized by the following research question. For achieving the purpose of this thesis, we concentrate on medical IoT applications.

How do developers' security practices in the design and development of IoT medical applications challenge security requirements engineering methods?

The first part of the research question focusses on understanding the security practices recommended by the field of SRE. Based on the SRE literature, four tasks are highlighted to accomplish a secure development. First, the **elicitation of security requirements** which involves the identification of stakeholders and system's security goals and its operationalization into security requirements. Second, the **analysis of requirements** to derive a set of categorized security system requirements. Third, the **management of requirements** to integrate functional, non-functional requirements, and security system requirements into the application. Besides, security mechanisms are architected to fulfill one or more security requirements and to reduce security vulnerabilities. Finally, **requirements validation** ensures that security requirements satisfy security goals and **requirements verification** verifies that security mechanisms fulfill security requirements.

For steering the data collecting of developers' actions to handle security and the data analysis, an interpretative framework is developed. The process of building IoT applications occurs in an organizational context i.e. within companies. Then, organization activities are divided into two main levels *managerial* and *operational*. The *managerial level* concerns managerial decisions and actions that shape a secure product development, and the *operational level* concerns procedures, actions, and activities adopted by developers to perform a secure development. By following these definitions, we assume SRE recommended practices as activities at the *operational* level.

A qualitative study based on interviews and documentation analysis of three companies developing IoT medical applications (2 startups and 1 established company) were employed to gather data on developers' practices. By following grounded theory principles and the coding paradigm model, the data were analyzed.

Our results show that the interviewed small companies do not have a distinctive process to handle security requirements. The **elicitation of requirements** is focused on functional requirements and some non-functional requirements such as *usability*. The **analysis of requirements** involves the prioritization of application's functionalities rather than security requirements. The **management of requirements** involves incorporating required or well-known security mechanisms (e.g. encryption to prevent unauthorized access) to the application. Finally, **the validation and verification of requirements** focuses on providing evidence to regulatory authorities to make sure that applications meet required security requirements. These processes occur during the differences stages of the development process or when the product is already developed. From our analysis, factors such as *developers' perception of security, rules and regulations, hospital security knowledge, and development approaches* could be **considered as causal conditions for addressing security**. These factors lead to the adoption of **strategies to incorporate security requirements** into the application. Strategies such as *involve reliable stakeholders*, *incorporate security requirements when it is required*, and *adjust technology features* are commonly employed. Factors such as *developers' perception, customers' needs, challenges faced during the development, and companies experience* can be considered as **intervening conditions** which facilitate or constrain the adoption of the developers' strategies. The conditions for addressing security and the strategies are influenced by the *setting where the application operates* and the *stage of development* of the medical application, which are part of the **context**.

By comparing SRE recommended practices and developers' practices for handling security we can obtain preliminary results regarding the participating companies. Nevertheless, our results are exploratory, and some of them will need to be validated for large populations. Results show that most of the **developers' practices do not match SRE recommended practices**, and thus, the process described by SRE is deficiently adopted. Stakeholders and developers do not express their security concerns regarding an asset. Security goals are not stated explicitly, then, security requirements cannot be elicited because there is not a security goal to fulfill. Developers tend to elicit basic security requirements or security mechanisms according to their experience. Deriving a set of security requirements which are the product of the analysis of stakeholders' concerns is not part of the development process. Neither prioritizing system security requirements by conducting a risk assessment. Security mechanisms are elicited directly at any stage of development. For entering the healthcare market, security mechanism comes from rules, regulations, and standards. It seems that security requirements are not validated because security goals are not clearly established.

These differences exist because SRE recommended practices assume that stakeholders bear in mind security concerns regarding assets of the system to be, and companies include security expertise during the requirement engineering process. However, our finding suggests that developers primarily be concerned with functional requirements and security expertise is not always part of the development process. Then, *developers' perception of security, rules, and regulations, hospital security knowledge* motivate developers to address security in the applications' development. SRE process takes security requirements into account early in the development process. But, in practice, developers employ different strategies to incorporate security requirements into the application. Strategies might be adopted at any time in the development process. SRE process does not consider the impact that factors such as *developers' perception of security failures, customers' needs, challenges faced during the development, and companies' experience* have over developers' strategies.

The main implications for SRE frameworks lie in the incorrect assumptions regarding developers' motivation to address security, the need for methodologies that better meet interactive and incremental product development approaches, and SRE methods that consider the dynamic nature of security.

The latter can be done by including guidance to update security requirements and mechanisms after some time. Moreover, among the implications to developers, our findings suggest that an environment that motivates to produce secure products within companies is needed. Managers, engineers, and designers need to be aligned to develop secure applications.

Contents

List of Figures

List of Tables

1

Introduction

Since the emergence of commercial internet service providers in the 1990s (Coffman & Odlyzko, 2002), the internet has dramatically improved the way in which people share resources, access to information, and communicate. Now, the next evolution of the internet is the Internet of Things (IoT) (Evans, 2011; Jadoul, 2015). In the IoT paradigm, everyday objects communicate with each other to form a worldwide dynamic network (Borgia, 2014). In this connected ubiquitous universe, machines and humans will interact to manage resources efficiently (Jadoul, 2015). Nevertheless, such as dynamic network also brings serious *security issues* to users, society, and even to the internet (Covington & Carskadden, 2013). To take advantage of the growing opportunities that the IoT brings to users and businesses, a secure development of the IoT devices, applications, and services must be encouraged. A secure system development can be achieved by disseminating knowledge of security and development among academy and industry. Developers have relevant experiences and ideas that can inform academic research, and researchers have frameworks, tools, and studies that could benefit developers (Cunningham, Gupta, Lindqvist, Sidiroglou-Douskos, & Hicks, 2016). In this thesis, we aim to explore the development of IoT applications for healthcare to gain insights on how security practices of developers challenge security literature. Thus, the objective is to better understand the differences between security literature and developers' practices to manage security requirements. This section begins with a general introduction of the properties that make the IoT vulnerable to cyberattacks and the security issues. This leads to the practical and scientific gap which is the basis and justification for this thesis. The section concludes with the specification of the research questions and research approach.

1.1 Cybersecurity in the Internet of Things era

Smart devices and applications are increasingly spread into many spheres of everyday life (Sadeghi, 2016) personal gadgets such as fitness trackers, smart appliances, and smart medical devices are some example. However, the proliferation of objects connected to the internet also attracts malicious intentions. One of the first public cases of cyberattacks to the IoT was the case of TRENDNet. TRENDNet is a firm that sells baby monitors and Internet-enabled security cameras. Even though the devices were publicized as secure, the Federal Trade Commission found that "*the cameras had faulty software that left them open to online viewing, and in some instances listening, by anyone with the cameras internet address."* (Federal Trade Commission,

2014b, p. 18). Following a public comment period, the Commission approved a final order settling charges against TRENDNet. The Commission alleged that the negligent security practices of the company led to the exposure of the consumers' private lives on the internet for public viewing (Federal Trade Commission, 2014a).

In October 2016, a huge cyberattack took down some of the world's most popular websites, including Netflix, Twitter, Spotify, CNN, PayPal, Pinterest, Fox News, the Guardian, the New York Times and the Wall Street Journal. The attack appears to be focused on Dyn, a company that runs the domain name system of the internet (Thielman & Johnston, 2016). To perform this attack, hackers used hundreds of thousands of internet-connected devices, such as webcams and digital video recorders, which were previously infected with a malicious code called Mirai to force an especially potent DDoS attack to Dyn (Thielman & Johnston, 2016). Many of the products utilized for the attack originated from the high-tech company XiongMai Technologies and from other makers of inexpensive mass-produced IoT devices, which will remain a danger to others systems for a long time (Krebs, 2016).

Moreover, newspapers regularly report security flaws in internet-connected children's toys (Hautala, 2017; Yandron, 2016); and software-based network-connected smart pacemakers, insulin pumps, and x-ray machines (Zetter, 2017). All the examples mentioned above show that cyberattacks to the IoT are a reality. Malicious actors can take advantage of the security deficiencies of devices, networks, and applications. In the following section, we will explain why the Internet of Things is vulnerable to attacks.

1.1.1 Why IoT is prone to cyberattacks?

The Internet of Things refers to a "dynamic global network and service infrastructure of variable density and connectivity enabling services by interconnecting things" (Tarkoma & Katasnov, 2011, p. 5). Things become "smart" by incorporating sensing, processing, and acting capabilities (Minerva, Biru, & Rotondi, 2015). These smart things are able to collect information from the environment and interact/control the physical world (Borgia, 2014). In addition, when these things are connected through the Internet, things can interact with each other with or without human control and exchange information (Borgia, 2014; Li, Xu, & Zhao, 2015). Although this dynamic network of connected things can be used to create tailored services and applications which satisfy users' needs (Borgia, 2014), it also opens plenty opportunities for exploitation and abuse.

Billions of connected things which automate critical processes and manage sensible data are an attractive target of security attacks (Cheng & Atlee, 2007). An estimated 4.9 billion devices are connected to the internet, and this number is likely to increase to 25 billion by 2020 (van der Meulen & Rivera, 2015). Major IT players are actively developing IoT software platforms, protocol stacks, and cloud services. IBM Waston, Intel IoTivity, Microsoft Azure are some examples (Kolias, Stavrou, Voas, Bojanova, & Kuhn, 2016; Zitnik, Jankovic, Petrovcic, & Bajec, 2016). The expansion of the IT infrastructure plays a fundamental role to accelerate the application and adoption of IoT (Li et al., 2015). However, the increasing population of internet connected (smart) devices also represents an expansion of the attack surface because of the growing number of potential targets across the internet and within specific environments (Covington & Carskadden, 2013). Things that sense, process data, and share information through the internet generate vast

volumes of data (Gubbi, Buyya, Marusic, & Palaniswami, 2013) which becomes attractive to hackers, adversaries, marketers, and even governmental surveillance services.

Moreover, the intrinsic nature of the IoT makes it extremely vulnerable to attacks and create several security issues. First, IoT devices spend most of the time *unattended*, and thus, physical attacks are easily performed (Atzori, Iera, & Morabito, 2010). Besides, IoT devices have a single use (e.g. blood pressure or heart monitors). Thus, users can be profiled due to the detection of unique patterns of specialized devices (Babar, Mahalle, Stango, Prasad, & Prasad, 2010). Second, IoT devices typically connect to the internet via a broad range of *wireless technologies*. Any closer observer can intercept unique low-level identifier that is sent using this type of communication (Atzori et al., 2010; Babar et al., 2010). Third, IoT devices are *mobile* and often connected to the internet via a large set of service providers (Babar et al., 2010). Data and systems can quickly shift between environments making harder to establish appropriate access control, monitoring, and automated decision-making within bounded domains of visibility and control (Covington & Carskadden, 2013). Finally, most of the IoT elements are characterized by *low capabilities* regarding computing resources and energy. Thus, these elements cannot execute complex schemes to support security (Atzori et al., 2010) and it is harder to keep the software up-to-day in these devices (Keoh, Kumar, & Tschofenig, 2014). Moreover, due to the components' *diversity* security and privacy designs needs to accommodate a range of computational capabilities from PCs to low-end RFID tags (Babar et al., 2010).

Estévez-Reyes (2016) argues that IoT magnifies and reinforces existing information and communication technologies security issues. The expanding mass of 'things' will use software to process and communicate, and thus, vast amounts of software will be written. However, market failures led to insecure software; flawed code creates vulnerabilities and vulnerabilities are exploitable. Therefore, the IoT will also be exploitable (Estévez-Reyes, 2016). For instance, cloud-based IoT platforms are prone to IoT security issues but also to traditional web and network security attacks (man-in-the-middle, DoS, eavesdropping, spoofing) (Mineraud, Mazhelis, Su, & Tarkoma, 2016). Lastly, another issue is that security has not been considered in the design of devices which traditionally are not networked (Kocher, Lee, McGraw, & Raghunathan, 2004). This is because, in the past, many embedded systems had very limited connectivity and operated in controlled environments (Gürgens, Rudolph, Maña, & Nadjm-Tehrani, 2010). Nevertheless, today's challenge is to design system-scalable applications and services for a large number of connected devices while taking into account its heterogeneity and limitations (Sadeghi, 2016).

1.1.2 Security issues in the IoT

Security is one of the most important challenges in the development, deployment and widespread adoption of IoT technologies and applications (Borgia, 2014; Da Xu, He, & Li, 2014; Li, Tryfonas, & Li, 2016; Minerva et al., 2015; Miorandi, Sicari, De Pellegrini, & Chlamtac, 2012; Sadeghi, 2016). The Open Web Application Security Projects (OWASP) classified the Top 10 security issues associated with IoT as follows: insecure web interface, insufficient authentication/authorization, insecure network services, lack of transport encryption, privacy concerns, insecure cloud interface, insecure mobile interface, insufficient security configurability, insecure software/firmware, and poor physical security (OWASP, 2011). Figure 1 classified these security issues as data confidentiality, privacy, and trust (Li et al., 2016).

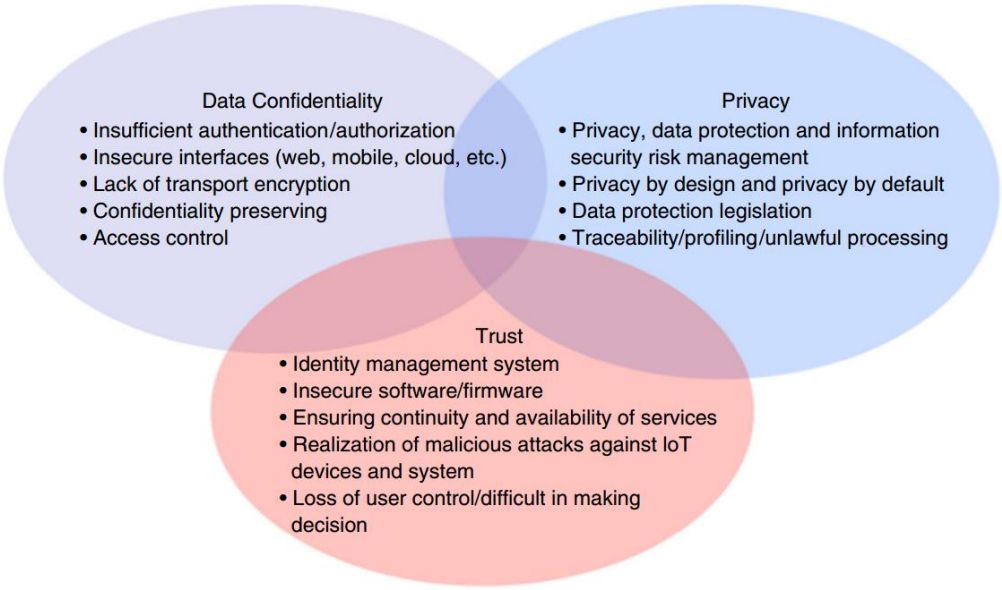

Figure 1: Security issues of the Internet of Things (Li et al., 2016)

Traditionally, security meant cryptography, secure communication, and privacy assurances (Li et al., 2016). However, as discussed in section 1.1.1, due to the extension of the internet to an internet of smart objects, the intrinsic nature of the IoT, and the reinforcement of existing IT security issues; security in the IoT includes a broad range of task. For instance, privacy protection, data confidentiality, access control for accessing networks and services, secure software development, hardware security, software update management, etc. (Keoh et al., 2014).

1.2 Practical Problem

Many smart and rarely updated devices in the Internet of Things contain a range of security vulnerabilities (Markowsky & Markowsky, 2015). These vulnerabilities make devices an easy target for exploitation. For instance, Kolias, Stavrou, Voas, Bojanova, & Kuhn (2016) found three severe and easy to abuse security and privacy threats in IoT applications: leakage of personally identifiable information, leakage of sensitive user information, and unauthorized execution of functions. By using simple and low-cost IoT versions of a motion-sensing light switch, a remote watering system, and an automatic control device; the authors show how fast production of IoT technologies is leaving users vulnerable to security and privacy risk (Kolias et al., 2016).

The Internet of Things is growing, and it will continue to grow by offering inexpensive devices with a lot of features. Fast production and low-cost devices encourage rapid consumer's adoption, but not always secure products or services (Estévez-Reyes, 2016). Besides, IoT security is not always well defined or understood by producers (Kolias et al., 2016). The Internet of Things consist of different technologies, and the integration of these diverse technologies brings extra complexity and uncertainty. In this sense, Alqassem & Svetinovic (2014) argue that *security requirements are not correctly handled* during the applications' development (Alqassem & Svetinovic, 2014).

Furthermore, for some developers of devices and applications, security is not a primary requirement nor a decisive one. It is argued that *security is hardly ever at the forefront of stakeholders*, except maybe to conform to basic standards (Viega, 2005). In the best scenario, security requirements tend to be added at a late stage in the development. In the worst, security controls are coded during the system‘s implementation (Crook, Ince, Lin, & Nuseibeh, 2002).

Authors from the field of requirements engineering agree that security requirements have to be considered from the very beginning in the design process to produce secure devices and applications (Firesmith, 2003; Haley, Laney, Moffett, & Nuseibeh, 2008; Mead & Stehney, 2005; Mellado, Blanco, Sánchez, & Fernández-Medina, 2010; Tondel, Jaatun, & Meland, 2008). Requirements are the starting point in any development process, and thus, a secure development begins by eliciting the security requirements that an artifact needs to fulfill. However, it seems that developers struggle to manage the security requirements of IoT applications. Then, the question becomes *how developers deal with the security requirements of IoT applications during their lifecycle from the initial design phase to the services running.*

1.3 Scientific gap

Security researchers argue that an effective manner of guaranteeing security in Information Systems is to include focused security practices in its development cycle (Mavropoulos, Mouratidis, Fish, Panaousis, & Kalloniatis, 2016). Practices will ensure that IoT devices and application meet specific security standards (Mavropoulos et al., 2016). The practice of how to include security practices in the development cycle is promoted by the field of security requirements engineering (Mavropoulos et al., 2016).

Security Requirements Engineering (SRE) provides techniques, frameworks, methodologies, tools, and norms for addressing security issues during the early phases of the information systems development cycle (Mellado et al., 2010). Before the system’s design is fixed, a coherent set of security requirements for the whole system must be established (Fabian, Gurses, Heisel, Santen, & Schmidt, 2009). This set of security requirements has to be consistent within itself and with the different types of requirements which are essential for the systems. Besides, the set of security requirements needs to be as complete as possible. Security requirements have an effect on functional requirements – i.e. what the system does – which in turn, and hopefully, determine the systems’ design (Fabian et al., 2009).

Although much has been written regarding SRE methods, elicitation and management techniques, and tools for facilitating the analysis of requirements; there have been few attempts from the security requirements engineering community to focus on new applications domain such as Internet of Things (Daneva, Damian, Marchetto, & Pastor, 2014). One exception is Mavropoulos et al., who have approached security issues in the IoT through the lens of Security Requirement Engineering. The authors introduce APPARATUS as a conceptual model for thinking and understanding security in the IoT. “Apparatus is architecture-oriented model and describes an IoT system as a cluster of nodes that share network connections” (Mavropoulos et al., 2016, p. 224). The security analysis is done by emulating a managed environment with known variables (Mavropoulos et al., 2016).

Furthermore, there exists little work about the methods that firms and engineers use to elicit and analyze security requirements (see for example Tryfonas, Kiountouzis, & Poulymenakou (2001) and Elahi, Yu, Li, & Liu (2011)). Are practitioners using specific framework or methods for arriving at security requirements during the design and development cycle of IoT applications? Do the SRE methods and frameworks satisfy developers' needs, especially in the development of a technology prone to security issues such as the IoT? As Tondel et al. mentioned in their article: "no matter how academically correct formally pleasing a method might be, if the intended audiences are not using it, it is not good enough" (Tondel et al., 2008, p. 25). Therefore, the security research community needs to understand developers and their issues and strengths to handle security requirements in the development process. Field studies in real environments to understand the ways in which individuals and business deal with security in the process of technological development could help researchers to provide methodologies that closely meet developers' needs.

1.4 Research objective and questions

As mentioned in section 1.2, knowledge about how developers deal with the security requirements of IoT applications is still lacking. Security requirements engineering field could enormously contribute developers to handle the security requirements of this type of applications. However, for doing so, the security requirements engineering field need to understand better developers of IoT applications, their needs, and motivations to address security. Thus, developers' practices to address security during the development process should be brought in the field of security requirements engineering. By gaining knowledge of current practices, security researchers could provide methods and frameworks that better fit developers' necessities and cope with the challenges that developers are facing. In this thesis, we aim to address the gap between developers' practices to handle security requirements and security requirements literature. The objective of the theses is *to (better) understand the differences between security requirements engineering practices and developers' security practices in the context of IoT healthcare applications.* The healthcare domain is chosen because of the available information regarding this type of applications and its state of the art as trendy technologies with high social impact.

Before stating the research questions and sub questions to achieve the objective of the thesis, we briefly discuss why it is important to focus on security requirements.

1.4.1 Why focus on (security) requirements?

During the introduction to this thesis projects, it was mentioned that the IoT is prone to various cyberattacks. Since IoT should cope with cyber threats, security concerns need to be understood from the requirements, design, and testing to build-in security rather than batching security afterward (Ramachandran, 2015). Figure 2 illustrates a set of security activities that can be applied to various artifacts produced during the technology's development. Security activities apply within the three levels *the requirement level, the design and architecture level*, and the *code level* (Kocher et al., 2004). The requirements level is the starting point of any development process. In this level, security requirements must cover functional security i.e. security mechanisms as applied cryptography and emergent features (Kocher et al., 2004). Besides, NIST framework for Security Considerations in the System Development Life Cycle and Microsoft's

Security Development Lifecycle agrees on the importance of analyzing security during the first stages of development *the requirement level* (Jøsang, Ødegaard, & Oftedal, 2015).

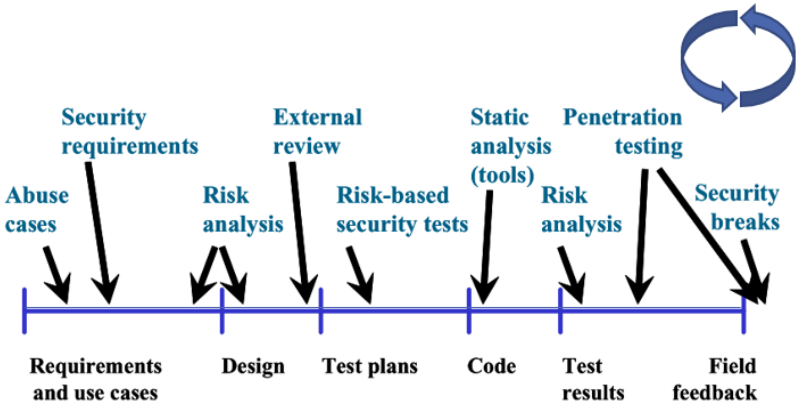

Figure 2: Security best practices applied to various software artifacts in the Software Design Life Cycle (Kocher et al., 2004). Cycle arrows indicate that activities can be cycled to follow iterative development approaches.

In addition, security practices to elicit and analyze security requirements early in the development lifecycle might bring economic benefits. Requirements are the cause of many issues during a project development (Firesmith, 2007). Mead & Stehney (2005) and Firesmith (2007) argue that poor requirements produce major cost and schedule overruns, canceled projects, and delivered inferior quality applications. The analysis of security requirements during the requirements phase of development might reduce future costs. Soo Hoo et al. (2001) have provided some evidence of the benefits. According to the authors, "findings indicate that significant cost saving and other advantage are achieved when security analysis and secure engineering practices are introduced early in the development cycle. The return on investment ranges from 12 percent to 21 percent, with the highest rate of return occurring when analysis is performed during application design" (Hoo, Sudbury, & Jaquith, 2001, p. 3).

1.4.2 Research questions

In order to achieve the research objective, and based on the research problem stated above, the following main research question is prepared:

How do developers' security practices in the design and development of IoT medical applications challenge security requirements engineering methods?

The main research question is divided in the following sub research questions. The first sub question will be answered through a literature review of scientific literature. This sub question aims to analyze the body of literature on security requirements engineering to arrive a set of recommended practices for handling security. Moreover, the collected information will serve as guidance for the study, especially the data collection.

1. Which security practices to handle security requirements are described in the literature of security requirements engineering?

After understanding the security practices posted by the literature, we need to get insights of how developers are handling security requirements in their real setting. An exploratory study will be

conducted to gather information regarding developers' security practices and the factors that influence such practices. The aim of sub question 2 is to gain insights on how manufacturers handle security requirements during de design and development of IoT medical applications. Sub question 3 will help us to understand the context in which IoT applications are being developed and the incentives that developers have for incorporating security requirements into the application.

2. What insights can we gather regarding how developers elicit, analyze, and manage security requirements during the IoT medical application development?
3. Which factors influence developers to accommodate security in the IoT medical application?

The last sub question aims to compare practice from the literature which was gathered in sub question 1 and developers' practices obtained from sub question 2. The information generated by this question and sub question 3 will help us to answer the main research question.

4. Which are the differences between SRE recommended practices and developers' security practices in the development of medical IoT applications?

1.5 Practical and scientific relevance

The results of this master thesis are relevant for both developers and academia. From a developers' perspective, this research can help companies to incorporate security requirements from the beginning of the design and development process, which in turn could produce more secure applications. Although providing recommendations is not the primary objective of this thesis, Chapter 2 presents a detailed set of practices that could be incorporated into the application' development process. Secure applications might reduce mitigation costs, such as patching in the future, because vulnerabilities could be envisioned in advance. Moreover, developers could acquire knowledge regarding the field of security requirements engineering and the different frameworks, tools and methods that this field promotes to develop secure products and software applications.

From an academy perspective, this thesis contributes to the field of security requirements engineering by providing insights on developers' perceptions and practices to manage security requirements in the development process. Security researchers could learn from the experiences of developers, and thus, understand better the complex process of technology development. Ideas and insight obtained from developers could help SRE field to provide frameworks and methodologies for handling security requirements that fit actual practices. Information from the industry is vital to develop targeted methodologies and to validate researchers' assumptions, which in turns, will result in more *relevant* and *useful* methods. The study might also identify new necessities and challenges that developers are facing, particularly in technologies such as the IoT, which may require immediate attention from the academia.

Furthermore, this research contributes to the growing demand of field studies in the area of security. The gathered insights regarding developer's techniques to elicit and analyze security requirements might provide useful information to the security field. This is because secure systems are not only the product of incorporating security technologies, it also requires analyzing

the root of the problems which might be in the technology development. Finally, this research can be considered as a pre-study in the field of security practices within companies developing IoT technologies. Researchers can learn from the decisions made during this study, especially in the research methodology. Future research could take different approaches to overcome our limitations.

1.6 Research approach

For achieving the objective of this research proposal (see section 1.4) and to provide an answer to sub question 1, first, an in-depth understanding of the security requirements engineering concepts and principles, and security requirement engineering methodologies and frameworks are needed. This knowledge will be acquired by means of an extensive literature review if the field of security requirements, security requirements engineering, and requirements engineering.

Next, in order to answer sub question 2 and sub question 3, an exploration of developers' security practices during the development of IoT medical applications, manufacturer's incentives for considering security during the design and development of these applications, and the tradeoffs among design requirements are required. The exploration is carried out using a *qualitative study*. A profound insight into the way that the design process takes place, and the reasons for choosing one option instead of another can be obtained by conducting interviews in combination with studying a different type of documentation, and with a detailed observation on location (Verschuren & Doorewaard, 2010). Therefore, a qualitative research method fits with the information needed for answering the main research question of the research project. The qualitative research is organized in four phases: design, data collection, analysis, and result presentation. Figure 3 shows the research approach, and Chapter 4 provides a detailed explanation.

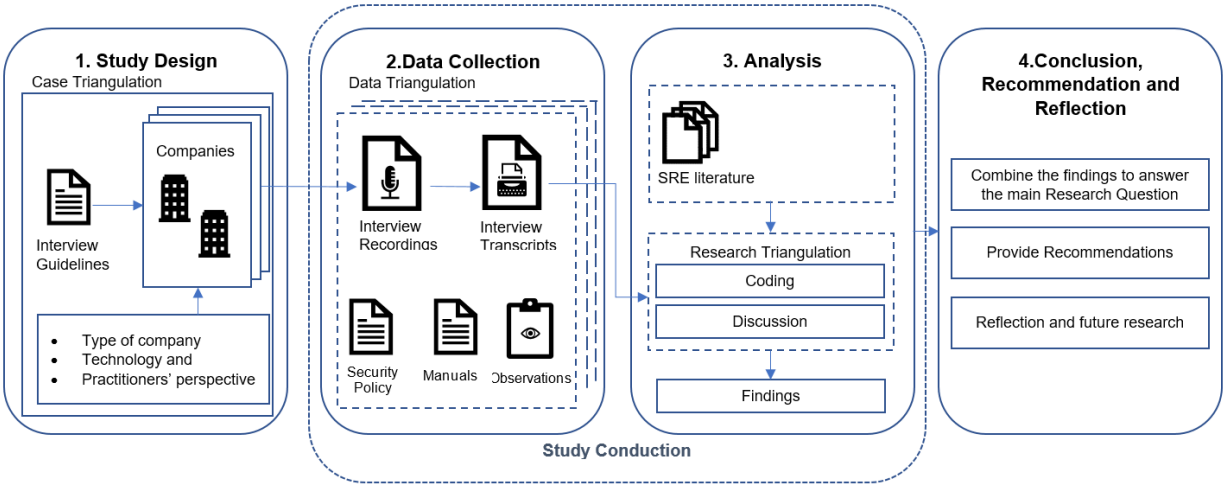

Figure 3: Research approach

The definition of the unit of analysis is related to the way in which the initial research question has been defined (Yin, 2014). For achieving the aim of this research, and due to its explorative nature, a qualitative study will be performed with IoT application developers as a unit of analysis.

1.6.1 Clarification of the term “practices”

Before starting the development of this master thesis, a clarification of the term practices is required. *Practices* in the context of this thesis refer to the set of actions and activities developers undertake to elicit, analyze, manage and verify security requirements of a system to be, and the practices to incorporate these security requirements into the system’s design. Therefore, the term practices refer to both the procedures or actions to handle security requirements and the procedures or actions to integrate the security requirements into the application.

1.7 Thesis outline

In Chapter 2 we provide a detailed set of security practices from the perspective of the security requirements engineering. This Chapters also introduces our interpretative framework for data collection and data analysis. Chapter 3 briefly introduces the main challenges of developing IoT medical applications. Chapter 4 explains the methodology employ for this research project. Based on the data obtained from three developers of IoT medical applications, Chapter 5 describes developers’ practices to handle security requirements during the development process. Chapter 6 gives a detailed explanation of the factor that influence developers’ practices. The factors were identified as part of the data analysis by following grounded theory principles. Chapter 7 compares developers’ practices to handle security with the practices recommended by the field of security requirements engineering. A discussion on the implications of our finding to the field of security requirements engineering and developers is provided in Chapter 8. Finally, Chapter 9 presents the conclusions of our research project, limitations of the research, and offer some recommendations.

2

Theoretical Background

Security experts from the academia argue that security must be considered at all stages of information systems development, being especially important during the early stages of design and development (Dubois & Mouratidis, 2010; Mead, Viswanathan, & Padmanabhan, 2008; Mellado et al., 2010; Ramachandran, 2015; Souag, Salinesi, Mazo, & Comyn-Wattiau, 2015). In this sense, *security requirements are conceived as the heart of developing secure systems* (Ramachandran, 2015). Based on the literature on security requirements engineering, section 2.1 combines frameworks, techniques, and security requirements methods to arrive a set of practices for handling security requirements during the design and development of information systems. Section 2.2 describes our interpretative framework to categorize developers' security practices within an organization. Finally, section 2.3 provides a preliminary conclusion.

2.1 Security Requirements Engineering

Security Requirements Engineering (SRE) is the "process of eliciting, specifying, and analyzing security requirements for a system" (Haley, Laney, Moffett, & Nuseibeh, 2006, p. 16). SRE provides techniques, methods, and norms for addressing security in the early phases of information system development cycle. Furthermore, the set of security requirements should be complete and understandable to the different stakeholders involved in the development process (Mellado et al., 2010).

Security requirements engineering emerged as the combination of requirement engineering and software security. Current requirements engineering techniques were found insufficient for representing security related requirements effectively (Ramachandran, 2015). Firesmith (2003) argue that requirements engineers derive, analyze, identify, and manage some non-functional or quality requirements (interoperability, availability, performance, reliability, usability). However, some engineers are lost when security requirements need to be elicited because of the lack of experience in security matters (Firesmith, 2003). Requirement engineering methods have only considered "what the system must do, but not what the system must not do" (Ramachandran, 2015, p. 316). Moreover, the engineering of requirements for business, software application or system, and components comprises more than just designing and building its functional requirements (Firesmith, 2003).

Requirements are something that the product must do to support its owner's business; it must make the product suitable, acceptable and attractive to customers/consumers (Robertson & Robertson, 2013). Requirements are classified as *functional requirements* which deal with the functionality of the systems related to the services that the system should provide (Mellado et al., 2010; Ramachandran, 2015). *Non-functional requirements* on the other hand are global requirements related to for example features of quality, performance, standards, regulation, interfaces, reliability, security, and other implementation requirements (Ramachandran, 2015). *Security* is considered as a non-functional requirement or quality requirement (Haley et al., 2008), and non-functional requirements tend to be neglected or added later on (Ahmed, Aung, & Svetinovic, 2013).

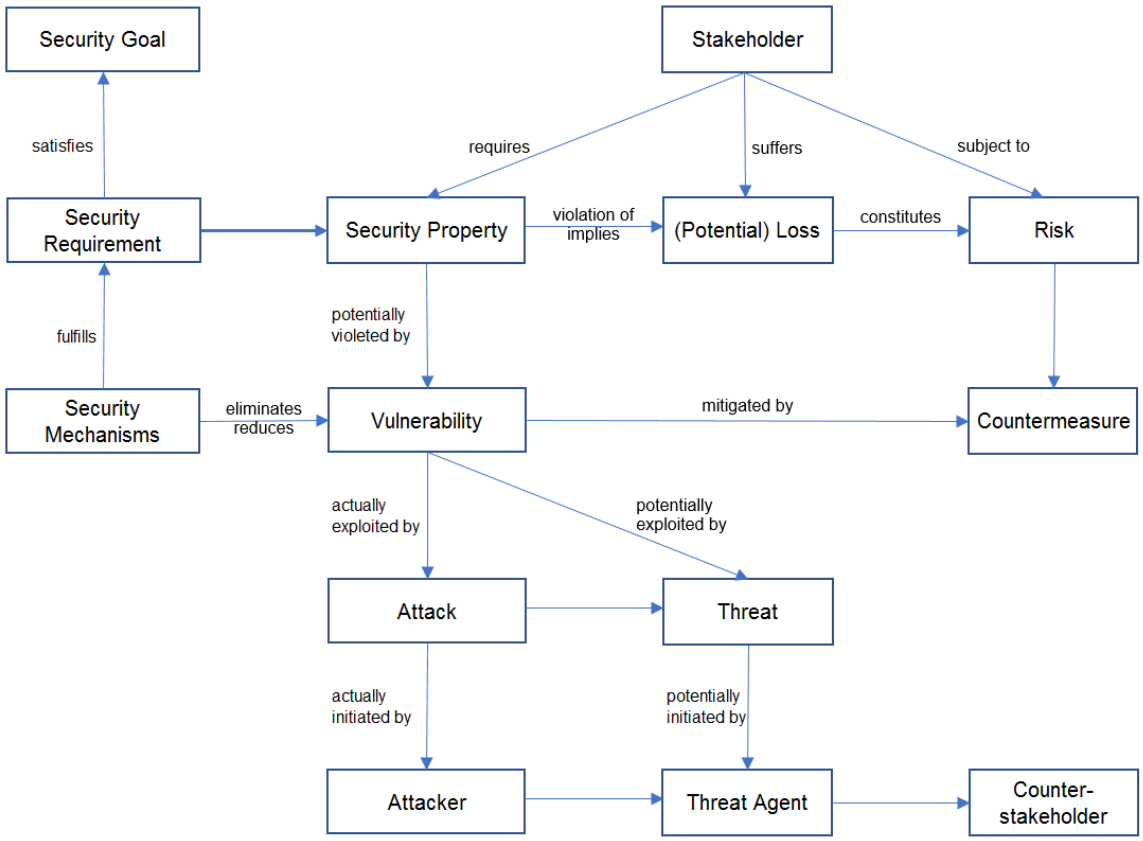

Figure 4: Relations among security concepts for security requirement engineering (adapted from Fabian et al. (2009) and Firesmith (2004)

Haley et al., (2006) argue that developers need to understand first the "what" of security requirements before the "how" of their construction and analysis. For doing so, we start defining *security* by using the definition proposed by Firesmith (2004). According to the author, *security* is "the degree to which malicious (i.e. unauthorized and intention) harm to valuable system assets is prevented, reduced, and properly respond to. *Security* is about protecting assets, which can be data, services, hardware, and personnel from harm due to various kinds of attacks (e.g. password sniffing, spoofing, virus) that may be performed by various types of attackers (e.g. hackers, international cyber terrorists, industrial spies, employees)" (Firesmith, 2004, p. 61). In this sense, *security requirements* are "quality requirements that specify a required amount of security in terms

of a system-specifies criterion and minimum level of an associated quality measure that is necessary to meet" (Firesmith, 2004, p. 63). However, to understand security requirements across the different stages of the development process, the definition proposed by Haley et al. (2008) has proved to be more useful (Dubois & Mouratidis, 2010). According to the authors, security requirements are constraints on the function of the systems – i.e. functional requirements – sufficient to protect the assets from identified harms (Haley et al., 2008). Figure 4 illustrates the security concepts that influence and are influced by security requirements. In the following section, we describe in more detail the different tasks that need to be carried out for the process of security requirement engineering. Related concepts illustrated Figure 4 are clarified along the section.

2.1.1 Security practices: from security requirements to security practices

For achieving the purpose of this thesis, we consider *security practices* as the set of actions and activities practitioners undertake to elicit, analyze, manage and verify security requirements of a system to be, and the practices to incorporate these security requirements into the system's design. Security requirements engineering comprise many techniques, process, and methods that aim to facilitate the management of security requirements in the development of information systems (Mellado et al., 2010). Therefore, security practices will be derived from security requirements methods.

The process of requirements engineering can be decomposed into four tasks: *elicitation, requirements analysis, validation and verification, and requirements management* (Cheng & Atlee, 2007). We use the same approach for referring to security requirements engineering. This section is based on the conceptual framework (CF) for security requirements engineering proposed by Fabian et al. (2009). This conceptual framework explicitly illustrates the interrelation among the different concepts and ideas used in security engineering (Fabian et al., 2009). Furthermore, we incorporate notions and ideas from: Security Quality Requirements Engineering (SQUARE) (Mead & Stehney, 2005), Multilateral Security Requirements Analysis (MSRA), Security Requirements Engineering Process (SREP) (Mellado, Fernandez-Medina, & Piattini, 2006), Security Requirements Engineering Framework (SREF) (Haley et al., 2008), and Security Requirements Engineering for Socio-Technical Systems (STS-Tool) (Paja, Dalpiaz, & Giorgini, 2014) to arrive a complete set of recommended practices.

Elicitation of security requirements

Requirements elicitation "comprises [actions] that enable the understanding of the goals, objectives, and motives for building an intended system (Cheng & Atlee, 2007, p. 287). In the case of security requirements, elicitation includes identifying the requirements that the resulting system must meet to successfully reach the system's *security goals* (Cheng & Atlee, 2007; Haley et al., 2008). Table 1 provides an overview of the practices for eliciting security requirements and the related SRE methodology in which they are mentioned.

Multilateral Security Requirements Analysis and STS-Tool have emphasized the need for identifying stakeholders involved in the project early in the elicitation phase (Fabian et al., 2009; Paja et al., 2014). Ramachandran (2015) pointed out that representing security requirements is a

joint effort which involves more actors than traditional requirements engineering. To identify and analyze security requirements, for instance, social engineers, security specialists, experts on modeling business process, and end users should also be considered (Ramachandran, 2015). Besides, a secure design is possible when knowledge about the system and knowledge of security practices are presented during the system's design (Flechais & Sasse, 2007). Therefore, to ensure that both pieces of knowledge are accessible, relevant actors in a system need to be distinguished and represented in the development process. Stakeholders should involve individuals who know better the system, for instance, potential users, manufacturers or managers; and individuals who are more aware of security practices, such as security experts (Flechais & Sasse, 2007).

Furthermore, achieving a common understanding of security concepts and principles is fundamental to any secure design. Approaches such as SQUARE and SREP have included "agree on definitions" as a first stage in the process of eliciting security requirements (Mead & Stehney, 2005; Mellado et al., 2006). This step aims to facilitate the communication among stakeholders, designers, and engineers (Fabian et al., 2009). Werlinger et al. (2009) argue that stakeholders have different understandings of security and risk, and do not have security as a prime concern. Then, security practitioners a have to adequately communicate security issues (Werlinger, Hawkey, & Beznosov, 2009). Therefore, an understanding of concepts and principles might reduce conflicts and confusion among stakeholders.

Although attempts to assign responsibility to different stakeholders or to ensure their motivation to manage security issues extend beyond the scope of any design methodology, Flechais & Sasse (2007) strongly recommend the clear assignment of responsibilities, especially who is in charge of security, as one of the initial steps of a security process. Besides, responsible stakeholders should have the authority to implement security decisions during the security requirements engineering process (Flechais & Sasse, 2007).

Security requirements operationalize one or more security goals (Fabian et al., 2009; Haley et al., 2008). A s*ecurity goal* is a very general statement about the security of an *asset.* In information systems, an *asset* is any data, device, or component that the owner places value upon (Fabian et al., 2009). Identifying or defining security goals is a central step toward the elicitation of security requirements. Methods such as SQUARE, MSRA , SREF, SREP have highlighted this aspect (Fabian et al., 2009; Haley et al., 2008; Mead & Stehney, 2005; Mellado et al., 2006).

Security goals describe the desire to protect valuable assets against harm by "describing the involved asset(s) and the harm to be prevented" (Haley et al., 2008, p. 134). Different applications have different assets to protect, and different assets will be subject to diverse kinds of threats (Firesmith, 2004). Therefore, valuable assets and threats to these assets need to be identified to define the security goals of a system-to-be (Firesmith, 2004; Haley et al., 2008; Mellado et al., 2006).

After determining the security goals of a system, requirements engineers should make concrete these stakeholders' security goals by translating the security goals into security requirements (Fabian et al., 2009). Techniques such as misuse cases, scenarios, and templates forms can help requirements engineers and other stakeholders to deeply reflect on security and be more accurate

about their security requirements (Mead & Stehney, 2005). Misuse cases represent "behavior not wanted in the system to be developed" (Tondel et al., 2008, p. 23). During the elicitation of requirements, however, misuse cases should be 'essential' misuse cases which do not comprise unnecessary architecture and design constraints (Firesmith, 2003). Scenarios are used as communication's tools that support security analysis. Scenarios serve as a way to communicate security notions, reflecting on security principles and explaining different visions (Flechais & Sasse, 2007).

Security requirements "refer to a specific piece of *information* or service that explains the meaning of an asset in the context of the system under construction" (Fabian et al., 2009, p. 15). Therefore, the initial set of security requirements which have operationalized the security goals should clearly specify the information, counter-stakeholder against whom the requirements are directed, and under which circumstances it must be satisfied (Fabian et al., 2009). For example, in a medical application, a customer's security requirement states that his/her medical history must not become known to arbitrary administrative employees or other patients. In this case, the information is the medical history, and the counter-stakeholders are administrative personnel and other patients. The circumstances described, e.g., that an authorized hospital worker who requires the patient's history for a particular purpose may nevertheless get to know it.

Task	**Activities**	**SRE Method**
Elicitation	▪ Identify relevant stakeholders for the system under construction, which includes security expertise.	MSRA, STS-Tool
	▪ Agree on definitions to achieve a common understanding of security concepts and principles.	SQUARE, SREP
	▪ Identify or define security goals by: (1) Identifying valuable assets (2) Identifying threats to these valuable assets	CF, SQUARE, SREF, MSRA, and SREP
	▪ Operationalize stakeholders' security goals into security requirements. Techniques such misuse cases, scenarios, and templates forms are useful to consider more deeply security requirements.	CF, SQUARE, SREP, MSRA
	▪ Ensure that the security requirements consider information, counter-stakeholders, and circumstances under which it must be satisfied.	CF
	▪ Ensure that the elicit security requirements are not an implementation or architectural constraints.	CF

Table 1: Practices for eliciting security requirements (adapted from SRE methods)

The elicited security requirements should not be an implementation or architectural limitations (Fabian et al., 2009). Firesmith (2003) argues that one of the most common problems, when security requirements are specified at all, is that security requirements tend to be substituted with security-specific design constraints which might restrict engineers from applying the most suitable security mechanisms to meet the underlying requirements (Firesmith, 2003). Engineers should

specify what is required from the system, for instance, a specific level of identification and authentication, rather than the architectural mechanism by which it must be accomplished, such as user ID and password (Firesmith, 2004).

Analysis of security requirements

Requirements analysis includes "techniques to better understand requirements, their interrelationships and their potential consequences, thus, more informed decisions could be made" (Cheng & Atlee, 2007, p. 289). Table 2 provides an overview of the practices for analyzing security requirements and the related SRE methodology in which they are mentioned.

From the elicited security requirements, a set of *security system requirements* needs to be derived. Fabian et al. (2009) argue that security system requirements are the result of the reconciliation of stakeholders' view for security requirements. Different requirements are interdependent and communicate with each other. Interactions among requirements could be positive or negative (conflicts). Affirmative interactions might be beneficial in prioritizing requirements or redundancies in requirement analysis. And inconsistent or conflicting requirements might be the initial point for obtaining valuable insights that might otherwise be ignored (Fabian et al., 2009).

Conflicting security requirements might appear because stakeholders have different and most of the time contradicting, security concerns (goals) about an asset. Besides, conflicts might arise for different reasons at different stages of requirements engineering (Fabian et al., 2009). According to Holmström & Sawyer (2011), discussing conflicts among stakeholders is a common, if not unavoidable, part of requirements engineering activity. Thus, collecting, describing, and prioritizing requirements includes engaging in negotiations (Holmström & Sawyer, 2011). An efficient negotiation process may occur when main stakeholders confront each other and discuss their points of view. This conflict resolution or negotiation process might also be useful to arrive a shared understanding of the requirements that are essential for the system (Holmström & Sawyer, 2011).

Methods such as SQUARE and SREP have categorized and prioritized security requirements as part of the analysis process. According to SQUARE, the elicited security requirements should be categorized according to the following criteria: essential, non-essential and system level. Furthermore, it is believed that not all requirements can be fulfilled. Therefore, the most critical requirements must be identified by performing a risk assessment of categorized security requirements. As a result, security requirements are prioritized by the stakeholders. Finally, in the latest step, requirements are check for ambiguities, discrepancies, inaccurate assumptions and the like (Mead & Stehney, 2005). The reviewed security requirements are documented; this document is designed to satisfy the security goals of the organization (Fabian et al., 2009; Mead & Stehney, 2005). SREP also rank security requirements according to the impact and the likelihood of the threats to harm a valuable asset, that is according to the risk (Mellado et al., 2006). In the fifth activity in SREP, Mellado et al. perform a risk assessment to determine the likelihood of each threat and to assess its impact and risk (Mellado et al., 2006). Therefore, risk information is used to categorize and prioritize security requirements.

Finally, after the analysis process, security requirements have to be documented. As it has been described during this section, security requirements focus on *what* should be achieved rather than *how* it should be achieved (Tondel et al., 2008).

Task	Activities	SRE Method
Analysis	▪ Derive a set of security system requirements by: (1) Analyzing and managing security requirements interactions. (2) Reconciling stakeholders' views for SR or confliction SR. (3) Engaging in negotiations to solve conflicts (conflict resolution).	CF, MSRA
	▪ Derive a set of categorized and prioritized security requirements by: (1) Categorizing SR according to essential requirements, non-essential requirements, system level requirements. (2) Perform a risk analysis of categorized security requirements (3) Prioritized categorized security requirements, it could be done based on the risk analysis results. (4) Check for ambiguities, inconsistencies, mistaken assumptions	SQUARE, SREP
	▪ Document security requirements for stakeholders.	CF, SQUARE, SREP

Table 2: Practices for analyzing security requirements (adapted from SRE methods)

Management of security requirements

Requirements management is an "umbrella activity which contains many duties related to the management of requirements, including the evolution of the needs over time" (Cheng & Atlee, 2007, p. 290). In this area, we will also consider the integration of security requirements into the design of a system. Table 3 provides an overview of the practices for managing security requirements and the related SRE methodology in which they are mentioned.

Security system requirements have to be incorporated in the system requirements. System requirements define attributes the system must have after the artifact or system has been developed (Fabian et al., 2009). Fabian et al. (2009) argue that interacting functional requirements, non-functional requirements, and security system requirements need to be reconciled to arrive at a consistent *set of system requirements*. As explained in the analysis section, conflicting system requirements should be addressed by engaging in negotiations. In the situation of opposing functional and security requirements, MSRA suggests exchanging functionality for security and vice versa (Fabian et al., 2009).

The consolidation of system requirements, which integrate functional requirements, security system requirements and non-functional requirements, is a fundamental step in requirements engineering because security requirements constraint functional requirements to satisfy

applicable security goals (Haley et al., 2008). Therefore, by accommodating those three types of requirements, security will be a property that the systems must have.

The consistent set of system requirements is redefined into a system's specifications as well as *facts* and *assumptions* about the environment where the system will operate. This set of properties (specifications, facts, and assumptions) "must be sufficient to satisfy the system requirements" (Fabian et al., 2009, p. 13). *Assumptions* are an important part of the design process of information systems. Verschuren & Hartog (2005) argue that designers should not only design the artifact in such a way that it satisfies the wishes of potential users and requests coming from the context. Also, they should establish what properties are needed from end users, and the context of operation to make a productive use possible (Verschuren & Hartog, 2005).

Specifications constrain the system to be produced, while facts and assumptions describe (or constrain) the environment of the system (Fabian et al., 2009). An artifact cannot satisfy or enforce security requirements unconditionally on its own; it can, however, implement security mechanisms that contribute to systems security (Fabian et al., 2009). Security mechanisms, also known as countermeasures, are architectural mechanisms that help to satisfy one or more security requirements, and thus, to reduce security vulnerabilities (Firesmith, 2004).

At the design level, the design of the system together with the additional facts and assumptions will refine the specifications (Fabian et al., 2009). Meanwhile, at the implementation level, assumptions are refined to organization procedures and processes that guide users on how to employ the implemented artifact to reach security (Fabian et al., 2009).

Task	**Activities**	**SRE Method**
Management	▪ Arrive at a consistent set of system requirements by: (1) Reconciling interacting functional requirements, non-functional requirements, and security requirements. (2) Engaging in negotiations to resolve conflicts.	CF, MSRA
	▪ Redefine system requirements into system specifications, facts, and assumptions about the environment.	CF
	▪ Architect security mechanism to fulfill security requirements.	CF, SREF
	▪ At the design level, ensure the satisfaction of security requirements.	CF
	▪ At the implementation level, refine assumptions into organization procedures and processes that guide users on how to employ the artifact to reach security.	CF

Table 3: Practices for managing security requirements (adapted from SRE methods)

Validation and verification

Validation and verification of requirements are needed at the different stages of the security requirements engineering process. Table 4 provides an overview of the practices for validating

and verifying security requirements and the related SRE methodology in which they are mentioned.

Requirements validation ensures that models and documents accurately express the stakeholders' needs (Cheng & Atlee, 2007). Once the security requirements have been developed or identified, requirements engineers must validate that the security requirement satisfies the intended security goal (Haley et al., 2008). Besides, validation of requirements for completeness, lack of conflicts, or others matters is needed alongside the elicitation, analysis, and management of security requirements (Tondel et al., 2008). Conflicts or conflicting requirements might appear in any stages of the security requirement engineering process, not only in the analysis phase. Thus, requirements engineers must recognize, address, and resolve conflicts among stakeholders; but also, examine the underlying structure to understand the source of these conflicts (Holmström & Sawyer, 2011).

Verification techniques can be used to prove that system specification meets the security requirements. Verification can be proven by checking that a system satisfies some constraints (Cheng & Atlee, 2007). For instance, security mechanisms need to be verified to make sure that the mechanism indeed fulfills the security requirement.

Task	**Activities**	**SRE Method**
Validation and Verification	▪ Validate that the security requirement satisfies the intended security goal.	SREF
	▪ Validate requirements for completeness, lack of conflicts, or others matters.	SQUARE
	▪ Verified that the security mechanism indeed fulfills the security requirement.	

Table 4: Practices for validating and verifying security requirements (adapted from SRE methods)

2.1.2 Summary of security practices from SRE methods

During section 2.1.1 practices for eliciting, analyzing, managing, and verifying security requirements according to the security requirement engineering literature were extensively discussed. In this section, we aim to summarize the main practices. Table 5 gives an overview of the security practices to elicit, analyze, manage, and verify security requirements.

The **elicitation of security requirements** involves identifying the requirements that the system must satisfy to achieve the system's security goal. During requirements identification meetings, practitioners should *include stakeholders* who bring knowledge about the system and knowledge of security concepts and principles. Then, a clear communication among stakeholders should be enabled by *agreeing on definitions regarding security*. After that, *security goal* should be identified or defined. Security goals describe stakeholders' desire to protect valuable assets against harm. Thus, *valuable assets and threats* to these valuable assets need to be determined. After establishing security goals of a system, detailed *security requirements are elicited* to operationalize the security goal. Security requirements should include a specific piece of information that explains the meaning of an asset (e.g. medical history), the counter-stakeholder against who the requirements are directed (e.g. arbitrary employee), and circumstances under

which the requirements must be satisfied (e.g. authorized employee who needs to know the patient's history have access to the information). Finally, developers should *ensure that the elicited security requirements are not architectural constraints*. Requirements specify what is needed from the system (e.g. specific level of identification and authentication) rather than the architectural mechanism (e.g. user ID and password).

The **analysis of security requirements** allows practitioners to better understand requirements, their interrelationships, and their potential consequences. From the elicited security requirements, a set of *Security System Requirements needs to be derived*. For doing so, practitioners should analyze and manage security requirements interactions, *reconcile stakeholders' view regarding security requirements*, and *engage in negotiations to solve conflicts*. Conflicting security requirements appears because stakeholders have different, and most of the time contradicting, security concerns (goals) about an asset. Thus, managing conflicting requirements is a common part of requirements engineering activity. Following that, a *categorized set of security requirements* should be derived. Practitioners might categorize security requirements according to essential requirements, non-essential requirements, and system level requirements. Then, a *risk analysis* should be performed to identify the most important requirements. For performing the risk analysis, the impact and the likelihood of threats to harm a valuable asset are examined. When the set of categorized security system requirements has been obtained, practitioners need to check for ambiguities, inconsistencies, and mistaken assumptions. Lastly, *security requirements are documented*.

The **management of requirements** comprises various tasks related to the handling of requirements, including the evolution of requirements over time. Categorized security requirements should be *integrated with functional requirements and non-functional requirements* to arrive a consistent set of System Requirements. *Conflicts between functional requirements and security requirements* should be solved by engaging in negotiations. Security requirements will constrain functional requirements to satisfy applicable security goals. After that, the consistent set of system requirements is redefined into a systems' specifications, facts, and assumptions about the environment where the system will operate. Specifications constrain the system to be built, while facts and assumptions describe (or constrain) the environment of the system. Then, *security mechanisms are architected* to fulfill one or more security requirements, and thus, to reduce one or more security vulnerabilities. Finally, at the design level, practitioners should ensure the satisfaction of security requirements. Besides, at the implementation level, assumptions are refined into organizational procedures which describe how the system must be employed to achieve security.

The **requirements validation and verification** ensures that models and documents accurately express the stakeholders' needs. Practitioners must *validate that the security requirement* satisfies the intended security goal. Validation of requirements for completeness, lack of conflicts, or other matters is needed alongside the elicitation, analysis, and management of security requirements. Verification techniques are used to prove that *system specification meets the security requirements*.

Task	**Activities**
Elicitation	▪ Identify relevant stakeholders for the system under construction, which includes security expertise. ▪ Agree on definitions to achieve a common understanding of security concepts and principles. ▪ Identify or define security goals by: (1) Identifying valuable assets (2) Identifying threats to these valuable assets ▪ Operationalize stakeholders' security goals into security requirements. Techniques such misuse cases, scenarios, and templates forms are useful to consider more deeply security requirements. ▪ Ensure that the security requirements consider information, counter-stakeholders, and circumstances under which it must be satisfied. ▪ Ensure that the elicit security requirements are not an implementation or architectural constraints.
Analysis	▪ Derive a set of security system requirements by: (1) Analyzing and managing security requirements interactions. (2) Reconciling stakeholders' views for SR or confliction SR. (3) Engaging in negotiations to solve conflicts (conflict resolution). ▪ Derive a set of categorized and prioritized security requirements by: (1) Categorizing SR according to essential requirements, non-essential requirements, system level requirements. (2) Perform a risk analysis of categorized security requirements (3) Prioritized categorized security requirements, it could be done based on the risk analysis results. (4) Check for ambiguities, inconsistencies, mistaken assumptions ▪ Document security requirements for stakeholders.
Management	▪ Arrive at a consistent set of system requirements by: (1) Reconciling interacting functional requirements, non-functional requirements, and security requirements. (2) Engaging in negotiations to resolve conflicts. ▪ Redefine system requirements into system specifications, facts, and assumptions about the environment. ▪ Architect security mechanism to fulfill security requirements. ▪ At the design level, ensure the satisfaction of security requirements.

	▪ At the implementation level, refine assumptions into organization procedures and processes that guide users on how to employ the artifact to reach security.
Validation and Verification	▪ Validate that the security requirement satisfies the intended security goal. ▪ Validate requirements for completeness, lack of conflicts, or others matters. ▪ Verified that the security mechanism indeed fulfills the security requirement.

Table 5: Security practices to elicit, analyze, manage and verify security requirements (adapted from SRE methods).

During this section, we have referred to practices for managing security requirements from the lens of the security requirements engineering. Nevertheless, the process of building or developing an IoT application occur in an organizational context, i.e. within companies. For achieving the purpose of this research project, developer's security practices will be compared with SRE recommended practices. To do so, we need to describe both developer's security practices and SRE recommended practices on a common basis. An interpretative framework will allow us to organize developer's practices during the data collection, and to examine the differences during the data analysis. In the following section, an interpretative framework will be developed to describe developers' security practices within companies.

2.2 Security practices within an organization: an interpretative framework

The process of building or developing information system artifacts, such as IoT applications, occurs in an organizational context, i.e. within companies. This means that applications' development is an *organization activity*, where the context, the firm's environment, the human actors with its complexities are as well as essential for the success of development as the transformation of hardware components and software into a desired artifact (Päivärinta & Smolander, 2015). The influence of organizational arranges in security practices has also been acknowledged by the security field. For instance, Werlinger et al., (2009) presented a list of human, organizational, and technical challenges that security experts face within their organizations. The authors define organizational aspects as those related to the structure of the organization which includes firms' size and managerial decisions (Werlinger et al., 2009).

Development efforts take place in a development context (Päivärinta & Smolander, 2015) which is part of the organization activities. Generally speaking, organization activities can be divided into two main levels *managerial* and *operational*. Security practices during applications' development within companies can also be referred to these two levels. Tryfonas et al. (2001) followed a similar approach to address and combine security issues with development practices and their involved stakeholders. The authors categorized "popular mainstream and old security practices" by using three levels of abstraction within organizations: strategic, tactical, and operational level (Tryfonas, Kiountouzis, & Poulymenakou, 2001, p. 187). Although their categorization will produce data to gain a depth understanding of the role management on security practices; in this study, we will refer to the security development practices in two levels *managerial* and *operational*. The aim of

using these two layers is to distinguish activities which are required during the development process and managerial decisions that might shape a secure development process.

- The *managerial level* refers to a set of managerial decisions and actions that determine the long-run performance of a company (Schilling, 2005). It involves creating security policies, dealing with people issues, and evaluating threat and risks (White, 2009).

- *Operational level* answer the question "what security procedures and practices are to be utilized" (White, 2009, p. 72). In this layer, security practices are the ones concerning the practices adopted to perform a development.

According to Tryfonas et al., (2001), in the strategic and tactical level security practices are concerned with establishing and using security policies, complying with standards, conducting threat or risk analysis according to the risk appetite of the company, and protecting sensitive information (Tryfonas et al., 2001). For this study, the *managerial level* includes Tryfonas et al., strategic and tactical level of abstraction, and thus, we can assume that these security practices are the corresponding security practices of our managerial level. Besides, we can include Flechais & Sasse (2007) recommendation of assigning responsibilities to address security during the product design because this is a managerial decision that motivates developers to consider security as part of their duties.

In the operation level, Tryfonas et al., (2001) consider applied cryptography solutions, network security and use of firewalls, access control mechanisms, software security practices, and intrusion detection techniques as the corresponding security practices of this layer. Although security technologies or mechanisms are essential in a system, a thoughtful and reasoned utilization of these technologies and mechanisms is even more relevant if developers aim to produce secure artifacts. During section 2.1 we have argued that security can not be achieved by just incorporating security mechanism into the application but rather through the analysis of the security requirements that are needed to satisfy the system's security goals. In this sense, Kocher et al. (2004) pointed out that security issues are likely to appear because of a problem in a standards part of the system instead of in some given security mechanism. According to the authors, "just as you cannot test quality into a piece of software, you cannot spray paint security features onto a design and expect it to be secure" (Kocher et al., 2004, p. 755). Firesmith (2003) has claimed that the elicitation of security-specific architectural constraints limits the use of appropriate security mechanisms which are needed to meet security requirements. In addition, as mentioned above, practices in the operational level respond to the question what security procedures and practices are to be utilized for a secure development. Therefore, security practices in this level are the ones concerning the procedures, actions, and activities adopted by developers to perform a secure development. Following this idea, and considering the literature of SRE, we could assume that practices to achieve a secure product development include the procedures, actions, and activities *to elicit security requirements, analysis security requirements and their interactions with another type of requirements, manage security requirements throughout the development lifecycle, validate that security requirements meet security goals, and verify that security mechanisms fulfill security requirements*. Table 6 provides an overview of the security practices within an organization.

Level within organization	Corresponding security practice
Managerial	Establishing and using security policies Complying with security standards Conducting a threat or risk analysis Copyright protection Assigning responsibilities to address security
Operational	Eliciting security requirements Analyzing security requirements and the interaction with another requirement. Managing security requirements through the development lifecycle Validating security requirements Verifying security mechanisms

Table 6: Security practices within an organization (adapted from Tryfonas et al. and SRE methods)

Table 6 serves as the basis for our interpretative framework to address security practices within companies. We will use this framework to organize the data collection and data analysis of security practices in companies developing IoT medical applications. Nevertheless, our framework could be used in any information systems' development process. Table 7 illustrates the interpretative framework. Development activities within organizations are divided into two levels *managerial* and *operational*. The *managerial* level is concerned with the managers' decisions and actions that might shape a secure product development. The corresponding security practices include any high-level decision or action that influence in the secure development of medical IoT applications.

The *operational* level is concerned with the procedures, actions, or activities followed by developers to perform and achieve a secure development of the medical IoT application. In this layer, the corresponding security practices include four of the main tasks proposed by the security requirements engineering field: *elicitation, analysis, management, validation and verification* of security requirements. For defining these tasks, we utilize Cheng & Atlee definitions in the context of security. Besides, we aim that our framework could capture most of the developers' practices to manage security requirements in the development process.

Level within an organization	Corresponding security practice
Managerial (concerning the decisions and actions that might shape a secure product development)	*High-level decisions and actions*
Operational (concerning the procedures and actions adopted to perform a secure product development)	***Elicitation*** involves identifying the requirements that the resulting system must satisfy in order to achieve the system security goals. ***Analysis*** involves understanding requirements, their inter-relationships, and their potential consequences. ***Management*** comprise several activities related to the management of requirements, including the evolution of requirements over time and the integration of requirements in the system's design. ***Validation & Verification*** involves verifying that security requirements and security mechanisms meet security goals.

Table 7: Descriptive framework for categorizing and analyzing developers' security practices.

2.3 Preliminary conclusion

This chapter includes two sections. Section 2.1 provides a list of recommended practices to handle security requirements during the development process, and thus, this section answer sub question 1. Based on a literature review of security requirements engineering frameworks and methods, practices to elicit, analyze, manage, validate and verify were derived. These practices include the *operationalization of security requirements from security goals*; the *analysis of security requirements and their interrelations with functional and non-functional requirements, and the prioritization of security requirement*; the *management of security requirements during the design and implementation*; and the *validation of security requirements and verification of security mechanisms*. The framework differentiates *managerial* actions which might motivate a secure development and *operational* procedures and actions to consider security requirements early in the design and development process. The framework is employed during the data collection of developer's practices and data analysis.

3

Challenges in the design of healthcare IoT

Applications are at the top of any IoT architecture. These are able to export all the systems functionalities to the final user (Atzori et al., 2010). In this section, we provide a brief introduction of IoT healthcare applications and the challenges during the design of these type of applications. The aim of this section is to gain an understanding of the complexity of medical applications. This chapter starts explaining IoT healthcare applications in section 3.1. Next, section 3.2 addresses the main challenges indicated by the literature of medical IoT and provide some information to deal with these. Finally, a preliminary conclusion is provided in section 3.3.

3.1 IoT Healthcare applications

Healthcare Internet of Things envisions a dramatic transformation in real time patient care due to the growing availability and connectivity among sensors, medical devices, hospitals' information systems and mobile technology (Wortman, Tehranipoor, Karimian, & Chandy, 2017). IoT application in the healthcare sector will enhance current assisted living solutions and reduce delay for treatment of critical patients (Miorandi et al., 2012). IoT healthcare applications refer to "monitoring solutions to support wellness, prevention, diagnostics and treatment" (Mazhelis et al., 2013, p. 19). By incorporating IoT in current healthcare systems, for instance, Physiological parameters such as blood pressure, heart rate, cholesterol level, respiratory activity, etc. can be collected and monitored real time because patients will carry medical sensors. Both wearable sensors such as gyroscopes or accelerometers and fixed sensors such as proximity will gather information for monitoring patients' activities in their natural habitat. Data obtained by the sensors will be aggregated and sent to remote healthcare institutions. In these institutions, advanced remote monitoring will be performed to provide rapid response actions when required (Miorandi et al., 2012). Moreover, personalized healthcare and well-being solutions are another relevant application for the sector. Daily activities such as steps walked, exercises completed and calories burned can be tracked by using wearable sensors and mobile applications (Miorandi et al., 2012). Thus, interventions by medical staff will be upon discovery of diseases which might lead to health deterioration (Miorandi et al., 2012).

According to MarketResearch.com, the healthcare Internet of Things market will reach $117 billion by 2020 (McCue, 2015). A network of connected identifiable medical devices to the internet and hospitals' information systems results in the generation of vast volumes of data which need to be stored, processed and displayed in a seamless, efficient, and easily interpretable form

(Gubbi et al., 2013). Thus, from a business perspective, the healthcare IoT presents various opportunities for firms such as devices developers, software developers, telecom operators, application and service providers, platform providers, integrators, and medical personnel to create a profitable IoT business ecosystem (Kim & Lee, 2014; Mazhelis et al., 2013).

IoT applications could provide many benefits to the healthcare sector and the global economy. However, to succeed in the development and implementation of services based on IoT, security and privacy need to be ensured (Abie & Balasingham, 2012). Insulin pumps, pace maker, inhalers, cardioverter defibrillator, etc. with wireless capabilities could enhance patients' care by providing treatments which respond to patients' conditions. Nevertheless, the connection of these devices to the internet and hospitals' information systems also create an entire attack surface which might be exploited to threaten patient's life (Wortman et al., 2017). Therefore, as well as another type of IoT applications, medical IoT application face the same security issues presented in section 1.1.2. But, one of the difference is that due to the nature of the medical field, failures in healthcare IoT applications' security will be a cause of life or death (Wortman et al., 2017).

3.2 Challenges of secure medical IoT design

During the design and development of IoT medical applications, there are many security challenges which are interconnected. Williams & McCauley (2017) found that these challenges refer to the regulatory environment of healthcare, diversity of devices, trusted and dynamic connectivity, the technical debt of engineering devices, patients' safety, and privacy (Williams & McCauley, 2017). Besides, Wortman et al. (2017) have also considered the cost of materials and production activities as one of the constraints during the IoT development. Cost considerations have an impact on the amount of resources that are employed in the design and development of a system. Cost decisions are often made at the expenses of security, and IoT medical devices are not the exception (Wortman et al., 2017). Another challenge is to prepare the healthcare industry for providing secure devices and applications. It is argued that the care industry is not prepared for coping with the security issues of medical IoT. According to a statement issued by the FBI, compared to other industries such as financial or retail sector, the "healthcare sector is not technically prepared to combat against cyber criminals' basic cyber intrusion tactics, techniques and procedures, much less against more advanced persistent threats" (FBI Cyber Division, 2014, p. 1). Thus, there is a likelihood of increasing cyber intrusions against health care systems (FBI Cyber Division, 2014). Finally, Mittelstadt (2017) argue that IoT applications in healthcare should "be designed to be technologically robust and scientifically reliable, while also remaining ethically responsible, respectful of users' rights and interest, and trustworthy" (Mittelstadt, 2017, p. 2). According to the author, due to the scale, scope and complexity of a medical system that creates, gathers, and examines personal health data, users (which are in vulnerable position as patients) face the impossible task to maintain control over their data (Mittelstadt, 2017).

Although all the challenges mentioned above difficult the development of medical IoT applications, we will briefly discuss *technical debt of engineering devices*, ***patient's safety and privacy***, and *ethical implication during the design of IoT medical applications* because these issues are closely related with the focus of this thesis project.

3.2.1 Technical debt of devices engineering

The following example used by Brown et al. (2010) gives an illustration of this concept. “The idea is that developers sometimes accept compromises in a system in one aspect (e.g., modularity) to meet an urgent demand in some other aspect (e.g., a deadline), and that such compromises incur a “debt” on which “interest” has to be paid and which the “principal” should be repaid at some point for the long-term health of the project” (Brown et al., 2010). Developers will incur a “debt” due to the lack of consideration of cyber security threats during the application design. They will pay an “interest” continuously during the development. However, at some point, they will have to repay the “principal” (Williams & McCauley, 2017) by integrating security technologies into the application to safeguard the medical device against potential threats. The cost of integrating security technologies at this stage will be higher than if the cyber security threats would have been analyzed early in the development process. Therefore, during the design and development of the IoT application, developers must consider security threats and vulnerabilities for the medical application. Based on a service architecture, Table 8 provides a summary of the security threats and vulnerabilities that can affect IoT medical systems. Although an in-depth analysis is required according to the applications’ characteristics, the provided information gives an idea of the main threats.

Sensing layer an IoT end-nodes	Network layer	Service layer	Application layer
Unauthorized access: sensitive information is captured	Data breach: release of private information to an untrusted environment.	Privacy threats: malicious location tracking or privacy leakage	Remote configuration: fail to configure at interfaces
Availability: end-node stops working due to physically attacked	Public key and private key: compromise of keys in network	Service abuse: unauthorized users access to services / unsubscribed services	Misconfiguration: at remote devices, nodes, gateway
Spoofing attack: attacker masquerades end-node/gateway by falsifying data		Identity masquerade: device, node, gateway is masqueraded by attacker	Security management: log and key leakage
Selfish threat: nodes stop working to safe resources due to network’ failure		Service information manipulation: service’ information is manipulated	Management system: failure to manage the system
Malicious code: Virus, Trojan, and junk messages	Malicious code	Repudiation: denial the required operations	
Denial of services (DoS): make node resource unavailable	Denial of services	Denial of services	
Transmission threat: interrupting blocking, data manipulation, forgery	Transmission threat	Transmission threat	
Routing attack: attack on a routing path	Routing attack	Routing attack	

Table 8: Security threats and vulnerabilities of IoT per layer (Li et al., 2016)

IoT medical applications will share information among all four layers. Thus, cross-layers threats need also to be analyzed. Sensitive information which might not be protected at the border of layers could be leaked. Identities in different layers might be spoofed. Sensitive information is spread at different layers and cause information leakage (Li et al., 2016).

3.2.2 Safety and privacy of patients: value consideration

Patients' safety and privacy protection are two fundamental values for the care sector. Broadly speaking, values can be defined as "what a person or group of people consider important in life" (Friedman, Kahn Jr., & Borning, 2006, p. 2). Values in IoT medical technologies were identified by (Detweiler & Hindriks, 2015). By following a Value Sensitive Design approach, the authors categorized fundamental values in medical IoT. Physical well-being, social well-being, freedom, privacy, responsibility, and safety are recognized as relevant values for users. Detweiler & Hindriks (2015) argue that by understanding the relationships between values, technologies, and medical IoT context of use; designers could better consider the values that the technology affect. Thus, more social desired medical applications, i.e. devices that take into account societal values, could be launch into the market.

Safety refers to protecting of patients of unintentional harm. Failures in security put in danger patients because malfunctions can harm the people's well-being (Williams & McCauley, 2017). Moreover, there is not widely accepted a definition of privacy because the same word denote different things for different people. Nevertheless, in the healthcare domain, privacy refers to the question who has access to personal information and under what conditions. Privacy is "concerned with the collection, storage, and use of personal information, and examines whether data can be collected in the first place, as well as the justifications, if any, under which data collected for one purpose can be used for another (secondary) purpose" (Nass, Levit, & Gostin, 2009, p. 76). Privacy is considered as a human right which has a value by itself. Besides, privacy promotes other fundamental values such as autonomy, individuality, respect, and dignity (van de Poel & Royakkers, 2011). IoT devices gather personal identifiable information which in the case of medical applications include health information. Privacy concerns appear because data shared among mobile applications and cloud services connected to smart devices could potentially be accessed by unauthorized actors due to failures in security (Williams & McCauley, 2017).

Developers might not even notice, but technology, and thus design decision, affect the values mentioned above by offering or performing functions that support some actions, and not others (Detweiler). For instance, a fall detection system could enhance (positively affect) patients' *safety* by alarming care personnel when patients fall. However, the same system could diminish (negatively affect) patient's privacy if designers decide to employ cameras, and record imagines, to provide the intended functionality. Developers play an important role on how technology affects values because their decision shapes the functionalities that a device will offer. Value considerations are not part of design approaches. However, understanding of which values technology affect could help developers to produce products that respect people's fundamental values, and thus, products which are accepted by patients, users, medical personnel, and society.

3.2.3 Ethical implication during medical IoT design

Laudon & Laudon describe four key technological trends that raise ethical issues in the case of information technologies. Computing power doubles every 18 months, rapidly decline of data storage's cost, advances in data analysis, advances in networking, growing impact of mobile technologies (Laudon & Laudon, 2014). As we can notice, the internet of things leverages these trends. Increasing computing power allowed the creation of small devices with sensing and processing capabilities. The decline of the cost of data storage allows using cloud technologies for storage and processing. Advances in networking allow the interconnection of small smart devices. Finally, mobile technologies are used as gateways to integrate smart devices to the network. Thus, IoT technologies raise ethical issues that might be difficult to address during the technological development.

Besides, as mentioned by Detweiler & Hindriks (2015), medical IoT technology has ethical implications because these technologies can contribute supporting human values, for instances the well-being of the population. However, these technologies can also hinder fundamental human values which bring less desirable outcomes to society. For example, medical IoT technologies might fail to secure sensitive health-related data violating patients privacy. In the previous section, we briefly discussed the role of values in the technology development. Then, what should be noticed is that ethical implications and moral values are deeply intertwined. In the remainder of this section, we list Mittelstadt's guidelines to assist developers in addressing ethical challenges with the healthcare Internet of Things in the real world. According to Mittelstadt (2017), an ethical design of health IoT will help developers, medical personnel, and care services to meet their moral responsibilities in supplying healthcare.

- Give users control over data collection and transmission;
- Iteratively adhere to industry and research confidentiality standards;
- Design devices and data sharing protocols to protect user privacy by default;
- Use alternative consent mechanisms when sharing H-IoT data;
- Meet professional duties of care and facilitate inclusion of medical professionals in H-IoT mediated care;
- Include robust transparency mechanisms in H-IoT data protocols to grant users oversight over their data;
- Report the uncertain utility of H-IoT data to users at the point of adoption;
- Provide users with practically useful mechanisms to exercise meaningful data access rights;
- Design devices to be unobtrusive according to the needs of specific user groups.

3.3 Preliminary conclusion

Healthcare Internet of Things applications refers to monitoring solutions to support wellness, prevention, diagnostics and treatment of medical conditions. These applications can enhance current assisted living solutions and reduce delay for treatment of critical patients. Nevertheless, besides the security issues of the technology, developers also face additional challenges that are interconnected. Challenges during the design and development refer to the followings: the regulatory environment of health care, diversity of devices, trusted and dynamic connectivity, the

technical debt of engineering devices, patients' safety, privacy, cost decisions that are made at the expenses of security, lack of preparation of the healthcare industrial sector, and ethical considerations. Patients' safety and privacy are fundamental values that might be positively or negatively affected by the functionalities of the technology. Developers make decisions on the functionalities that the technology offers. Thus, developers' decisions shape the technology development and the values that the technology could affect. Besides, the intrinsic nature of the IoT and the impact of this technology on moral values brings ethical implications. Value consideration and ethical implications should also be considered during the design and development process. Then, developers could offer applications that better deal with human values and which are socially accepted.

4

Qualitative Study

This chapter presents the research methodology employed for capturing developers' security practices and motivations for handling security requirements during the design and development of medical IoT applications. First, in section 4.1, we start describing the qualitative study design followed by the procedures for the sample selection of companies in section 4.2. Next, data collection process and data analysis approach are explained in sections 4.3 and 4.4 respectively. Finally, a reflection on the response rate is provided in section 4.5.

4.1 Qualitative study design

In this thesis project, we aim to understand the differences between security practices suggested by security requirements engineering field and developers' security practices in the context of IoT healthcare applications. IoT medical application developers are considered as the unit of analysis. For achieving the thesis purpose, we need to gather information regarding developers' practices to handle security requirements, their motivation for including security requirements into the medical applications, and their reasons for not incorporating security early in the development process. Data from interviews and documentation review are analyzed based on grounded theory principles.

4.2 Sample selection

The sampling strategy adopted in this research project is a *convenience sampling* (Miles, Huberman, & Saldana, 2014), which led to the identification of three companies. A detailed explanation of the company sampling process is given in this section.

Healthcare is one of the segments with the greatest market potential in terms of both revenues and growth rates in the IoT market (Mazhelis et al., 2013). In the healthcare arena, IoT applications and services can provide competitive advantages over current medical solutions (Miorandi et al., 2012). Moreover, new business opportunities attract companies for developing innovative healthcare applications and services. For convenience, companies developing healthcare IoT applications should be located in The Netherlands. The Netherlands has become a hub for innovation because of the facilities that the country offers to incumbent companies and startups companies (Egusa & Cohen, 2015). Furthermore, there is a strong incentive to promote

the development of innovative startups focus on digital healthcare applications (Amsterdam business news, 2016; Philips Research, 2004).

For finding suitable candidates in The Netherlands, an extensive Internet (re)search is conducted. The terms “medical IoT” or “healthcare IoT application” are not always used by organizations to define their applications. Therefore, terms such as MedTech, digital health, e-Health, smart health, care innovation, telemedicine, and digital wellbeing were used for the search. Because our focus is the development process, we started searching for startup companies developing medical applications. Web sites of business incubators and startup accelerators such as StartDelta, Iamsterdam (StartupAmsterdam), WorldStartupFactory, Yes!Delft, Startup+Health, rockstart, startupbootcamp; and sites such as Postscapes, foundedinholland, dutchstartupmap, ZorgInnovatie were carefully explored. Five startup companies were initially identified. Next, we continue searching for established companies that offer products or applications which could be considered as IoT medical application (see selection criteria above). As a result, eight companies were identified. However, it is worth to mention that from the website information, it was not clear if established companies are developing the technology or if they are integrating existing technologies to offer a service. The latter case might incentive companies to accept or reject the invitation.

In total, 13 companies between established companies and startups appear to be suitable for the study. For selecting the cases, the following selection criteria are defined.

Required selection criteria:

- The firm must develop applications for healthcare.
- The application should include at least one smart object (objects with sensing capabilities).
- Data from the smart object is processing as part of the application.
- Smart objects are connected to the internet or will be connected to the internet in the near future.
- Type of companies: startup and small or medium sized companies.
- For convenience, companies must have offices in The Netherlands.
- The firm is willing to participate in the study.

General information about the candidate companies (product, description, location, contact details) was collected from the company’s website. For contacting the candidate companies, an invitation letter via email was sent. The invitation introduced the research project and showed the company our interest in conducting an interview with them about security requirements. Thirteen emails were sent receiving three responses (2 positives) within one week. For improving the response rate, we contacted the companies, which did not answer the email, via a phone call. During the phone call, we ask if they were not interested in the interview or if they did not receive the email. After executing the phone calls, we were able to schedule an interview with one company, reaching a total number of three companies. Besides, one company had changed the type of application, and thus, it was not suitable for our study anymore. Participants who did not accept to be part of the study stated that they did not have the time or staff available for the interview.

The similitudes and differences among the companies that are willing to participate in the study give us significant insights of the current state of security practices during the development process. Table 9 introduces the profiles' demographics in terms of the type of organization and position of the subject in contact. On the one hand, the three companies develop medical applications that are employed or will be used in a care setting. It means that the applications operate inside a hospital or it is connected with the hospital's information system. Therefore, companies face similar challenges when introducing their products on the market.

On the other hand, first, two types of companies (startup and established companies) are represented in our sample. Startups are new emerged companies that aim to meet the marketplace by developing innovative product or services. These companies face different challenges such as developing the technology as fast as possible, finding funding for their activities, trying to enter the market, and so on (Giardino, Paternoster, Unterkalmsteiner, Gorschek, & Abrahamsson, 2016). Thus, it is believed that security is an afterthought in the development process. Meanwhile, incumbent companies have experience in the commercialization of their products and might have enough security expertise in their development activities. Our sample meets the proposed criteria for including startup companies, who are working on the applications' design and development (design and development include developing, testing, piloting, implementing, or upscaling their product), and established companies who successfully have developed and commercialized their IoT medical application in the Dutch market. Therefore, differences in security practices between these two groups can be observed.

Second, companies have different development approaches. Company1 has fully developed the application. Currently, they are working to improve functional and non-functional features of the application. Company2 employ an agile in-house development approach. Agile approaches focus on responding to unpredictability through incremental, iterative work cadences and empirical feedback (agilemethodology.org, 2008). Company3 utilizes a component-based development approach with an in-house integration. In a component-based development, development efforts are allocated to relatively independent subsystems of a manageable size (Tryfonas et al., 2001). Therefore, the effect of the development approach could be seen in the company's security practices. Finally, companies operate in different sectors within the healthcare domain.

Contacted organization	Type of organization	Subjects' position within organization
Company1	Incumbent company	Manager (also worked in the technology development)
Company2	Startup company	Product developer
Company3	Startup company	Product developer & Clinical Engineer Chief Technology Officer

Table 9: Demographics of the field study's profile

4.3 Data Collection

For enhancing the quality of this research three principles of data collection are followed. First, using multiple sources of evidence, not just a single source, which is known as *data triangulation*

is employed. Second a case study database is created and finally a chain of evidence is maintained. These principles will help us to deal with the problems of establishing the construct validity and reliability of the evidence (Yin, 2014).

The data for the study is collected through in-situ semi-structured interviews and documentation review where documents are available. Although it was originally planned to perform direct observations of practitioners in their field, none of the companies contacted allow the researcher to participate in workshops or design meetings. Semi-structured interviews were held with relevant respondents of the organization who are involved in the design and development process of the medical IoT application (manager, engineers, or designer). The outline of questions/topics were not sent to the respondents to avoid prepared or socially desirable responses during the interviews. In total, 4 semi-structured interviews were conducted.

During the interviews, participants answer questions about their role in the company, their practices during the design and development of the product, their perceptions on security, and some questions related to the technology. Table 10 presents an overview of the interview questions and related topic. The interview questions regarding practices for handling security requirements were derived from Chapter 2. Besides, questions regarding peoples' perception of security and technical features were included to get a better understanding of the context in which applications are being developed. Saturation principle was followed when conducting the interview. Interviews were recorded and transcribed in the English language. All information collected during the interviews was stored to create a case study database.

	Topic	**Interview Question**
Practices for handling security requirements	Type of application	▪ What kind of medical IoT applications is developed in your company? ▪ Can you explain a bit more the application?
	Development process	▪ How does the development process for a new product usually looks like in your company? o *Which different considerations usually you have to balance during the development process?* o *Which role does security considerations play in your development process?*
	Stakeholders	▪ Which stakeholders are involved in the design and development process of the application?
	Requirements engineering process	▪ Could you shortly describe the requirements engineering process of the application?
	Stakeholders' security concerns	▪ How do stakeholders communicate their security concerns (or security goals, security requirements)?
	Security requirements identification	▪ How do you elicit new requirements, especially security requirements? ▪ How do security concerns are refined to security requirements? o *Who is the responsible for this step?* o *Are security requirements been documented?*

	Security terminology	▪ Is there any glossary of terms or terminology guideline which have to be used during the requirements elicitation?
	Conflicting requirements	▪ Do you know of any problems you had in the past with conflicting requirements? if yes, how did you handle it? ▪ How do you agree on conflicting requirements?
	Categorization of requirements	▪ How are security requirements prioritized? And which are the criteria applied to prioritize security requirements?
	Risk management	▪ *Is the risk* analysis part of the development activities? ▪ How the risk appetite of the company influences the development activities to manage *security?*
	Integration of security requirements into the application	▪ How security requirements and functional requirements are unified? ○ *Do security requirements influence (constrain) functional requirements?* ○ *Do you need to remove/change security requirements to fit functional requirements better?*
	Satisfaction of security requirements	▪ How are security requirements satisfied in the application?
	Communication with application' users	▪ Do you inform users how the application must be used in order to achieve the implemented security?
	Security requirements methodologies	▪ Do you follow any security requirements methodology for the elicitation, analysis, validation of security requirements?
	Additional information	▪ Is something regarding security (practices) that we have not mentioned yet?
People's Perception	Meaning of security	▪ How do you define/conceive security? What does it (security) mean for you? ▪ If creativity is the main driver for innovation, security is…?
	Importance of security in the application	▪ How important is security for your IoT application? Why?
Technology	Technical features	▪ Which features/characteristics of the technology facilitate to consider security during the design and development of the application? ▪ Which features/characteristics of the technology make difficult to consider security during the design and development of the application?

Table 10: Overview of interview questions and related topic.

During the interviews, topics were not discussed at the same level of detail due to the different roles of respondents. Besides, not all topics were discussed because some companies do not include the expected practices in their development activities. Nevertheless, in the interviews, we try to keep an open conversation with participants to understand their reasons for adopting certain practices or for avoiding managing security requirements.

4.4 Data analysis approach

Data analysis consists of examining, categorizing, tabulating, testing, or otherwise recombining evidence, to produce empirically based findings (Yin, 2014). Field notes, memos, transcriptions, database, and other documents are used for the analysis of the data.

Grounded theory principles are employed to analyze the transcripts of the interviews and other documentation. Grounded theory is an inductive approach to discover "things" grounded in the data (why of doing things). The focus of this method is to understand the data from the data. This methodology has proven useful in gaining a better understanding of security issues, such as those surrounding user's perspective during the design of usable security (Flechais & Sasse, 2007) and in those in the area of information systems (Orlikowski, 1993).

The analysis process starts with an *exploration phase* that intends to discover concepts. For doing so, labels were assigned to raw data, and a low-level conceptualization was conducted using both in-vivo and open coding. This process was performed twice, and in different moments, to avoid researcher biases toward specific concepts. In this stage, *security practices were identified based on our descriptive framework*. Thereafter, the *specification phase* helped the researcher to develop further previous concepts and the make decision on core concepts. Concepts were grouped together into categories and subcategories. By means of axial coding, different relations between categories were described, and then, relations between subcategories and categories. Categories were redefined several times to adjust concepts. Coding paradigm introduced by Corbin & Strauss was utilized to find the relation between categories. Finally, in the reduction phase, selective coding helps us to focus on core concepts and in the analysis. The central idea is to construct a story line around core categories to explain the central phenomenon (Corbin & Strauss, 1990).

Multiples rounds of coding and multiples iterations among the exploration, specification, and reduction phases are performed to refine existing codes and discover emerged patterns in the data set. As described by Miles, Huberman, & Saldana; data analysis is not a linear process. Therefore, researchers need to move back and forward from the data collection, data display, data reduction, and conclusions drawing in an iterative way (Miles et al., 2014).

4.4.1 Coding Paradigm Model

To achieve the objective of this thesis project, we need to understand the reasons behind developers' actions. The process of *coding paradigm modeling* is useful because it helps to structure the categories in order to explain and understand the underlying phenomenon. According to Strauss & Corbin, the coding paradigm serves "as a reminder to code data for relevance to whatever phenomena are referenced by a given category" with special attention to "conditions, interactions among actors, strategies and tactics, consequences" (LaRossa, 2005,

pp. 846–847). The basic idea of the coding paradigm is to propose linkages and look to the data for validation (move from asking questions, generating propositions and making comparisons) (Pandit, 1996). Figure 5 depicts the basic features of the model.

Figure 5: Paradigm model process

The coding paradigm model says that *causal conditions* lead to the *central phenomenon* which might happen in certain *context* with *intervening conditions* where people take *actions/strategies* that have *consequences* (Pandit, 1996).

- *Causal conditions*: events or incidents that influence the central phenomenon.
- *Context*: stage or location of events.
- *Intervening conditions*: that shape, facilitate or constrain actor's strategies that take place within a specific context.
- *Actions or strategies*: strategies planned to manage, handle, carry out, or respond to the phenomenon under a set of perceived conditions.
- *Consequences*: outcomes or results of actions or interactions, results from the strategies.

Finally, to compare developers' practices to handle security and SRE recommended practices, we will employ our interpretative framework. The comparison will be based on the tasks identified in our interpretative framework (see section 2.2). The comparison is based on the adoption of SRE recommended practices by developers. Nevertheless, we special attentions is given to the existent differences and similarities. Moreover, we performed a comparison with the state-of-art to validate our findings.

4.5 Reflection on the response rate

As evidenced in section 4.2, the response rate for our study was very low (25%). Some of the reasons that could explain the low response rate are the following. First, security is a highly controversial topic. For instance, companies might prefer to bear the costs of security failures rather than to disclose data breaches. Failures in security damage the companies' reputation which is translated into financial losses. Thus, companies will prefer to keep and manage their security practices private rather than to share these practices with outsiders. Second, during the development of new technology, companies aim to protect their intellectual property. Product development is a risky activity, and businesses will opt for keeping a secrecy of development activities. Therefore, developers might prefer that outsiders do not have access to secret information regarding the technology. Besides, it is possible that the companies we contacted are neither developing new technology nor IoT applications and they will opt for not participating in the study. By checking websites of companies is not always possible to assess the phase of development of the product. Finally, due to the nature of the study, we needed to interview participants involved in the product development process. As indicated by three companies we contacted again after the first invitation were sent, technical personnel were not available for the

interviews. It is possible that developers were busy with the technology development and they did not have time for participating in the interviews.

5

Description of developers' practices

Based on the information provided by three SME companies developing IoT medical application, in this Chapter, we aim to gain insights on how developers handle security requirements during the design and development of IoT medical applications. Section 5.1 offers an overview of the data analysis. Section 5.2 describes developers' security practices within the interviewed companies developing IoT medical applications based on the interpretative framework introduced in section 2.2. Finally, section 5.3 provides a preliminary conclusion.

5.1 Data analysis overview

During the interviews, participants described a variety of activities that they perform during the design and development process. As mentioned in section 4.4, developers' security practices were identified during the *exploration phase* of the data analysis. In this phase, data from Company1, Company2, and Company3 was individually coded and examined. In this part of the analysis, we aimed to find practices that could be connected with *actions to direct a secure development* and actions for *eliciting security requirements, analyzing requirements, managing of requirements, and validating of requirements*. These tasks originate from the framework built in section 2.2 to structure the data collection and data analysis of developers' security practices. Initial *concepts* were obtained from Table 6 (Security practices within an organization) and SRE recommended practices (summarized in section 2.1.2). These initial concepts are illustrated in Table 11. Nevertheless, we tried to keep an open mind regarding emerging concepts that are not included as part of SRE practices but are relevant to developer's practices.

Task	**Concept**
High-level decisions and actions	Security policy Complying with security standards Risk analysis Assigning responsibilities
Elicitation	Stakeholders' involvement Identification of security goals Identification of security requirements
Analysis	Conflicting requirements Engaging in negotiations Prioritizing requirements

Management	Conflicting functional requirements and security requirements Security mechanisms Integration of security
Validation and verification	Validation of security requirements Verification of security mechanisms Satisfaction of security requirements

Table 11: Initial concepts derived from SRE recommended practices to explore developers' practices

To assess if an activity can be considered as an "elicitation practice," for instance, concepts such as *stakeholders' involvement*, *identification of security goals*, *identification of security requirements* were explored within categories and subcategories of the open coding (see Chapter 4 Methodology). Practices that could be linked to these concepts were classified as "elicitation practices." The same process was followed to assess if the activity can be described as an "analysis practice," concepts such as *conflicting requirements*, *engaging in negotiations*, *prioritizing requirements* were scanned in the open coding. For practices considered as "management practices," the main concepts were *conflicting functional requirements and security requirements, security mechanisms, integration of security*. Lastly, for practices considered as "validation and verification practices" concepts such as *validation of security requirements, verification of security mechanisms, and satisfaction of security requirements* were explored.

Figure 6 shows an example of the process followed to identify elicitation practices. Codes in column F represent the open coding of developer's statements. These codes are related to the concept *identification of security requirements*, which is a concept linked to "elicitation practices." For instance, the code 'security officer as a source of security requirements' implies that developers obtain security requirements from security officers. Thus, the practice of gather security requirements from security officers is considered as an "elicitation practice." Column J helps us to distinguish the type of practice.

E	F	G	H	I	J
#	**Open coding**	**Concept**	**Subcategory**	**Categories**	Is a SP the actio
23	Consult company's security officer according to the topic	Identification of SR	Development process	Stakeholders involvement	SP - Elicitation
31	Security officer as source of security requirements	Identification of SR		Development process	SP - Elicitation
32	Security requirement based on regulation	Identification of SR		Development process	SP - Elicitation
33	Security requirements as a trend among customers	Identification of SR		Development process	SP - Elicitation
34	Hospital's security officers as sources of SR	Identification of SR		Development process	SP - Elicitation
38	Gaining knowledge from Hospital Security Officer	Identification of SR	Security knowledge	Stakeholders involvement	SP - Elicitation
54	Third party (host) as source of SR/concerns/information	Identification of SR		Stakeholders involvement	SP - Elicitation
79	Same requirement popping up might be a good idea to consider	Identification of SR		Development process	SP - Elicitation

Figure 6: Example of open coding for describing developers' practices

Moreover, if an activity cannot be linked to any of the concepts illustrated in Table 11, but it could be interpreted as an elicitation, analysis, management, or validation and verification practice

(according to the definition provided by the framework in section 2.2), the action was also classified as security practices.

5.2 Developer's practices to handle security

As explained in section 2.2, the interpretative framework is used to organize developers' security practices. The framework differentiates *managerial* actions which could motivate a secure development and *operational* procedures and actions to consider security requirements early in the design and development process. Table 12 presents the categorized security practices of the developers in the companies interviewed. Developers' practices are distributed on two levels: *managerial* and *operational*. Within the *managerial* level, for instance, to comply with rules, regulations and standards early in the development process is a practice that might drive actions to fulfills legal requirements. Thus, this practice is considered as a *managerial* practice. Within the *operational* level, companies' elicitation, analysis, management, and verification practices are categorized. Columns display the practices of Company1, Company2, and Company3 respectively. In the remainder of this section, table entries are described in these two levels within a company (*managerial* and *operational*).

5.2.1 Practices at the managerial level

Managers in C3 opted for taking into account the required requirements that allow them to be CE compliance. They have personnel and consulting companies to check the regulation. C1 aims to ensure that the final product will meet the required rules and regulations. Besides, as part of the documentation that C3 should present to the regulatory entities, they should include a risk analysis of the product. Managers of C3 have commanded the required actions to fulfill this requirement. C1 is already HIPAA, FDA, and CE compliant. Nevertheless, they try to follow all the required updates and procedures for continuing being regulation compliant. In the case of C2, manager work on the technology development. At this point, they prefer to focus on the functionalities of the technology. Thus, complying with rules and regulations could be a common practice for companies. However, it does not mean that companies consider rules and regulations early in the development process.

None of the interviewed companies have indicated if they have established security policies for the product development or if they use the companies' security policies. In addition, it seems that companies do not conduct a risk analysis to determine their risk tolerance. Finally, responsibilities for addressing security are not assigned during the development process.

5.2.2 Practices at the operational level

In practice, the interviewed companies do not have a distinctive process to handle security requirements. When security requirements are identified, if they are, it is alongside functional requirements. Thus, in this section, security practices of Company1, Company2, and Company3 (i.e., C1, C2, and C3) are described from their standard development process. Nevertheless, developers' practices are categorized based on our descriptive framework.

Elicitation of security requirements:

Involving different stakeholders is a common practice during the design and development of medical applications. As part of the development process, C2 and C3 include medical personnel (i.e. doctors, nurses, and specialists), potential customers, technology experts, programmers, designers, among others. According to C2 and C3, including a variety of expertise allows them to understand better the problem that practitioners or patients face, the necessities of customers, and the technologies available. Moreover, university hospitals provide a space to C2 and C3 for testing some features of the application in real settings. C3 includes compliance and regulatory consulting companies and notified bodies[1] to incorporate the required legal requirements into the application. Partners or third parties working with the companies to provide the service or to develop the technology should also comply with the existent rules, regulations, and standards according to C1. Only C1 has included a security officer as part of their stakeholders.

Identifying functional requirements is a standard step in the development process. Requirements and features of the technology are identified from the stakeholders mentioned above. C2 and C3 use meetings, design workshops, and interviews to gather information regarding customers' needs. C1 attempts to incorporate customer's desired features into the application to offer a better product. Moreover, C3 has opted for including requirements from rules and regulation during the requirements collection. In this case, all requirements needed to comply with rules, regulations, and standards are included early in the development process.

Regarding security requirements, C1 *gathers security requirements* from different sources. Rules and regulations, company's security expertise, hospital IT personnel or hospital security officer, customers, third parties' security expertise are the most common. It is worth to mention that developers refer to security concerns and security mechanisms as security requirements. For them, there is no difference between those three concepts.

Analysis of requirements:

Requirements are prioritized to build a prototype of the application. As part of the development process, C3 has built a prototype to test the feasibility of the application's functionalities. C2 is working to introduce its prototype of the application at the end of the year. According to these companies, a prototype allows developers to test functionalities, analyze performance, and scan customers' reactions to the product. For the interviewed companies, the main criteria to prioritize requirements is functionality. C2 prioritizes functionalities based on discussions between stakeholders and developers. C3 ranks requirements according to developers' experience; rules, regulations and standards; and usability.

In the case of C1, the application is already developed and placed on the market. Developers of C1 *prioritize security requirements* based on rules and regulations. Whenever a change in rules and regulations require new security requirements, these new requirements will be integrated into the application. After that, hospitals' security requirements are included. To incorporate hospitals' security requirements, developers analyze the financial consequences of including the required

[1] Notified body is an organization designated by an EU country to assess the conformity of certain products before being placed on the market. These bodies carry out tasks related to conformity assessment procedures set out in the applicable legislation, when a third party is required (European Commission, 2017).

security measure. Moreover, C1 *engage in negotiations to gain time for incorporating the needed requirements*. Developers enter negotiations with hospital's administrative personnel if the product meets a particular hospital's need. The aim is to gain valuable time to incorporate all the required security requirements. Finally, C1 documents the application's security requirements to comply with rules and regulations, and with hospitals' regulation.

Management of requirements:

Security mechanisms are employed to safeguards some features of the application. During the requirement phase of the development, C3 considered security requirements from regulations as part of applications' functional and non-functional requirements. Then, they will incorporate the necessary security mechanisms into the application to comply with regulations and standards. Besides, well-known security mechanisms are employed to protect assets or features that developers consider important. Communication, for instance, is understood as a fundamental aspect of C1's application. Thus, encryption is used to prevent unauthorized access. C2 and C3 prefer to connect key components of the application using cables instead of wireless during the prototype phase. Both companies consider cable as a secure mean of communication. In the case of C1, developers also include the security mechanisms required by law or by hospitals into the application. C1 intends to use security technologies that fit their market to avoid extra complication to users.

C1 also *maintains open communication with the third parties* involved in delivering the intended solution. An open communication allows developers and third parties to share experiences about vulnerabilities and possible threats that affect the system. During the training programs, C1 *explains to users how to utilize the application and the security mechanism* that they will encounter. For instances, to choose a valid username and password for user' identification and authentication which should not be used in another type of applications.

Validation and Verification:

The validation and verification process is more related to *ensure that technology features operate in a specified manner*. For instance, C2 tries to ensure that the application is getting the right measurements or inputs. C3 works to ensure that right data is transmitted to the right patient monitor. C1 and C3 also verify that the application fulfills the required security requirements and security mechanisms to provide evidence to regulatory authorities. The application of C1 is HIPAA compliance. Thus, external parties will assess the security of the application by performing a penetration test.

In the following section, we aim to understand why, in practice difficult to be considered security requirements early in the development process.

5.3 Preliminary conclusion

In this chapter, section 5.2 describes some of the practices adopted by developers of the interviewed companies during the development process. Even though we aimed to emphasize practices to handle security requirements during the design and development of IoT medical applications, there were few actions that can be classified as a practice to manage security requirements. It appears that early in the development process security requirements are not as

important as functional requirements. At the managerial level, managers aim to comply with rules and regulations to be able of commercializing their products in the healthcare sector. At an operational level, stakeholders and developers focus primarily on functional requirements, and security requirements are left for later phases of the development. Some security mechanisms are elicited at any stage of development based on developers' experiences.

	Company1	Company2	Company3
Managerial	▪ Complying with rules, regulations, and standards (HIPAA, FDA, CE)		▪ Working to comply with required medical standards (CE) ▪ Performing Risk Assessment of the product as part of regulations.
Operational	Elicitation: ▪ Involving certified security expert (when required) and reliable partners that comply with required rules and regulations. ▪ Gathering security requirements (concerns or information) from different sources.	▪ Involving stakeholders with different expertise (hospital personnel – nurses, doctors, ICT department – professors). ▪ Discussing security with hospitals' ICT department. ▪ Identifying requirements from stakeholders.	▪ Involving multiple partners with different expertise (electrical, mechanical, software, usability, system engineering, regulatory, quality, manufacturing, notified bodies). ▪ Involving different expertise as part of the company (medical, technical, clinical). ▪ Involving hospitals departments that check devices used in hospitals. ▪ Identifying requirements from different sources.
	Analysis: ▪ Prioritizing security requirements by law, customer requirements, financial consequences / incentives. ▪ Identifying differences between customers' needs and security officers' requirements. ▪ Assessing the origin of security requirements (law, hospital, security officer). ▪ Engaging in negotiations with customers and/or hospitals'	▪ Prioritizing requirements based on discussions between stakeholders and developers.	▪ Prioritizing requirements based on company experts' experience; rules, regulations, and standards; usability.

	security officers to gain space for fulfilling all hospital security requirements. ▪ Documenting security requirements.		
	Management: ▪ Including security mechanisms that fit the market ▪ Keeping open communication regarding security, updates, vulnerabilities with third parties involved. ▪ Training customers on how to utilize the application for achieving its security. ▪ Achieving a balance between usability, technical working, and security regulation.	▪ Using cables as a mean of communication	▪ Applying cryptography solutions to secure communication between transceiver and receiver (security mechanism). ▪ Using cables as a mean of communication ▪ Third parties disintegrate company' requirements and arrive at specifications.
	Validation and verification: ▪ Providing evidence of compliance with hospital's security requirements. ▪ Assessing security of the application (pen test as part of HIPAA compliance).	▪ Ensuring application is getting the right measurements or inputs.	▪ Ensuring that right data is transmitted to the right patient monitor. ▪ Working to provide right evidence to notified bodies for meeting medical standards.

Table 12: Developers' security practices during the medical applications' design and development.

6

Factors influencing developer's practices

The overall objective of this Chapter is to find factors that could explain why security requirements are not incorporated early in the development process as showed in section 5.2. Section 6.1 offers an overview of the data analysis process. Next, section 6.2 gives an overview of the identified factors which could influence developers' strategies to address security during the application's development process. Sections 6.3, 6.4, 6.5, 6.6 explain the identified factors within high-level categories. Afterward, section 6.7 states the consequences of the factors influencing developers' practices. Section 6.8 briefly discusses the generalization of our findings. Finally, section 6.8 offers a preliminary conclusion of the chapter.

6.1 Data analysis overview

For analyzing the factors that influence developers' security practices, the data obtained from the interviews was examined per individual company, as well as across companies, to detect similarities and compare differences. Data from Company1 and Company2 was individually coded and analyzed. Data from Company3 was included later in the analysis because personnel from Company3 was not available for the interview during the interview period. The initial focus of the analysis was on the development of concepts, proprieties, and relations. Once data from both organizations was scrutinized, emergent concepts were organized by recurring theme.

Emerging concepts from the analysis were linked and categorized by following the coding paradigm model (see section 4.4.1). The model used for the analysis is described below. The high-level categories are *context, conditions for addressing security, strategies to deal with security, and intervening conditions*. These categories were developed to understand better what motivates developers to consider or to neglect security during the design and development of the medical application, how developers approach to manage security, and the circumstances that might influence developers' decision and actions.

- *Central phenomena:* addressing security during the development process.
- *Causal conditions* that lead to address security (central phenomena).
- *Strategies* that developers take to avoid or to incorporate security into the application.
- *Intervening conditions* that facilitate or constrain the developers' *strategies*

By following a constant comparative analysis, categorized concepts from Company1 and Company 2 were systematically compared and contrasted. During this process, we aimed to

determine the set of categories and emerging concepts that cover as much data as possible. Some categories and emerging concepts generated from Company1's data matched the categories and concepts generated from Company2. Thus, similar patterns were found between these two companies. Meanwhile, some concepts from Company2 did not match categories from Company1. For instances, the concept *managing responsibility* was initially considered as a part of the category "intervening conditions," however, we could see that for Company2 it was not an intervening condition that influences their strategy. After re-analyzing the data, *managing responsibility* was redefined as a sub-concept of *rules and regulations*. Because, due to rules and regulations, companies are held responsible for ensuring data protection, for example. Therefore, responsibility is part of *rules and regulations* rather than an intervening condition for the companies' strategy. Table 13 illustrates a summarized matrix that displays the compared key events, triggers, and outcomes from the three companies organized in high-level categories according to the coding paradigm model (see Appendix A for a complete version). The first two columns represent categories and concepts, and the next columns show the supporting information from each company.

The data obtained during the interviews conducted in Company3 helped us to corroborate concepts and categories. Data from Company3 was initially coded and analyzed. Next, emerging concepts and sub-concepts were re-examined and re-coded using this proposed scheme. Constancy among categories and concepts were validated by an iterative exploring and analyzing of the information. In the remainder of this section, we provide a detailed description and explanation of the identified factors.

Categories	Concepts	Data from Company1	Data from Company2	Data from Company3
Context	Setting where application operates.	▪ Patients are monitoring from home, and the information is sent to the hospital.	▪ Application for monitoring patients in a hospital.	▪ Application for monitoring patients 24 hours in a hospital.
	Stage of development	▪ The application is fully developed (upscaling).	▪ Prototype phase close to launch a pilot product. ▪ Prototype version does not need all functionalities.	▪ Testing feasibility of a prototype application. ▪ Working on converting the device in a small version.
Conditions for addressing security	Developers' perception of security	▪ Security is considered as a selling point.	▪ Security is considered as a feature of integrating the app with ICT.	▪ Company is satisfied if customers feel secure using the application. ▪ Security is interpreted differently in different countries; thus, rules and regulations also differ.
	Rules and regulation	▪ Regulations for e-Health products that process personally identifiable data. ▪ The company is held responsible for patients' data. ▪ Hospital and parties processing/analyzing patients' data sign an agreement.	▪ Analyzing rules and regulations for different part of the application	▪ Medical devices need to comply with European rules, regulations, and standards. ▪ Companies should provide evidence to notified bodies. ▪ Risk assessment as part of the regulation.
	Hospital security knowledge	▪ Hospitals required specific security requirements. ▪ Hospitals have security officers.	▪ Hospital's ICT should approve data transmission. ▪ IT questions security issues brought by the application.	▪ Hospital has specific departments to check all devices used in the hospital (prototype, clinical trials, etc.)
	Development approach	▪ Development cycles and road map to organize activities. ▪ Focus on providing new functionalities and improving the application/product.	▪ Iterative and incremental product development process ▪ Features prioritized based on stakeholders' needs.	▪ Multiples parties to develop the technology. ▪ In-house concept and integration of the application.
Strategies for dealing with security	Involving reliable stakeholders	▪ Incorporate security expertise as part of the company. ▪ Involving responsible third parties that follow CE rules.	▪ For the integration with ICT company expect to work with a third-party expert on security or a well-known brand that generates trust.	▪ Involve multiples parties with different expertise. ▪ Involve partners with experience in healthcare

	Incorporating SR when it's required	▪ SR are included in the roadmap and addressed according to priority. ▪ SR are prioritized by law, customer, and financial consequences.	▪ The security issues of integrating device with ICT will be tackled as a final phase in the development process.	▪ The company will follow all the rules and regulations for IoT medical applications when the application arrives at that point (expected in 2020).
	Adjusting technology features	▪ Balance hardware and software of the application. ▪ Building the system in accordance with rules and regulations.	▪ Balance hardware and software of the application. ▪ Very output/measurements of the device. ▪ Application does not transmit patient's data.	▪ Ensuring right data is transmitted to the right receiver. ▪ Apply cryptography. ▪ The application does not take patient's information.
Intervening conditions	Developers' perception	▪ Security requirements constrain usability, and customers feel the constraint.	▪ Failures in security damage reputation. ▪ Lack of expertise make difficult to address security issues.	▪ One small incident can stop the whole business.
	Customers' necessities	▪ Different necessities between the customer and security officers. ▪ Customers place less importance on security.	▪ Customers has to like product. First, focus on satisfying customers' needs. ▪ Nurse/specialists are usually not concerned with (ICT) security.	▪ Customers (doctor, nurses, and parents) have to trust on the product. ▪ The focus is to gain acceptance of medical professionals.
	Challenges facing during the development	▪ Rules and regulations differ per country. ▪ Rules and regulations change raptly. ▪ Not all required requirements can be integrated into the app.	▪ The product should be protected against bacteria. ▪ Product's features should not harm patients' private information or hygiene. ▪ Bounded rationality regarding all possible threats and situations.	▪ The product has to be safe enough to place on the patient. ▪ Achieve the required battery life performance. ▪ Manage whole team and all partners while protecting the companies' IP.
	Company's experience	▪ Commercializing the application in the European market for more than six years.	▪ Working on the product/application development for more than one year.	▪ Working on the application development for more than one year. Founders have experience in medical field and technology development.

Table 13: Factors influencing developers' security practices: concepts and findings (summarized version)

6.2 Factors overview

As mentioned above, Table 13 shows the high-level categories according to our coding paradigm model (context, conditions for addressing security, strategies for dealing with security, and intervening conditions) and the related concepts. Figure 7 illustrates these categories and concepts that emerge from the data analysis, and their interactions. Relation are represented by arrows and categories are represented by blocks. Figure 7 can help us to formulate possible explanations for why developers choose specific strategies to address security, their reasons for incorporating, or avoiding, security requirements in the IoT application, and to understand the intervening conditions that facilitate or constrain developers' strategies.

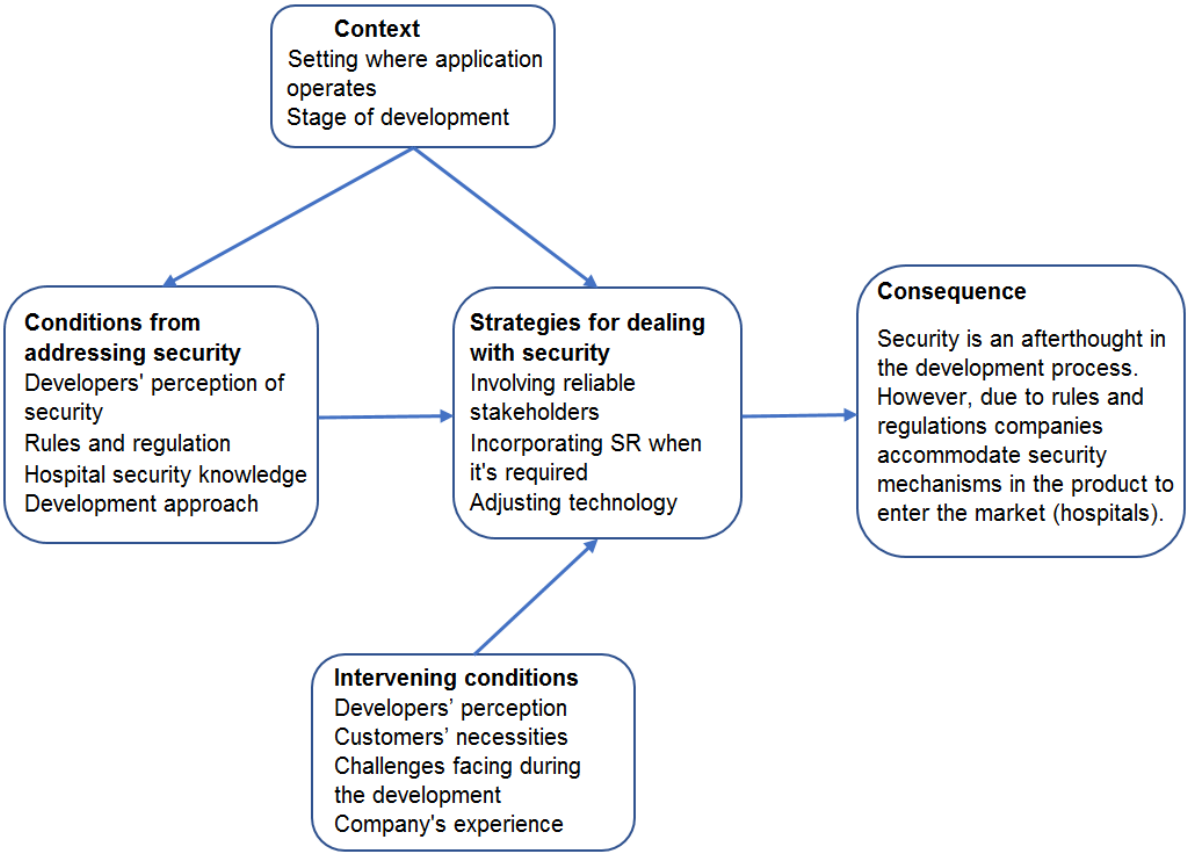

Figure 7: Factors influencing developers' actions: key concepts and interactions.

From the data analysis, factors such as *developers' perception of security, rules and regulations, hospital security knowledge, and development approaches* could be **considered as causal conditions for addressing security**. These factors lead to the adoption of **strategies to incorporate security requirements** into the application. Although developers have different approaches (strategies) to deal with security; *involve reliable stakeholders*, *incorporate security requirements when it is required*, and *adjust technology features* characterize to some extend our companies. Factors such as *developers' perception, customers' needs, challenges faced during the development, and companies experience* could be considered as **intervening conditions** which facilitate or constrain the adoption of the developers' strategies. Intervening conditions are

fundamental in our study because they could explain the reasons behind developers' strategies. Finally, the conditions for addressing security and the strategies adopted for dealing with security might be influenced by the *setting where the application operates* and the *stage of development* of the medical application. The two latest factors are part of the category **context**.

In the remainder of this section, we explain the categories and concepts presented in Figure 7. These concepts embody the factors that are likely to explain developers' practices during the technology development. During this section, we use identifiers of the companies presented in Table 9 (i.e., C1, C2, and C3) to highlight statements made by the interviewees.

6.3 Context

The category context refers to the stage or locations where events occur. The context could help us to understand the circumstances that surround developers' practices. From the data analysis, we found that factors such as *setting where the application operates* and *the stage of development* can be considered as part of the context. Moreover, the context can influence the causal conditions to address security and the strategies adopted by developers.

6.3.1 Setting where application operates

The three applications, which are part of this study, operates in the same environmental context: healthcare institutions. C1's application allows doctors and nurses to monitor patients at home real time. Patients perform the self-administrative test in their houses, and the results of the test are sent in real time to the hospital. The application from C2 and C3 allows doctors and nurses to monitor patients in the hospital environment. Because these applications are integrated with the hospital's information system or operate inside a hospital, the applications needed to comply with rules, regulations, and requirements which are different from the consumer market. Section 6.4 (rules and regulations) explain this point in more detail. Besides, as these applications will operate in healthcare institutions, they should comply with hospitals' requirements. Thus, causal conditions for addressing security also depend on the setting where the application operates.

6.3.2 Stage of development

From our three applications, the application of C1 is fully developed. According to developers of C1, they are upscaling the product to provide better functionalities to users and to raise the standards and quality of the application. The company organizes the development per quarter, and the needed changes are incorporated in the roadmap of activities. Upgrades in software are implemented according to the urgency and planning. Changes in hardware are more difficult to implement.

C2 and C3 are developing the application. C2's application is in a prototype phase. The company expects to launch a pilot of the product to the end of the year. According to developers of C2, as a prototype, the application will include key functionalities and features. Besides, functionalities and features will be interactively refined over time, and new functionalities will be added in the future. C3 is testing the feasibility of a prototype application. They are also working on converting the device in a small version. For both companies, the purpose of the prototype is to test functionalities in a real environment and to validate the product in the market.

Development practices occur during the development process; however, practices change and evolve according to different stages of development. For instance, in the case of C1, the company is working toward improving the product. Meanwhile, C2 and C3 are developing the product. C1 is taking into account the required security requirements to continue selling the product. And C2 and C3 are focusing on testing the feasibility of their concepts. Thus, these three companies have different priorities regarding security which are reflected in their strategies to address security.

6.4 Conditions for addressing security

The category conditions for addressing security refers to events or incidents that could influence developers to incorporate security requirements into the application or to avoid handling security requirements during the applications' development. From the data analysis, we found that factors such as *positive developers' perception of security, rules and regulations, and hospital security knowledge*, and *development approach* could be considered as causal conditions to manage security.

6.4.1 Developers' perception of security

Developers, which include managers and engineers, have different perspectives on security. These perceptions of developers might influce their decision to address security. As C1 mentioned: "*At the end, I would like to comply [with the hospital security requirements] as much as possible because if my application is very secure, that is a selling point to other customers as well…"* Besides, *"…when we finish with X hospital, the next customers were easy because we could say we comply with all security requirements of X hospital…* […] *We are proud of all the effort that we put into this, and we use it as a selling argument, we sell that we are CE compliant HIPAA compliant*…" Thus, this suggest that developers' perception of security as a mean to maintain customers or as a selling argument influence developers to incorporate security requirements in the application.

Other arguments provided by developers imply a different perception of the role of security. For instance, C1 commented: "*I think security is very important, patients should be able to trust that their data is secure and not leak on the street by anyone or any system.*" C2 also has mentioned: "...*for us guarantee trust is very important. Also, guarantee privacy, the privacy of the patients' data…"* Developers consider security as a feature to provide trust on the application and to protect patients' information. Besides, as C3 mentioned: *"… we are in the world of startups, [startups] have problems to access the market because maybe they have a great product, but they don't how to sell it. And doctors, they don't trust on startups."* Companies, especially startups, would need to gain trust from medical personnel and hospitals. Doctors, nurses, and patients should feel secure using the device. And information of patients should not be employed for other purposes or exposed to unauthorized actors. Therefore, developers' perception of security as a mean to generate trust on the application by protecting patients and data motivate developers to make an effort for accommodating security into the application.

At the same time, developers have a less positive connotation of security. C3 mentioned: *"…our current focus is on the usability more than infrastructure… […] usability comes third to prioritize requirements, the overall workflow should come easy…*" In this regard, C2 commented: "*It is all*

about the nurse if the nurse like our product and [she] can use our product, we are happy…" Developers consider usability as a key requirement. Users (doctors, nurses, and even patients) should be able to operate the system in an easy way. However, as C1 pointed out: *"…between security and usability there is a balance that needs to be found, and it is difficult to explain to your customers why they have to do certain things…"* Developers perceive security as a constraint of an easy to use application. And users are the ones who feel the constraint. C1 remarks: "…*there is some rule that says that customer needs to log out after X minutes not using the system. If a nurse is login in a real-time test that she is following, she has it on the screen but she does not do anything, after certain period of time the system automatically logs her out which is pain for her [….] she needs to log in again…"* In addition, As C1 commented: "*Security is a limitation for bringing your product to the market we know that every day...*" "*If you want to launch a new product … [...] you need to comply with so many rules and regulations that it is almost impossible to start in the Netherlands with a new product...*" In this regard, C2 also mentioned: *"I think [security] does not constraint functionality, but it could potentially constrain the further development of the product…"* Security is perceived as a limitation to bring new products to the market. Therefore, it suggest that developers' perception of security as a constraint to usability and as a limitation for bringing products to the market quickly negatively affect the incorporation of security requirements into the application.

Figure 8: Developer's perception of security (based on developers' arguments)

As discussed above, developers interviewed in this study understand in different ways the same idea "security." Figure 8 illustrates these different perceptions of security. Developers' perceptions represented in color turquoise and green positively influence the decisions and actions of incorporating security. On the other hand, developers' perceptions represented in color blue could

negatively affect the integration of security. Finally, some misunderstandings lead to think that security is only an issue of the ICT or a part of the infrastructure. Then, security could not be addressed until the product is integrated with the hospital information system. This conception of security might result in developing products that lack basic security requirements for being networked.

6.4.2 Rules and regulations

Rules, regulations, and standards could play a role in addressing security. In this regard, C3 commented: "…*we work on the requirements for CE Mark, for having the QMS certification which is quality management system for medical devices… […] …we cannot sell the product until we get the CE Mark… when you got it, you can sell it in Europe and in other countries that accept the CE Mark.*" C2 also mentioned: *"By law, I need to have a contract about privacy and security… […] basically it says that I am working with the customers' data, and that is a mandatory document that as soon as [hospitals] allow other thirds parties to work on their data they need to sign a contract."* Medical devices need to comply with existing rules, regulations, and standards to be sold in Europe and other countries. For instance, according to the information provided by the website of the European Commission, medical devices should bear the CE Mark to indicate their conformity with the provisions of the European's rules and regulations to enable them to move freely within the Community (European Parliament, 2007). The CE marking signify that products sold in the European Economic Area have been assessed to meet high safety, health, and environmental protection requirements. (European Commission, 2017a). Furthermore, documents examined during the documentation review indicated that medical applications that process personally identifiable data are regulated by the Personal Data Protection Act., and NEN standard 7510, information security in care (a compilation of ISO 27001, 27002 and 27799). Under the EU law, personal data can only be gathered legally under strict conditions, for a legitimate purpose. Persons or organizations which collect and manage personal information must protect it from misuse and must respect certain rights of the data owners with are guaranteed by EU law (European Commission, 2017b). According to Dutch Standard Institute, information security is especially important in the healthcare sector where medical and patient data is managed and exchanged. NEN 7510 provides guidelines and assumptions for determining, setting, and maintaining measures that healthcare providers and others administrators need to take in order to protect the information (NEN, 2017). Lastly, according to the Dutch Association of Hospitals (NVZ), external parties processing and (scientific) analyzing hospitals/patients' data should sign an agreement with the hospital. The agreement ensures that requirements and responsibilities in the field of information security and privacy are properly regulated and recorded (Nederlandse Vereniging van Ziekenhuizen, 2016). Therefore, it seems that rules, regulations, and standards are a motivator for addressing security during the design and development of an application or later before its commercialization. If developers do not comply with the existing regulation, they cannot place their products on the market.

Responsibility is a sub concept of the category rules and regulations. As C1 remarked: *"It is their patients, but I record and store their data, I am responsible for securing the data…"* Companies are held *responsible* for guaranteeing the compliance of rules, regulations, and standards. Thus, companies need to ensure that the application meets the security standards required by the

authorities. In addition, C1 mentioned: "*…as long as I am responsible for anyone behind me, customers don't care, because if something goes wrong, they come to me, and then, I need to pay the fine, which is high fine…."* As legal representatives of the application in front of the authorities, companies are responsible not only for themselves but also for all parties working with them to provide the service. Financial consequences could ensure that companies safeguard patients' data, and thus, incorporate security into the application. To manage the *responsibility*, however, companies could opt for transferring this responsibility to third parties. For instance, as mentioned by C1: "*security insurances*" are purchased to minimize risk. Companies also have the same agreement (agreement with external processors) with third parties working with them. Although these mechanism helps to manage *responsibility*, companies still need to fulfill the required rules and regulation to be insured.

6.4.3 Hospital security knowledge

It seems that there is a growing awareness of security within medical institutions. In this regard, C1 commented: *"[role of security considerations in the development process] …a lot. E-health products are put under a radar…. Also, a lot of hospitals now employ security officers I would say e-health specialists. When we started with our first customers no one ask for privacy of the patient and how it will be secure… now that is the first question."* Hospitals are incorporating security expertise, information security officers, as part of the organization. Furthermore, as C1 mentioned: *"…X hospital, they have a requirements list of security requirements needed. If you don't comply with one, you don't get the job; you don't get to sell the product. That is the first thing they communicate.*" Some hospitals provide a list of security requirements that application should meet in order to operate in the healthcare institution. Hospitals will list security requirements for software applications, which process personally identifiable data, that are developed for or on behalf the hospital and/or put into use in the institution (according to hospital X security requirement document). Hospital security officers could verify that software applications meet the required security requirements. Even though, it is not clear if hospitals' awareness is a consequence of rules and regulations, we could say that hospitals asking for and verifying security requirements could motivate developers to incorporate these security requirements into the application. Otherwise, it is difficult for them to sell the product in the hospital.

This is also be true for companies testing their application in university hospitals. As mentioned by C3: *"…this […] is the department that checks all the devices that you can use in hospitals for prototype, clinical trial, clinical trials, everything…"* C2 also remarked: "…*IT department, they want to know if there is a potential way for a hacker to come through your device and do something within the network internally in the hospital.*" *"They [IT] also say that there are some protocols that are needed. It is a basic call…"* During meetings between developers and hospitals' personnel, hospital IT departments inform developers about some protocols that they need to consider during the application's development. Thus, a basic level of security needs to be incorporated to test the prototypes in real settings. In addition, as C1 commented: *"… we don't send information outside the hospital because it is a no go for the ICT department."* Special permissions are needed to send information from the hospital to the outside. The fact that hospitals' IT departments are also asking about the security of the application could influence developers to consider security requirements during the design and development of the product.

6.4.4 Development approach

Developers choose different development approaches. In this regard C2 commented: *"We work with scrum, we have spins of one week, and with sprint planning on Monday and then review and review and retrospective on Friday… […] scrum model helps us to reduce our budget needs and for that reason we choose such a way of work."* As indicated by C2, in SCRUM, developers present a design of the product to stakeholders. Stakeholders make questions about the design and developers reply to these questions. During these discussions, requirements and tasks are prioritized based on the stakeholders' opinions and necessities. C3 has opted for a different approach. As C3 also mentioned*: "…our model is like […] we have different technology partners, of course, we cannot do everything in house because it's way too expensive. So, we partner with the right companies to get this product on time in the market, and then we share the revenues with them."* Thus, developers choose development approaches that meet their needs and fit their capacities.

Furthermore, as remarked by C2: "*… so, based on that [the problem] we have functionalities. We start with steps, and then we go further… […] we have changed the product a lot, but we try to have a minimum, the functionalities need to work each time in some way…"* C3 also confirmed: "*for us, functions and features come the first all the time. We have to secure that part because that is the core of our business.*" Developers place a high value to functionalities. The development approach employed to build the IoT medical application should allow developers to provide functionalities as fast as possible. C2 remarked in this regard: "… *as a startup; you don't want to spend a lot of time to test your product in the real world, you want to be there as quickly as possible*…" Therefore, this suggests that development approaches, such as iterative and incremental agile product development process, are a key factor to underestimate security during the design and development of the application.

Incremental development approaches provide additional advantages to developers. For instance, as C3 comment: "*After 3 to 5 years, once the whole IoT space is going to stabilize, hopefully by 2020, we should have our cloud space as well*..." "[...] *we expect that we could monitor our system. Or cloud will be a necessity because patients need to be monitor for another few months at home…."* Applications can be designed to accommodate new functionalities in the future. This aspect is fundamental in the development of IoT applications because, from the organizations we have visited, companies first start developing the "smart" object with a limited connection to the Internet. This helps developers to test functionalities in a real environment without bringing extra complications. After that, when they have proved that the application operates effectively, developers aim to move forward to integrate the smart object with the Internet. In this case, security measures for the integration with cloud solutions or the Internet will be analyzed at that moment.

6.5 Strategies for dealing with security

The category strategies for dealing with security refers to the actions take by developers to incorporate security requirements during the design and development of the application. These strategies could also focus on avoiding the accommodation of security requirements into the application. Besides, some strategies could serve both purposes: accommodating and neglecting

security. From the data analysis, we found that factors such as *involve stakeholders, incorporate security requirements when it is required*, and *adjust technological features* could be considered as part of the strategies adopted to accommodate security.

6.5.1 Involving reliable stakeholders

Companies involve different stakeholders during the development process and later when they offer the product or services. As C1 mentioned: *"…internal stakeholders involved in the development process include… […] security officer. He does not see all request we sent through. But the major ones when we think security is affected we let him know… […] security officer is from the mother company, he is external."* "*We have regular meetings with him where we discussed what he learned from the market and what he learned from other companies… […] a large part of the input [security] comes from him, where he addresses the things that might come out in the future."* Organizations could opt for hiring security officers. Security officers will provide valuable information about the required security requirements, threats and vulnerabilities in the field of application, and changes in legislation that could have something to do with security or privacy. Moreover, C1 commented: "…*our hosting company is CE register, and they follow all rules and regulations for hosting and securing medical data.*" Third parties working with the company also have to comply with all rules and regulation to ensure that security measures are properly considered. As we mentioned above, by involving regulated partners developers will manage responsibility.

Stakeholders also collaborate in certain aspects of the application development. In this regard, C3 mentioned: "…*for now, we have an agreement with medical device manufacturer… […] they are leader in this area... It is a third party protocol, third party interface. [They] have standardized in a requirements list…*" Collaborating with a third company with more experience could minimize the security issues of the application. This is because these parties could have in place procedures and methods to incorporate the required security requirements which could not be known by inexpert developers. Moreover, as C2 commented: "*I envision that we'll work together with a sizable third party, who have the brand, expertise, or the name of really having good security. And they can take the security issue away from us…*" This suggest that the lack of resources and security expertise influence companies to look for third parties with security expertise, such as security consulting companies, to analyze the security aspects of the application. Another option is to partner with companies whose brand generates trust among potential customers. Companies with experience in the field could be known as companies that produce secure products. Nevertheless, when companies work together with more experienced firms, developers would need to adjust their technology to the meet the partner's expectations. Experienced companies facilitate the integration of security into the application because they already have the knowledge and experience to handle security.

Some companies involved third parties which are more related to the legal aspects of the technology development. As C3 commented in this regard: "*we are in this process with the notified body… […] they have the requirements expectations, and we will make the testing with them. They will be our external testing and certification body for the EU….*" Companies could also consider third parties such as compliance and regulatory consulting companies, external testing, and notified bodies in the development process. By including these parties, companies aim to

ensure that their product will meet all the required rules and regulations. Implicitly, companies assume that the required level of security should be achieved by following all regulations and standards. Therefore, it seems that involving reliable stakeholders is the preferred method chose by companies to deal with security.

6.5.2 Incorporating security requirements when it is required

The following security requirements listed by a healthcare institution in the Netherlands illustrates an important characteristic of security. According to the requirement, *the application is only available to authorized users. Identification and authentication are based on username and password. Additional agreements on identification and authentication may be made now or in the future*. Security requirements and security mechanisms evolve over time. From the example, it is clear that changes in the required identification and authentication mechanisms are a possibility now or in the future. This is because, the threat landscape also changes, and new vulnerabilities are discovered every day and new targets, which were not attractive before, emerge as potential revenue sources for the bad guys. Besides, as C1 noticed: "…*we have learned a lot over time. When we started, almost 6 years ago, rules and regulations were not always as strict as they are today. Also, I would say, no much about hacking and stuff like that was on the radar… […] When we started security was the username and the password…"* Security is a dynamic process. Therefore, due to the dynamics of security, companies might opt for including security requirements when it is required. They could keep a grounded base, but additional security requirements will be accommodated in accordance with the security needs.

Moreover, as we have shown during this section, developers are more concerned with providing functionalities that meet users' needs. However, medical applications should comply with rules, regulations, and standards to be commercialized in the European Union and to operate within healthcare institutions. Some hospitals also could ask security requirements to safeguard their information systems. Then, developers need to incorporate the required security requirements into the application. In this sense, C1 commented: *"…if it is required by law, we will always make it because that is required in our market. Otherwise, we don't have a product."* Nevertheless, as C1 mentioned: *"we have a lot of requests, but we do not always immediately put them through. And the other thing is that there are development cycles... Development cycles go for quarters and I might want to have a new feature but it is not going to happen."* Companies need some time for adapting their development process, including the required security requirement or security mechanisms into the product and testing that the implemented countermeasures do not interfere with the product's functionalities. Thus, this situation reinforces the strategy of incorporate security requirements when it is required. First, focus on testing the product and provide the intended functionalities, and then, meet rules and requirements.

6.5.3 Adjusting technical features of the applications

The last strategy employed by developers is reflected in the technology. As explained by C2: "*We believe in a combination of hardware and software. So, building some parts in hardware and some parts in software, and not only doing in software. It is ways cheaper in software, but the problem is the security issue. Doing in hardware is really difficult to pull it down or to change and doing in software is really easy to change.*" By following this strategy, developers aim to hinder possible

actions to corrupt the signals of the sensors. In addition, as C3 commented: *"… the protocol that we are writing that we want to send per customer is a proprietary part, and which have the encryption and de-encryption on the receiver site. So, it is not a big problem*…" Developers try to protect their IP by using proprietary software. Finally, as C1 mentioned: *"There is a balance between usability, technical working the best, and security… So, it is about finding the balance. We twist things a lite bit, we test things sending to some customers saying this is what we build. Is this what you like? Too slow; too fast?"* More experienced developers could be aware that some features of the technology need to be adjusted to accommodate the required security.

6.6 Intervening conditions

The category intervening conditions facilitate or constrain the strategies adopted by developers to handle security. From our analysis, we found that *developers' perceptions, customers' needs, challenges faced during the development,* and *companies' experience* reinforce developers' strategies.

6.6.1 Developers' perception

It has been discussed in section 6.3 that developers' perceptions of security play a fundamental role as a causal condition for addressing security during the product development. Moreover, developers' perception regarding the impact that failures in security have in their business could influence developer' strategies. As C2 remarked: *"…security for us in the company is that if you fail to be secure in your device, you have a hard time trying to convince people otherwise...*" C3 confirmed this notion: "*I think one small incident can stop the whole business. We don't want to fail in that. We don't want to get in that…*" Although achieving a right level of security is a challenging task, not achieving security could have a serious effect on the company. Failure in security could damage the company's reputation, which may lead to stopping the business. Thus, these developers' perceptions facilitate the decision of involving more knowledgeable stakeholders in the products' development or partnering with companies which have earned a reputation in the medical field.

Furthermore, as C2 commented: "*It is a risk to make the security part yourself… […] it is like a company recommended themselves as the best… […] If we say, yes! Our security is good, it is not really secure, especially as a startup.*" It appears that the lack of security expertise weakens the confidence of companies for taking security in their hands. It is known that startup companies face a severe lack of resources. Then, for them, it could have more sense to focus on developing functionalities quickly rather than spending valuable time in trying to fulfill non-functional requirements such as security. Besides, developers could opt for accommodating non-functional requirements into the application when these requirements are needed.

6.6.2 Customers' needs

Customers are important actors during the development process. In this regard, C2 commented: "*It is all about the nurse if the nurse like our product and can use our product, we are happy. […] If the IT department is really happy with our security, but the nurse does not like our application or cannot use it, it does not matter how secure the application is, we have nothing.*" For

manufacturers, the customer is the person who will use the product, which includes medical personnel, patients, and some hospital departments. IT departments is not considered as the product's customers. Besides, as C2 mentioned: *"we start at the market. So, we asked a lot of specialists and nurses what the problem is. Why there is […] when you can prevent it… what is going wrong? They [medical specialists] say… [they explain the situation] So, based on that we have functionalities…"* Developers try to understand the problems that medical staff or patients are facing, and based on that; they intend to offer a solution that facilitates practitioners' work and patients' care. The solution must incorporate the required features to accomplish these tasks. Then, requirements for the application are derived from product's customers. According to developers, customers do not care too much about the products' security. Customers might assume that the company will take all the necessary measures to provide a secure product. Therefore, customers will ask for their preferred product' features, and it seems that security is not one of these preferred functionalities.

Product' customers and hospital's security officers could have different necessities. Customers expect an *easy-to-use* application that satisfies their needs. Meanwhile, hospital security officer demand applications that meet the hospital's security requirements. As explained above, developers focus on satisfying customers' needs. Thus, it seems that the strong focus on product's customers reinforce developers' strategies of incorporating security requirements when it is needed. Which might not necessarily mean that security needs to be addressed during the development process.

6.6.3 Challenges faced during the development

Technology development brings a series of issues to developers. As C2 commented: "…*like the false outcomes that are really important to check because you need to know how many times something goes wrong. In my opinion, the most problematic thing is the randomness of the situation. In a lab setting, you cannot test all the potential possibilities, and you cannot assess all potential threats.*" Uncertainties regarding the operation of the application could keep developers busy during the technology development. Thus, they focus on verifying that functionalities operate well. Furthermore, applications operating in the healthcare sector deal with human lives. In this regard, C3 mentioned: "*Safety is high priority… […] it is a battery device placed on the chest of the patient. So, it has to be IP67, and no water should flow. It should be able to be cleaned. It should not corrode, battery should not explode…*" Healthcare sector places an enormous value on patient's safety. Developers in this domain need to ensure that the application does not entail any physical risk for patients. Therefore, safety becomes a priority in the product development. Finally, regarding rules and regulations, C1 mentioned: *"…some markets demand rules and regulations very strict and some [markets] no. […] [application] is HIPAA compliance, the USA rules and regulations. And FDA compliance, which is more important than CE and our local regulations. Some features are developed twice: to our rules and regulations and their rules and regulations."* Rules and regulations differ per country. Thus, developers might focus on one market and then scale the product to comply with the rules and regulations of other potential markets in the future.

Technology uncertainties, providing safety applications, and the diversity of rules and regulations are factors that could require urgent attention during the development. Thus, these factors

influence developers' strategy of including security requirements when they are needed or adjusting the technology as necessary.

6.6.4 Companies experience

Lastly, developers experience could influence the strategies adopted by companies to address security. For instance, as C3 comment: "…*we don't want to get last minute surprises with the regulation body and notified body. It is worth spending more time upfront to find what is required rather than waiting or assuming something.*" Companies whose member have more practical knowledge on the technical or medical aspects of the field include reliable stakeholders early in the development process. Developers with less experience might involve additional stakeholders later in the development process. Besides, the lack of knowledge and experience regarding security requirements influence developers to incorporate security requirements when someone asks for those requirements. Or to partner with more knowledgeable third parties.

6.7 Consequences

Security is not at the forefront of stakeholders' concerns during the design and development of medical IoT applications. However, because of legal rules and regulations and hospital security awareness, developers need to take actions for incorporating security requirements into the application. Otherwise, applications cannot be sold in the European market. Involving knowledgeable stakeholders allows developers to tackle security during the application development and to cope with the perception of business' failure due to a security failure. Besides, incorporating security requirements when they are needed enables developers to focus on delivering functionalities that satisfy customers' needs, and that can be tested quickly in the market.

6.8 Discussion on generalization

During section 6.7, we stated the consequence of the identified factors. In this section, we aim to generalize our findings to a general population. For doing so, we compare our results with literature analysis. We look for similar studies and compare our results with those studies (Deutsche Forschungsgemeinschaft., 2000). Our findings suggest that security is an afterthought during the development process because developers are concerned with the functional requirements of the application. This idea of security as something that is added later has also been claimed by Mead et al. (2008) and Viega (2005). According to the authors, compared with functional requirements, non-functional requirements such as security have been an afterthought during systems' development. Our findings also agree with Giardino and his colleges. In a recently published paper, Giardino et al. (2016) found that during the software development, startup companies give a low priority to product quality. Quality aspects, which refers to non-functional requirements such as security, considered during the products' development are only geared toward usability according to the study (Giardino et al., 2016).

Besides, we found that developers' perception of security influence developers to address security requirements in the application. Perception is a human action that refers to the way in which something is regarded, understood, or interpreted (Oxford Dictionaries, 2017). Some

studies have considered human aspects in security. For instances, Werlinger, Hawkey, & Beznosov (2009) found human, organizational and technological challenges that security experts face within their organizations. Holmström & Sawyer (2011) focuses on the social construction of information systems requirements. The authors argue that developers often choose to ignore, and thus hide, the complexities of gathering requirements in order to simplify both the difficulties of their tasks and their relations with customers (Holmström & Sawyer, 2011). As these studies show, human actions are relevant during the security requirements process because humans make the decisions that shape technology.

Similarly, rules, regulations and standards and hospital security knowledge could be a motivator for addressing security during the design and development of an application or later before its commercialization. Rules, regulations, standards, and security knowledge are linked with responsibility. Developers are held responsible for the application to the authorities. Our findings agree with Flechais & Sasse. During the development of e-Science applications, Flechais & Sasse (2007) found that responsibility is a key motivator to address security. According to the authors, factors such liability and reputation affect the motivation for security during the development process. Besides, liability is the product of legal responsibility (Flechais & Sasse, 2007).

As results of this factors, it is likely to occur that developers involve knowledgeable stakeholders within the development process. According to Flechais & Sasse (2007), a socio-technical secure system design is possible when stakeholders such as security experts are included in the process. Security experts have knowledge about security concepts and principles, the understanding of the needs for security, insights into threats, vulnerabilities, and risk.

Finally, our analysis suggests that developers incorporate security requirements when they are needed. We have not found studies that support our claim. However, we also have found information that falsifies our result. If security requirements are not analyzed during the development process, it is likely to occur that developers incorporate security mechanisms at any stage of development or implementation. When security requirements are incorporated into applications during the development is a question that still needs to be answered.

6.9 Preliminary conclusion

In this chapter, we aimed to capture the factors that influence developers to address security. *Developers' perception of security* as a mean for maintaining customers, selling the application, or generating trust influence developers to incorporate security requirements in the application. Similarly, *rules, regulations and standards* and *hospital security knowledge* could be a motivator for addressing security during the design and development of an application or later before its commercialization. If developers do not comply with the existing regulation, they cannot place their products on the market. However, *perceptions of security* a constraint to usability or as a limit for bringing products to the market quickly could negatively affect the incorporation of security into the application. Likewise, *development approaches*, such as iterative and incremental agile approaches, is a key factor to underestimate security during the applications' development. Developers could choose different strategies to address security. *Involving reliable stakeholders* with security knowledge or more expertise could be the preferred option. In addition, companies

might opt for *including security requirements when it is required*. They keep a grounded base, but additional security requirements would be accommodated in accordance with the security needs. Finally, *developers' perception* regarding the impact that failures in security have in their business, strong focus on *customers' needs*, *challenges faced during the development process,* and *companies' experience* could reinforce developer' strategies of involving reliable stakeholders or accommodating security when it is required.

7

Comparing developers and SRE practices

In this chapter, we aim to find the differences between the practices recommended by the field of security requirements engineering and the practices adopted by developers to handle security requirements. Section 7.1 compares both literature recommended and developers' practices. Afterwards, in section 7.2, we discuss three issues we found during the comparison and preliminary results.

7.1 Comparing practices to handle security

Security requirements engineering recommended practices were obtained through an extensive literature review of SRE frameworks and methodologies. These recommended practices are summarized in section 2.1.2. Developers' practices to handle security requirements were obtained from developers of IoT medical applications. Developers' practices are described in section 5.2. For the purpose of the comparison, we only considered developers' practices that are categorized as *operational* according to our framework. This is because at this level, we find the main task (elicitation, analysis, management, and validation verification) that organized the data collection and data analysis.

The comparison of SRE recommended practices and developers' practices to handle security requirements is presented in Table 14. Practices are categorized based on four main tasks: *elicitation, analysis, management, and validation and verification*. These tasks were indicated in our interpretative framework for data collection and data analysis (see section 2.2). From left to right, these tasks are shown in the first column. The second column indicates the practices recommended by the SRE field within each of the mentioned tasks. The third column fulfillment (F) shows the degree of fulfillment of SRE recommended practices by developers' practices. "Y" symbolizes that developers' actions meet SRE recommended practices. "P" shows a partial fulfillment and "X" indicates that developers' actions do not meet the recommended practices. Finally, the last column shows the identified issues.

Task	**SRE recommended practice**	**F**	**Identified issues**
Elicitation	Identify relevant stakeholders including security expertise.	P	Involving stakeholder is a common practice during the product development. However, security expertise is not always part of the stakeholders.

	Agree on security concepts and principles.	X	During stakeholders' meetings and training, some security mechanisms are explained to customers. However, as there is no discussion of security during requirements' workshops, security terminology is not needed.
	Identify valuable assets	X	Valuable assets are not identified. Stakeholders and developers prefer to focus on functional requirements.
	Identify threats to valuable assets	X	As valuable assets are not identified, threats to these assets are also not identified.
	Identify or define security goals.	X	Valuable assets and threats to these assets are not identified early in the development process. Then, security goals are not defined either. Stakeholders desire to protect a valuable asset; however, functional requirements are the main focus in the application's development. Developers elicit, when they do, security requirements directly.
	Operationalize security goals into security requirements.	X	The step from security goals to security requirements does not occur in practice. Basic security requirements and security mechanism are elicited based on developers' expertise. Besides, security requirements are gathered from different sources. However, these security requirements are not the product of stakeholders' security concerns.
	Ensure the elicited SRs are not implementation mechanisms	X	Security mechanisms are also considered as security requirements. Developers do not find the difference between a security mechanism and a security requirement. Thus, the gathered security requirements could sometimes be implementation mechanism.
Analysis	Manage security requirements interactions	X	As only basic security requirements and security mechanism were elicited, there is not much interactions or conflicts among these requirements.
	Reconciling stakeholders' views for SR or confliction SR (conflict resolution).	X	The basic security requirements elicited do not necessarily represent stakeholders' views for security. Thus, no much conflict occurs regarding these requirements.
	Derive a set of security system requirements	P	The elicited security requirements are managed together with functional requirements. Additional security requirements might be gathered from different sources. Besides, a set of security system requirements could be generated during the product's development or when the product is ready for the market, as a final step.
	Categorize SR according to essential, non-essential, and system level requirements.	P	Security requirements are categorized according to essential and non-essential. Essential requirements could be the requirements needed to comply with rules, regulations, and standards.
	Perform a risk analysis to categorized requirements	X	Risk analysis of categorized required is not a common practice among developers. Moreover, when security requirements come from rules and regulations, it is assumed they are required.

	Check for ambiguities and inconsistencies.	/	It is not clear if this practice occurs during the development process.
	Derive a set of categorized security requirements.	P	Security requirements could be categorized according to developers' necessities. Requirements from rules, regulations, and standards could be prioritized. As mention early, this process can be done during the development process or when the product is ready.
	Document security requirements	P	Some hospitals could ask for the security requirements documentation. Then, security requirements might be documented, but it does not mean that requirements are documented early as part of the development.
Management	Reconcile interacting functional, non-functional, and security requirements. And engaging in negotiations to resolve conflicts.	X	If security requirements from rules and regulations interfere with functional requirements, developers might adjust their technology to the needed requirements.
	Arrive a set of system requirements	P	System requirements will include some security requirements or security mechanisms. However, over time security requirements are included in the application.
	Redefine system requirements into system specification, facts, and assumptions	/	It is not clear if this practice occurs in reality.
	Architect security mechanisms to fulfill security requirements	P	Rules, regulations, and standards include specific security mechanisms and security requirements. Besides, for developers, security requirements and security mechanisms are the same. The selected security mechanisms respond to market's needs rather than an analysis process.
Validation and Verification	Ensure the satisfaction of security requirements	P	When companies are applying to certifications, they need to provide evidence showing that the application meet the needed security requirements.
	Validate that security requirements satisfy set security goals	X	As security goal was not identified, security requirements satisfy rules and regulations.
	Verify that security mechanism fulfills security requirements	P	Security assessment techniques are used to verify that security mechanisms ensure the product's security.

Table 14: Comparison of SRE recommended practice and developer's practices

As we can see in Table 14, most of the developers' practices do not match or partially match SRE recommended practices. In the remainder of this section, we will explain the main differences be between SRE practices and developer' practices per task.

During the **elicitation phase**, there are *different stakeholders involved* in the requirements identification process but security expertise is not always part of these stakeholders. Besides, as we showed in section 6.4 and 6.6, developers employ development approaches which focus on

providing functionalities quickly, and stakeholders and developers place great value to satisfy customers' needs. Then, stakeholders and developers are primarily concerned with functional requirements and some non-functional requirements such as usability. Consequently, none of these actors (stakeholders or developers) express their security concerns early in the development process.

Moreover, *valuable assets and threats to these assets are not identified*. As security concerns towards an asset, what is called *security goal*, are not stated explicitly; security requirements are not elicited because there is not a security goal to fulfill. Nevertheless, developers tend to elicit basic security requirements or security mechanisms according to their experience. These *basic security requirements can be an implementation mechanism* which will constrain technical staff from choosing the most appropriated security mechanisms.

During the **analysis phase**, basic security requirements and security mechanism elicited by developers do not bring major problems or conflicts among stakeholders. This is because the elicited requirements will aim to secure straightforward feature of the application. Thus, deriving a *set of security requirements which are the product of the analysis of stakeholders' concerns is not part of the development process*. In addition, *prioritizing system security requirements* by conducting a risk assessment is an activity that does not occur in the development process either. Then, *there is not a set of categorized and prioritized security requirements* for the system to be.

During the **management phase**, developers do not necessarily arrive at a *set of system requirements which reconciles interacting functional and non-functional requirements*. However, developers might arrive at a set of requirements which includes some basic security requirements and security mechanisms based on developers' experience. But it does not necessarily mean that these requirements become fixed requirements for the application. According to the SRE, at this stage, developers should architect security mechanisms to fulfill security requirements. However, as we have seen during this section, developers, and also authorities, do not necessarily differentiate those two notions. Thus, developers architect security mechanisms directly at any stage of development.

For entering the market, however, applications should comply with rules, regulations, and standards to operate in medical institutions. Some developers could incorporate the required security requirements in the development process or later when the product is almost ready. Then, *security requirements could come from different sources*. Developers should manage the required security requirements and the interaction of these requirements with functional requirements. This aspect is not considered by SRE field.

During the **validation and verification** phase, security requirements could not be validated because security goals are not clearly established. However, to comply with rules, regulations, and medical standards, developers need to provide the evidence that the application meets the required requirements. Thus, developers try to ensure that the application meet the required security requirements. Security assessment techniques can be used to verify that security mechanisms ensure the product's security.

Finally, it seems that security practices to handle security requirements as described by security requirement engineering field are deficiently adopted by developers. Security requirements are

left for later phases. Thus, for stakeholders and developers, security is not a major consideration during the product's development.

7.2 Uncovering issues

In Chapter 5 we describe the actions that developers undertake during the development process. Even though we try to focus on how developers manage security requirements, it was evident that few actions were performed to consider and manage security during the development process. In Chapter 6, we identify factors that possibly explain developers' actions or decision regarding security requirements during the development process. Moreover, in this Chapter, section 7.1 compare recommended actions to handle security requirements and developers' practices. By analyzing the obtained information, we have identified three general issues that harm the security requirements engineering process. We named these issues: *incorrect assumptions regarding developers' motivation for addressing security, SRE methods do not match iterative and incremental development approaches, dynamic nature of security*. These issues were made evident while we conduct the data analysis and preliminary results and the comparison of practices to handle security.

7.2.1 Incorrect assumptions regarding developers' motivation for addressing security

Security requirements engineering methods assume that stakeholders bear in mind security concerns regarding valuable assets of the system to be. As mentioned by Haley et al., security needs arise when stakeholders establish that some resources (asset) involve in a system are of value to the organization. These resources can be tangible or intangible. Stakeholders logically wish to protect themselves from any harm involving this asset (Haley et al., 2008). Fabian et al. argue that stakeholders have different views concerned with functional requirements, non-functional requirements, and security requirements. According to the author, stakeholders can express security concerns at different levels of detail. They can express abstract security goals or more detailed security requirements (Fabian et al., 2009). Moreover, SRE methods assume that companies include security expertise during the requirements engineering process. As stated by Mead & Stehney, the SQUARE methodology is more effective and accurate when conducted with a team of requirements engineers with security expertise (Mead & Stehney, 2005). SREP methodology also assumes a level of security expertise to perform the nine steps required by the method (Mellado et al., 2006). Requirements engineers or another expert stakeholder will guide requirements meeting toward addressing security during the application development. However, from our analysis, we argue that security is not a major concern for developers during the development process. Developers concentrate more on customers' needs to provide features that meet these needs. Besides, stakeholders and developers do not express their security concerns regarding valuable assets. These two actors are focused on the technology development rather than in analyzing the security requirements of the application. Moreover, security expertise is not always part of the development process. Thus, we can argue that developers do not bear in mind security concerns of the system and security expertise is not always part of the development process as assumed by the SRE methods and frameworks.

Rules, regulations, and standards can motivate developers to take security requirements into account in the development process of later as a final stage. Medical devices cannot be placed

on healthcare institutions if these devices do not meet the required security requirements. Besides, applications which process personally identifiable information and patients' data need to comply with specific information security requirements (see section 6.4 for more information). Therefore, incorporating security requirements into the application could be a rule and regulation driven activity rather than a developer or stakeholder' initiative. Although developers' perception of security as a mean to maintain customers or as a selling point could encourage developers to meet the required security, it does not necessarily imply that security requirements have to be analyzed early in the development process.

7.2.2 SRE methods do not match iterative and incremental development approaches

Security requirements elicitation, analysis, and management can be considered as a linear process. Although validation and verification of requirements are needed at the different stages of the security requirements engineering process, SRE follows a waterfall model. This means that tasks of each phase need to be completed before going to the next phase. For instance, to move from the elicitation phase to the analysis phase, stakeholders need first to arrive at a complete set of security requirements. Only then they can move to the analysis phase. On the other hand, developers employ development approaches such as agile were new or evolving requirements are specified in parallel with, or after requirements are already implemented (Jøsang et al., 2015). Due to the fast-moving nature of iterative and incremental product development process, we can argue that security requirements engineering methods do not match new product development approaches.

According to the Agile Alliance, iterative and incremental product development approaches promote adaptive planning, evolutionary development, early delivery, and continuous improving (Agile Alliance, 2015). We could see some of these characteristics in the companies' product development approach. Developers of Company2 and Company3 employ an evolutionary development which aims to continue improving the product. They started with features that change and evolve in accordance with customers' needs, technology availability, and context of the operation. Developers build an initial prototype of the application and iteratively refine the prototype over time. Stakeholders' meetings, application's testing in real settings, and customers' feedbacks help developers to refine the product and to improve its functionalities. In such a flexible development process, fixed security requirements engineering methods which require foreseeing threats to valuable assets, which might completely change or evolve, are less compatible. Then, developers opt for focusing on the functional features. After all, it is possible that initial functionalities have evolved to a point in which they are not recognizable anymore. Once functionalities are fixed, developers could consider additional quality requirements such as security.

7.2.3 Dynamic nature of security

From the data analysis, it was clear that security is not a static issue; it is dynamic. Thus, building applications that include all the required security requirements to ensure security are very difficult. Besides, security requirements are also dynamic. As discussed in section 6.5 (incorporating security requirements when it is required), security requirements and security mechanisms change over time. This is because the threat environment also evolves, systems become

attractive or profitable for the bad guys, new vulnerabilities are discovered, and new technologies for systems protection and systems attack are available.

The dynamic nature of security is also evident in the regulation. Rules, regulations, and standards need to be adjusted continuously to meet the required level of security. For this reason, developers might opt for maintaining a grounded level of security and complying with needed rules and regulations. Developers are aware that more security requirements will be needed in the future. Therefore, they will incorporate additional security requirements and security mechanisms when it is needed.

Although security requirement engineering methods provide guidance to build more secure technologies by incorporating security requirements early in the development process, a step forward is needed to tackle the dynamic nature of security. Validation and verification phases should also be performed after the product is developed. These processes will allow developers to update security requirements, security mechanisms, and security goals of a system

8

Discussion

In this section, we discuss the relevant implications that emerge from the comparison of SRE recommended practices and developers practices for handling security requirements during the development of IoT medical applications in Chapter 7, and from the factors that include development Chapter 6. Section 8.1 discusses the implications for the security requirements engineering process. Section 8.2 discusses the implications of our results to developers.

8.1 Implications for security requirements engineering process

Security research is an ongoing activity that requires a close relationship between researchers and industry. As security and the threat landscape evolves, security research also needs to evolve. However, to discover new information and reach an understanding on the security issues that society might face, the possible triggers of threats, the roots of the security problems, and means to deal with these issues; security researcher requires information from the actors that face the security issues, and that might also be part of these issues. Researchers and industry need to collaborate for achieving to a secure development. Researchers and developers have different incentives, motivations, and aims in their work. Then, it is required to foster understanding among these two groups by finding the things that they have in common rather than the differences. In this section, we discuss some implications of our results to security requirement engineering process.

Our findings lead to some important implication for security requirements engineering frameworks. First, in the case of the practices adopted by developers to handle security requirements in the IoT medical applications' development, it seems that security requirements as described by security requirement engineering field are deficiently adopted by developers. Our findings suggest that security requirements of IoT medical applications might not be the product of analyzing stakeholders' security concerns as suggested by the literature of SRE. It seems stakeholders do not explicitly indicate their security goals, and thus, security concerns are not operationalized into security requirements. Besides, security mechanisms are architected without clear security requirements. Developers might elicit security mechanism based on their experience. The field of security requirements engineering has been working around fifteen years (Dubois & Mouratidis, 2010) to expand the body of knowledge on techniques, framework, methodologies that facilitate the management of security requirements early in the development process (Mellado et al., 2010). However, security requirements, and thus security, are not be considered early in the development process. This suggests that SRE field need to find ways to

target their intended users. SRE methods and frameworks cannot help to improve security if developers are not applying these methods. The limitations of our study (such as the number of companies interviewed, biases of interviews, research methodology, and focus on one country) could hide a greater adoption of SRE recommended practices. However, it is still not clear if developers are using SRE tools. For instance, none of the small companies interviewed for this research have even heard about any of the SRE frameworks and methodologies. Besides, we do not find studies that provide insights regarding the adoption of SRE frameworks. Thus, research opportunities are open in this area to gain knowledge about the preferred SRE methods among developers and barriers to adopt such as methodologies.

Second, SRE emphasizes the differences among security goals, security requirements, and security mechanisms. However, it seems that developers and authorities do not distinguish the differences between these concepts. Security goals are a general statement about the security of an asset (Fabian et al., 2009). Security requirements operationalize one or more security goals (Fabian et al., 2009; Haley et al., 2008). Security mechanisms satisfy security requirements (Firesmith, 2004). The objective of this distinction is to understand security properties and to move from abstract security goals to concrete security requirements and mechanisms (Fabian et al., 2009). In practice, there is a slight difference between these concepts. However, by understanding the why (security goal) of the security requirement before the how (mechanisms) developers could be able of eliciting security requirements for their applications. Otherwise, it could be difficult to carry out the ordered SRE process. Developers can easily understand these differences. However, security requirements engineering could start explaining the information required to facilitate the application of the methods or frameworks.

Third, SRE methods assume that stakeholders bear in mind security concerns regarding valuable assets of the system under construction (Fabian et al., 2009; Haley et al., 2008). Also, SRE assume that companies include security expertise during the requirements engineering process (Mead & Stehney, 2005; Mellado et al., 2006; Paja et al., 2014). Nevertheless, our findings suggest that stakeholders and developers do not necessarily express their security concerns regarding valuable assets. Developers and stakeholders focus on developing functionalities that satisfy customers' needs as fast as possible. Besides, development approaches also foster this tendency. Giardino and colleges found a similar path during the software development. According to the authors, evolutionary approaches speed-up development process. During the development proces of IoT medical applications, rules, regulations, and standards might motivate developers to incorporate security into the application. However, it does not necessarily mean that developers will analyze require security requirements during the development process. Flechais & Sasse (2007) found that responsibility is a key motivator to address security. According to the authors, factors such liability and reputation affect the motivation for security during the development process. Besides, liability is the product of legal responsibility (Flechais & Sasse, 2007). Thus, as companies are held liable/responsible for the outcomes of security failures, these responsibilities motivate them to consider security requirements during the development process. Although rules, regulations, standards could help authorities to motivate developers for addressing security during the development process, these actions are reactive. Thus, developers actions appear as a response to external stimulus. It is socially desirable that developers include in the development process values that are relevant for society (such as security) because they understand the

importance rather than as an obligation. A final thoughts regarding the role of rules, regulations, and standards, technology has rapidly evolved to arrive a the revolution of the internet of things. Are rules, regulations, and standards able to catch up with this rapid growth? Which aspects of the IoT rules, regulations, and standards should take into account? One size fits all is enough to treat applications that generate medical data? What happens with IoT applications that produce health-related data but are not part of hospitals' information systems? Does regulation also protect this users' health-related data? Legal frameworks and policies need to be created for addressing this question. This opens new research opportunities to identify which aspects of the IoT needs to be considered to develop legal frameworks that guide developers on to deal with security and at the same time take into account aim to protect users. In addition, SRE methods and frameworks should start developing some guidance to make methods and frameworks more understandable to developers who are not experts in security.

Next, it was already mentioned that development approach fosters a rapid technology development. New development approaches aim to deal with uncertainty by testing functionalities quickly in the market. For instance, agile development approach focusses on adaptive planning, evolutionary development, early delivery, and continuous improving (Agile Alliance, 2015). Meanwhile, SRE methodologies describe a linear process that needs to be planned and constrain evolutionary development and continuous improving. Otherwise, how can developers protect valuable assets if those assets might change during the development process or how can developers integrate functional, non-functional, and security requirements into a set of system requirements if functional requirements vary and evolve all the time. It appears that these two visions need to find a common basis for working together. How to integrate these two contradictory ideas is one of the main challenges for the SRE literature. Mead et al. (2008) combined these two visions. The authors aimed to incorporate security requirements engineering into dynamic systems development method (Mead et al., 2008). Nevertheless, field studies are needed to analyze if developers are adopting this methodology. And which are the main challenges and advantages in applying SQUARE into dynamic process development. Besides, SRE field should develop frameworks and methods that naturally fit incremental of a dynamic development process. In the same way that this development process fosters evolution of functional requirements, SRE methods could star eliciting some initial security requirements which might change or evolve during the development process.

Finally, the dynamic nature of security needs to be considered in SRE frameworks and methods. It is widely accepted that to build applications that include all the required security requirements to ensure security is a difficult task. Vaughn et al. (2002) argued that security requirements are dynamic for various reasons. Security solutions depend on threats against the system, likelihood of threat to be exercised, the state of technology available for system protection, the state of technology for system attack, perceived value of asset's information. Besides, security solutions need to be developed to defend against most likely threats, act dynamically, and deal with the changes of threats against (Vaughn, Henning, & Fox, 2002). Thus, security requirements and security mechanisms change over time. This is because the threat environment also evolves, systems become attractive or profitable for the bad guys, and new vulnerabilities are discovered. Then, the question becomes how SRE methods might help developers to include new security requirements even though the product is already developed. SRE should also include an analysis

post development and service operation. Security requirements and security mechanisms need to be updated to provide the required security. This opens another opportunity for research. How to create and evaluate SRE methodologies that cope with the dynamics of security.

8.2 Implication for developers of IoT medical applications

Incorporating security requirements into technology is a complex activity that requires developers and managers' commitment to developing secure applications. People play a fundamental role in developing secure technologies because people decide which functionalities a product should offer. Developers choose which requirements (functional and not functional) they include into the application. Besides, it is known that technology shape society, but it is people who shape technology. Then, developers need to be more aware of the threats that IoT applications can bring to society. Failures in security due to vulnerabilities do not only affect users and the internet. It also affects companies' reputation which can lead to financial consequences and even to stop the business. Thus, although developers have different priorities due to various stakeholders, complex process of development, market uncertainties, and others; security needs to be enforced and promoted as part of the culture of companies. In the reminder of this section, we discuss some implication of our results to developers.

Our findings lead to implications for developers. In the case of developers' practices to handle security requirements; our results suggest that, at the operational level, developers (engineers, designers) engage in a few actions to address security during the development process. Besides, at a managerial level, it seems that there are not administrative actions or decisions to encourage a secure development. For instance, none of the interviewed companies had a security policy to provide guidance on how to deal with security during the development process. Policies are expressions of management intent and organization' principles. These will influence and guide decisions making. Policies are important in order to build secure systems because engineers should find in these documents the intent of the organization relative to security. Policies may contain a description of what assets the organization believes need to be protected, and responsibilities to implement that protection (Vaughn et al., 2002). In addition, developers' actions take place in an organizational context. The context of development, the firms' environment, and human activities play a fundamental role to develop the secure product. Without an environment that motivates to produce secure products within companies, developers might underestimate the role of security during the technology development. Then, work still needs to be done to find ways of encouraging a secure development environment within companies. Researchers could focus on the interactions between the context of development, firms' culture, and humans' perception.

Regarding how developers handle security requirements of the technology, our finding suggests that developers opt for applying different strategies. Developers might prefer to involve partners who are more knowledgeable, partners who have more experience in the field, or partners who have a reputation as trustworthy companies. Besides, developers might opt for incorporating security requirements when these requirements are required. It seems that requirements are required when developers need to provide evidence showing that the application meet rules, regulations, and standards. A research project which collects data by being part of the development activities might yield different strategies that were not evident during the interviews. A cooperative development might be beneficial to further the evolution of the technology. For

SME, it might be difficult to address issues for which they are not prepared. Then, expertise in different aspects, such as security, could be incorporated to speed-up the IoT development and adoption. By doing so, developers might offer applications that security meet customers' expectations. Thus, partners specialized in security might become part of the IoT ecosystem.

Besides tackling security early in the products' development process, security requirements engineering provides an extra benefit to developers. Discussions between stakeholders and developers about security concerns, valuable assets, threats, security requirements, and security mechanisms raise awareness about security matters to developers and stakeholders (Flechais & Sasse, 2007). Moreover, as a consequence of the discussions, actors can better understand the characteristics of their applications and the environment where it will operate. Thus, more informed decisions during the applications' development can be taken. However, when security requirements are not the product of analyzing security concerns, the discussions might not occur. Consequently, stakeholders and developers miss a fundamental aspect of the product development: understanding the properties of their applications and the threats that might affect the application in their operational context. The development process should encourage a discussion on the values which are not part of functional requirements. For instance, security is an aspect that should be understood before a product is developed. Besides, values that are important to society, and which are considered from the beginning in the development process, might facilitate the acceptance of the technology. These discussions also might help developers to deal with the different challenges they faced.

Finally, regarding the development of medical IoT applications, our findings suggest that SME, especially startup companies, might start designing and developing a prototype version of the smart device with limited connectivity. After functionalities are accepted by potential users and medical institutions, developers will move to the next phase by connecting the devices to the internet for expanding their capabilities. However, as security requirements for connecting objects to the internet were not considered in the product development, initial versions of the product might not include any security mechanism for being networked. Connecting devices that lack of minimum security requirements to be networked is a well-known issue in the security field. Thus, actions might be taken to avoid continuing in past mistakes. Vaughn et al. (2002) argue that in order to understand the security of a networked system, an analyst must look at the entire system – not just a particular set of technologies or components. Developers of IoT applications need to consider the security requirements of their applications as part of a bigger system. Smart devices will interact with other devices and new behaviors will emerge as a consequence of the system capabilities. In addition, accommodating security requirements for interconnecting objects to the internet as the latest stage of development might delay the development of IoT applications. Functional requirements and non-functional requirements might conflict with security requirements, and thus, applications' functionalities might need to be adjusted to fit security needs better. The internet of things calls for new approaches to think about security, security that does not only consider developers' own applications or devices but also the new security challenges that interacting devices might bring to society. Tools that capture these intrinsic difficulties are still a challenge for the security research community.

9

Conclusion and recommendations

In this chapter, we provide an answer for our main research question *how do developers' security practices in the design and development of IoT medical applications challenge security requirements engineering methods?* Section 9.1 provide the answer to the sub questions which leads to answer the main research question. In section 9.2, we elaborate on the limitation of the research. Section 9.3 provides recommendations for finding empirical data, developers of applications and security requirements engineering field. Finally, section 9.4 concludes with future work.

9.1 Main findings

Sub question 1: *Which security practices are described in the literature of security requirements engineering?*
The process of requirements engineering can be decomposed into four tasks: *elicitation, requirements analysis, validation and verification, and requirements management.* The **elicitation of security requirements** involves identifying the requirements that the system must satisfy to achieve the system's security goal. During requirements identification meetings, developers should *include stakeholders* with knowledge about the system and knowledge of security. Stakeholders and developers need to *agree on definitions regarding security*. After that, *security goal* should be identified or defined by *determining valuable assets and threats* to these assets. Detailed *security requirements are elicited* to operationalize the security goals. Finally, developers should *ensure that the elicited security requirements are not architectural constraints*. Requirements specify what is needed from the system rather than the architectural mechanism.

The **analysis of security requirements** allows practitioners to better understand requirements, their interrelationships, and their potential consequences. From the elicited security requirements, a set of *Security System Requirements needs to be derived*. For doing so, practitioners should analyze and manage security requirements interactions, *reconcile stakeholders' view regarding security requirements*, and *engage in negotiations to solve conflicts*. Following that, a *categorized set of security requirements* should be derived (essential, non-essential, and system level requirements). Then, a *risk analysis* should be performed to identify the most important requirements. After arriving at a set of categorized security system requirements, practitioners need to check for ambiguities, inconsistencies, and mistaken assumptions. Lastly, *security requirements are documented*.

The **management of requirements** comprises various tasks related to the handling of requirements, including the evolution of requirements over time. Categorized security requirements should be *integrated with functional and non-functional requirements* to arrive a consistent set of System Requirements. *Conflicts between functional requirements and security requirements* should be solved by engaging in negotiations. After that, the consistent set of system requirements is redefined into a systems' specifications, facts, and assumptions about the environment where the system will operate. Specifications constrain the system to be built, while facts and assumptions describe (or constrain) the environment of the system. Then, *security mechanisms are architected* to fulfill one or more security requirements, and thus, to reduce one or more security vulnerabilities.

The **requirements validation** ensures that models and documents accurately express the stakeholders' needs. Practitioners must *validate that the security requirement* satisfies the intended security goal. Validation of requirements for completeness, lack of conflicts, or other matters is needed alongside the elicitation, analysis, and management of security requirements. Verification techniques are used to prove that *system specification meets the security requirements*

Sub question 2: *What insights can we gather regarding how developers elicit, analyze, and manage security requirements during the IoT medical application development?*
Data obtained from three companies developing IoT medical applications suggest that, in practice, small companies do not have a distinctive process to handle security requirements. When security requirements are identified, if they are, it is alongside functional requirements. The **elicitation of requirements** involves different stakeholders. C3 *included compliance and regulatory consulting companies and notified bodies to take regulatory requirements into account*. In addition, third parties working with the companies should comply with the existent rules, regulations, and standards. Only *C1 has included a security officer as part of their stakeholders*. When it is required, companies *gather security requirements from different sources*. Rules and regulations, company's security expertise, hospital IT personnel or hospital security officer, customers, third parties' security expertise are the most common. Developers refer to security concerns and security mechanisms as security requirements.

The **analysis of requirements** involves the *prioritization of functional requirements*. The first criteria to prioritize requirements is functionality. In the case of C1, when the application is already developed and ready to be placed on the market, *developers prioritize security requirements based on rules and regulations*. After that, *hospital security requirements are included* according to the financial consequences. In addition, C1, engage in negotiations to gain time for incorporating the needed requirements.

The **management of requirements** involves *incorporating security mechanism to the application*. C3 considered security requirements from regulations as part of applications' requirements. Otherwise, known security mechanisms will be used to safeguard assets or features, which developers consider important. When the application is already developed, as in the case of C1, developers will *include the security mechanisms required by law or by hospitals*. Besides, during the application's training program, C1 explains to users how to employ the application and the security mechanism.

The **validation and verification of requirements**, for C2 and C3, involve *ensuring that the inputs and measures obtained by the application are correct*. Data transmitted between transceiver and receiver is also verified. Finally, developers need to *provide the evidence to regulatory authorities* to make sure the application meets all the required rules and regulations.

Sub question 3: *In practice, which factors influence developers to accommodate security in the IoT medical application?*

Our analysis suggests that *developers' perception of security, rules and regulations, hospital security knowledge, and development approaches* could be considered as causal conditions for addressing security. These factors lead to the adoption of strategies to incorporate security requirements into the application. *Involve reliable stakeholders*, *incorporate security requirements when it is required*, and *adjust technology features* are some of the strategies. In addition, factors such as *developers' perception, customers' needs, challenges faced during the development, and companies experience* could be considered as intervening conditions which facilitate or constrain the adoption of the developers' strategies.

Developers have different perspectives on security. Perceptions about security as a *mean to maintain customers* or as a *selling argument* or as *a mean to generate trust on the application* influence developers to incorporate security requirements into the application. Less positive connotation such as *security constraints usability* and *security limits the company for bringing products to the market quickly* negatively affect the incorporation of security requirements into the application. **Rules, regulations, and standards** and **hospitals' security knowledge** could become key motivators for addressing security during development of an application or later before its commercialization. Medical devices should bear the CE Mark to indicate their conformity with the provisions of the European's rules. Besides, medical applications that process personally identifiable data are regulated by the Personal Data Protection Act. In the Netherlands, companies need to comply with the standard NEN7510 for information security in care. To start operating within healthcare institutions, products should meet hospitals' requirements. **Development approaches**, such as iterative and incremental agile development process, can be considered as a key factor to underestimate security during the applications' development. This type of approaches places the greatest importance to develop functionalities quickly. Then, security is considered in the latest phases of development or when it is required.

Developers' perception regarding the impact that failures in security can have in the companies' business facilitate the decision of involving more knowledgeable stakeholders in the products' development or partnering with companies which have earned a reputation in the medical field. Failure in security *damage the company's reputation*, which may lead to stopping the business. In addition, lack of security expertise weakens the confidence of companies for taking security in their hands. **Customers' needs** play a significant role in developer's decisions and the development process because requirements derived from product's customers. *Usability* is a fundamental feature that applications should offer to product users. The focus on the product's customers reinforces strategies of incorporating security requirements when it is needed. Among the **challenges faced during the application's development**, *market and technology uncertainties*, *providing safety applications*, and the *diversity of rules and regulations* are factors that require urgent attention during the development. Thus, these factors influence developers' strategy of including security requirements when they are needed or adjusting the technology as

necessary. Finally, findings suggest that **developers experience** influence companies' strategies to address security. Companies whose member have more knowledge on the technical or medical aspects of the field include reliable stakeholders early in the development process. To generalize our main findings to a general population, we have compared our results with results of similar studies

Sub question 4: Which are the differences between SRE recommended practices and developers' security practices in the development of medical IoT applications?
Most of the developers' practices do not match or partially match SRE recommended practices. During the elicitation phase, *stakeholders and developers do not express their security concerns regarding an asset*. During the development, actors are primarily concerned with functional requirements and some non-functional requirements such as usability. As *security goals* are not stated explicitly, *security requirements cannot be elicited because there is not a security goal to fulfill*. Nevertheless, *developers tend to elicit basic security requirements or security mechanisms* according to their experience. These *basic security requirements can be an implementation mechanism* which could constrain technical staff from choosing appropriated security mechanisms.

During the analysis phase, deriving a *set of security requirements which are the product of the analysis of stakeholders' concerns is not part of the development process*. In addition, findings suggest that *prioritizing system security requirements* by conducting a risk assessment is an activity that do not occur in the development process either. Then, *there is not a set of categorized and prioritized security requirements* for the system to be.

During the management phase, developers do not necessarily arrive at a *set of system requirements which reconciles interacting functional and non-functional requirements*. However, developers can arrive at *a set of requirements which includes some basic security requirements and security mechanisms based on developers' experience*. In addition, developers and authorities do not differentiate security requirements and security mechanisms. Thus, security mechanisms are elicited directly at any stage of development. For entering the healthcare market, applications should comply with rules, regulations, and standards. Then, *security requirements could come from different sources*. Developers should manage the required security requirements and the interaction of these requirements with functional requirements. Security requirements demanded by authorities are not be considered by SRE field.

During the validation and verification phase, security *requirements cannot be validated because security goals are not clearly established*. However, developers need to provide the evidence that the application meets the required legal requirements. Security assessment techniques can be used to verify security mechanisms.

Main question: How do developers' security practices in the design and development of IoT medical applications challenge security requirements engineering methods?
Developers' practices to handle security requirements challenge SRE recommended practices in the following aspects. First, small companies do not have a unique process to handle security practices. Thus, the organized process of eliciting security requirements, analyzing requirements, managing requirements, and validating and verifying requirements do not occur. Second, companies elicit some security requirements and security mechanisms during their standard

development process. As security goals are not identified, the elicited security requirements or security mechanisms are not the product of the analysis of stakeholders' security concerns. Third, during the development process or when the product is almost ready, developers gather security requirements from rules and regulations, hospital requirements, and third-party security requirements. Besides, the collected requirements are categorized based on urgency. Fourth, the management of requirements is mostly concerned with incorporating security requirements, from regulations or customers, into the applications. Security mechanisms are architected based on what is already required (from regulations), or these mechanisms respond to customer's needs. Lastly, verification of security requirements is needed to enter the market. Companies should provide evidence that the application meets regulatory security requirements to get into the market.

Findings suggest that these differences occur because SRE recommended practices assume that stakeholders bear in mind security concerns regarding valuable assets of the system to be, and companies include security expertise during the requirement engineering process. However, among the factors that motivate developers to address security are *developers' perception of security, rules and regulations, hospital security knowledge, and development approaches*. *Developers' perceptions of security* as selling point or feature to provide trust to customers; *required rules, regulations, and standards* to place the product on the market; and *required hospital security requirements* to introduce the application into a hospital positively influence developers to incorporate security requirements into the application. Meanwhile, *developers' perceptions of security* as a constraint to usability or a limitation to bring products to the market quickly might delay the decision of incorporating security during the development process. In addition, SRE process takes security requirements into account early in the development process. However, in practice, developers employ different strategies to incorporate security requirements into the application. These strategies do not necessarily mean that security requirements are included early in the development process. One preferred strategy to address security is to *incorporate the required requirement into the application when these requirements are needed*. By following this strategy, developers can concentrate on functional requirements to satisfy customer' needs. This strategy also helps developers to deal with the dynamic nature of security. Security, security requirements, security mechanisms are dynamic and evolve over time. However, it seems that SRE does not provide further guidance to address new security issues that could affect the system. Finally, SRE process does not consider the impact that factors such as *developers' perception of security failures, customers' needs, challenges faced during the development, and companies' experience* have over developers' strategies.

9.2 Limitations

The study has some limitations which are discussed in this section. First, as it was already mentioned, access to empirical data regarding developers' security practices is one of the main limitations. Companies developing medical IoT applications or IoT-enabled medical devices were not interested in participating in the study. According to the companies, the reasons are the lack of time or unavailability of personnel for the interview. Although the companies who took part in the study gave us an interesting broad picture of the practical issues of managing security requirements during the design of an application, more participants could complement this picture

and provide new data which supports or challenges our findings. More participants could provide insights that have important implications for our findings. A bigger sample of companies developing IoT applications will allow as to generalize our findings. By following a comparison with similar studies, we generalized some of our results. However, more data will allow us to draw general conclusions about factor such as *adjusting technological features* as a strategy to deal with security; or *challenges faced during development*, *developers experience* as intervening conditions that shape the strategies adopted by developers. As it is known in the research field, the more cases we have analyzed the more general the conclusions we can be (Deutsche Forschungsgemeinschaft., 2000). Besides, our limited empirical data could hide insights regarding the role of managers during a secure development process, actions undertaken by engineers and designers to incorporate security requirements, stakeholders' security concerns during the development, and methods and tools employed to address security in the development process. In addition, the interviewed companies are small and medium enterprises which are starting in the technological development process. Although we have learned valuable lessons from small innovators, it is required to incorporate visions and insights from big companies who have experience in the development of healthcare technologies. These big companies also should be working to develop medical applications that could be considered as IoT.

Second, the stage of development of IoT applications brings another limitation. Our focus was the development process because we wanted to observe how developers manage security requirements during this process. However, as companies were first developing the IoT-enabled device, we cannot see the challenges that they face during the connection of the devices to the internet. We could notice that developers prefer to tackle the security implications of interconnecting de devices to the internet in the future when they actually connect the device rather than early in the development process. But we could not gain new knowledge of the main issues regarding security requirements during the connection with the internet or how this integration affect functional and non-functional requirements of the application. The process of developing technology requires time because of the different variables that interfere and are interfered during such a process. Then, longitudinal studies could be performed to understand how developers deal with security requirements during the development lifecycle from the requirements elicitation to operation and maintenance of the medical IoT application.

Third, data collection methods employed during this study bring another limitation. We conducted interviews to developers and manager and reviewed some documents provided by developers. The aim of these methods was to collect information on how they deal with security requirements during the technology development and the factors that affect their decision. Although we aimed to participatory observe developers' behavior, participants were not willing to allow outsiders to join in their meetings. Interviews suffer from bias because people tend to overestimate what they do or say in these meetings. Then, participants tend to provide responses that are socially accepted rather than the representation of real events (Yin, 2014). As it has been noticed, there is a difference between what people say they do, what they intend to do, and what they actually do (Päivärinta & Smolander, 2015). This issue could be especially relevant in a study of security, participants might prefer to say that they take into account security aspects of the technology rather than to admit that they have not considered security. Although, in this study, developers could overstate their answers; we think that we were able to get close to the reality that companies

face during the development process. We make this claim because our results do not show extremely complicated efforts to address security during the development process. What we think is closer to reality. Nevertheless, the study of practices requires more than interviews. Researchers might need to observe the behavior of developers in their real settings to assess practices better.

Finally, there is an immature IoT ecosystem for medical applications. Although there is a strong incentive for companies to develop IoT medical solutions, the ecosystem is still immature. Few small businesses are putting their efforts in this area, and large enterprises in the healthcare domain are still working on their applications or IoT platform. For instance, we cannot get access to the IoT platform for healthcare applications of one of the key medical device manufacturers because the platform is not ready to be used yet. Although this volatile ecosystem gives opportunities to study different aspects of the technology development such as the emergence of ecosystems, technology battles for dominance, networks dynamics among others; it also constrains the quality and quantity of empirical data available. Our results represent only a small version of a major problem.

9.3 Recommendations

In the Chapter 8 (discussion), we provide some recommendations to SRE methods and frameworks and developers. In this section, we summarize the main recommendation based on what we have learn during this thesis project. Due to the focus of this research, we focus on recommendations for accessing to empirical data regarding security, and recommendations to developers of medical IoT applications and security requirements engineering field.

9.3.1 Gathering data regarding security

During this research project finding empirical data regarding security practices was a major challenge. Although there is a call for more empirical field studies regarding security, there was not much interest from companies developing IoT medical applications to our research. Developer's lack of time and availability, the reputation of companies as secure developers, and desire to protect IP of the IoT application can influence developers to reject the invitation. It is known that accessing to data regarding security can be challenging can be challenging. In their research, Werlinger et al. (2009) faced a similar limitation to gather data. Initially, the authors aimed to use work shadowing[2] and contextual interviews to collect fine-grained data on the work of security practitioner. However, none of the contacted practitioners were willing to engage in these activities. We believe that a collaborative effort between academia and industry will help researchers to access data related to security. Besides, researchers could work with organizations that manage different companies developing their technology. For instance, business incubators, accelerators programs, and consulting companies might provide the required environment for communicating with developers regarding security and security requirements. It is important, nevertheless, to find mutual benefits for companies developing the

[2] Work shadowing refers to the activity of spending time with someone who is doing a particular job so that the observer learn how to do it (Cambridge Dictionary, 2017).

technology, organizations that aim to foster innovation and academic researchers. Companies and developers could understand better phenomenon or situations that they are facing and gain knowledge of tools that academia is developing. Third parties who aim foster innovation and consulting companies could gain knowledge to promote a secure development. Reputation as an organization that tries to foster a secure development also could also be an incentive. Researches could benefit from accessing companies to gather data. Being part of the organization could help researchers to perform participatory observation. Developers practicing in their natural setting provide valuable insights on the real security practices that might be exaggerated during interviews. It is worth to mention that it should be a commitment between developers and researchers to work together toward a secure development. Otherwise, more challenges that benefits might appear. For instance, developers can act in a certain way when researchers are around. On the other hand, researchers could frame information in orders to fit with their hypothesis.

Research strategies could also be beneficial to collect empirical data. Design research, for instance, could provide important insights on the challenges of incorporating security into the application. In design science, knowledge and understanding of a problem and its solution are achieved in the building and application of the designed artifact (March & Storey, 2008). Therefore, researchers could learn first-hand how developers deal with security requirements while providing attractive and useful applications. Researches develop an artifact which represents an IoT application. By developing this IoT medical application developers could gain knowledge of the complicated task of including security early in the development process. Besides, researchers are able to compare artifacts that early include security requirements with artifacts that employ different approaches. Methods for tackling security during the development process could be tested in this artifact. Methods could be refined to provide useful tools. Although by taking this approach developers might miss the contextual factors that influence developers, this solution is technically viable.

9.3.2 Security requirement engineering field

Existing methods and frameworks for managing security requirements should be adjusted to provide security requirement engineering approaches that better meet small companies' needs during the IoT applications development. Among techniques, frameworks, processes, and methodologies to facilitate the management of security requirements, Mellado et al. (2010) found twenty-two initiatives to develop more secure applications. Thus, work could focus on refining security requirements engineering methods to take into account the aspects listed below. Another option is to develop new security requirements engineering methods that better cope with the needs of the Internet of Things.

- Developers and stakeholders do not necessarily express their security concerns regarding assets of the systems. Nevertheless, they have interest which are affected if a security failure occurs. Then, the first stage will be work with worst case scenarios to gather the reactions of developers to these cases.
- Security expertise might not be part of the development process. Whether to require a security expert for carrying out the requirements process is a question that the SRE field should answers. Nevertheless, what it can be done is to provide explanations to better

understand the differences between security goals, security requirements, and security mechanisms. Engineers, designers, and managers could engage in identifying security goals that the systems must meet even though they do not have security expertise.

- After security goals have been established, and thus, there is an understanding of the security issues that might affect the application. Developers could consider incorporating security expertise for helping them to come up with security requirements and security mechanisms.
- SRE methods for developing IoT applications should help developers to think in a broad sense the role of internet enabled devices. Developers should be able of analyzing upfront the threats that connected devices will face in the future. Even though security requirements for connecting devices to the internet are not necessary at that moment.
- Applications' development process is not a linear process. Contrary, it is an iterative process in which applications evolve or change over time. Then, security requirements engineering process cannot be linear process either. SRE methodologies could encourage developers to think about the threats that new functionalities bring to the product. In this sense, security requirements might adapt changing functional requirements.
- Rules, regulations, and standards might already ask for some security requirements. Then, SRE need to incorporate guide to place these requirements in the development process.
- SRE methods and frameworks should be easy to use to developers. This kind of methods should provide guidance to be used even though developers have not a security background. Besides, at a high level, managers should be able to recognize main aspects of the methodology.

Moreover, security requirements engineering methodologies should be promoted among small companies and potential developers. It was mentioned early that it is still not clear if developers are using SRE tools. Developers might benefit from the techniques and frameworks developed by security experts to handle security requirements. However, to do so, they need to know that these methods exist. Lectures, meetings with developers, conferences, and seminars could help researchers to present their methods. Collaboration between developers and industry could also play a role here because, for accessing companies, researchers will need to share and exchange knowledge with developers and managers. Besides, security experts could feedback their methods with developers' experiences and knowledge.

9.3.3 Developers of medical IoT applications

Security is significant value in the healthcare arena. For instance, patients' privacy needs to be ensured, patients' data should not be accessible for unauthorized actors, and continue services should be guaranteed. Developers should consider values that are important for society early in the development to produce applications which gain acceptance of users. As we have seen during Chapter 3, technology affects values. Nevertheless, developers take decisions on the functions of the technology. Thus, developers' actions could produce technologies that enhance fundamental values of society. A value sensitive design approach might help developers to think about values early in the development process.

To produce secure products, companies need to create an environment that fosters a secure product development. Managers should undertake the responsibility of creating this environment. Managers could start by implementing a policy which includes the intent of producing secure products and applications, guidance on valuable assets for the companies, periods to update and verify security requirements and security mechanisms. Nevertheless, more vital that creating policies, is to spread the intention to develop secure product applications among the company. In this environment, developers, managers, and other relevant actors should be committed with developing products that better understand security.

Developers should try to incorporate security experts in the development process. For instances, there are small and medium companies who are offering services to test the security aspects of devices and applications. Besides, during applications' test, developers could also engage in testing things that the system should *not* do instead of only things that the system should do. This test could give developers an initial idea of what their products are capable of if bad guys take control of these devices.

Finally, developers could employ the tools, methods, frameworks which are being developed by the academy to achieve a secure product development. There is a variety of tools that might help developers to address the security aspects of technologies. We will recommend them to check methods to manage security requirements of the applications early in the development process.

9.4 Future research

During this research project, we have gained insights of the recommended security practices according to the security requirements engineering field, developers' security practices in a medical context, the factors that influence developers' practices, the differences between SRE security practices and developers' practices, and the reasons for such as differences. Future research could use our insights to extend the applications domain, first, to applications in the medical IoT which are not connected to hospitals or hospitals' information system, and then, to different IoT domains. Security practices during the development of applications that are not too regulated, as it is the case of medical applications, could offer interesting explanations regarding the factors that lead to address security and the factors that shape developers' strategies.

Similar studies could be conducted once an IoT ecosystem is established and visible. As we saw in Chapter 1 security in the IoT is a major challenge for developers, Academy, and users. Besides, security can influence the widespread adoption of IoT technologies. This challenge will expand when new players enter into the IoT ecosystem. Then, it would be interesting to understand how the ecosystem value security and which are the main incentives for addressing security during the early design and development.

Finally, new security requirement engineering methods and frameworks that better meet small companies' necessities need to be developed. Researchers could start adjusting current methods or develop new methodologies. Design-oriented research could be employed for undertaking this task. ■

References

Abie, H., & Balasingham, I. (2012). Risk-Based Adaptive Security for Smart IoT in eHealth. *Proceedings of the 7th International Conference on Body Area Networks*, (SeTTIT), 269–275. http://doi.org/10.4108/icst.bodynets.2012.250235

Agile Alliance. (2015). 12 Principles Behind the Agile Manifesto. Retrieved June 25, 2017, from https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

agilemethodology.org. (2008). The Agile Movement. Retrieved June 26, 2017, from http://agilemethodology.org/

Ahmed, K. A., Aung, Z., & Svetinovic, D. (2013). Smart Grid Wireless Network Security Requirements Analysis. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 871–878. http://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.153

Alqassem, I., & Svetinovic, D. (2014). A taxonomy of security and privacy requirements for the Internet of Things (IoT). *IEEE International Conference on Industrial Engineering and Engineering Management*, *2015–Janua*, 1244–1248. http://doi.org/10.1109/IEEM.2014.7058837

Amsterdam business news. (2016). Amsterdam growing as a digital healthcare hub. Retrieved April 26, 2017, from http://www.iamsterdam.com/en/business/news/2016/amsterdam-growing-as-a-digital-healthcare-hub

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. http://doi.org/10.1007/s10796-014-9492-7

Babar, S., Mahalle, P., Stango, A., Prasad, N., & Prasad, R. (2010). Proposed security model and threat taxonomy for the Internet of Things (IoT). *Communications in Computer and Information Science*, *89 CCIS*, 420–429. http://doi.org/10.1007/978-3-642-14478-3_42

Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications*, *54*, 1–31. http://doi.org/10.1016/j.comcom.2014.09.008

Brown, N., Ozkaya, I., Sangwan, R., Seaman, C., Sullivan, K., Zazworka, N., … Nord, R. (2010). Managing technical debt in software-reliant systems. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research - FoSER '10*, 47. http://doi.org/10.1145/1882362.1882373

Cambridge Dictionary. (2017). guilt Meaning in the Cambridge English Dictionary. Retrieved January 18, 2017, from http://dictionary.cambridge.org/dictionary/english/guilt

Cheng, B. H. C., & Atlee, J. M. (2007). Research Directions in Requirements Engineering Research Directions in Requirements Engineering. *Proceeding FOSE '07 2007 Future of Software Engineering*, 285–303. http://doi.org/10.1109/FOSE.2007.17

Coffman, K. G., & Odlyzko, A. M. (2002). Growth of the Internet. In I. Kaminow & L. Tingye (Eds.), *Optical Fiber Telecommunications IV-B: Systems and Impairments*. San Diego: Academic Press.

Corbin, J., & Strauss, A. (1990). Grounded Theory Research: Procedures, Canona and Evaluative Criteria. *Qualitative Sociology*, *13*(1), 3–21. http://doi.org/10.1007/BF00988593

Covington, M. J., & Carskadden, R. (2013). Threat Implications of the Internet of Things. *2013 5th International Conference on Cyber Conflict*, 1–12.

Crook, R., Ince, D., Lin, L., & Nuseibeh, B. (2002). Security requirements engineering: When anti-requirements hit the fan. *Proceedings of the IEEE International Conference on Requirements Engineering*, *2002–Janua*, 203–205. http://doi.org/10.1109/ICRE.2002.1048527

Cunningham, R., Gupta, P., Lindqvist, U., Sidiroglou-Douskos, S., & Hicks, M. (2016). IEEE SecDev 2016: Prioritizing Secure Development. *IEEE Security and Privacy*, *14*(4), 82–84. http://doi.org/10.1109/MSP.2016.71

Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, *10*(4), 2233–2243.

Daneva, M., Damian, D., Marchetto, A., & Pastor, O. (2014). Empirical research methodologies and studies in Requirements Engineering: How far did we come? *Journal of Systems and Software*, *95*, 1–9. http://doi.org/10.1016/j.jss.2014.06.035

Detweiler, C. A., & Hindriks, K. V. (2015). A survey of values, technologies and contexts in pervasive healthcare. *Pervasive and Mobile Computing*, *27*, 1–13. http://doi.org/10.1016/j.pmcj.2015.09.002

Deutsche Forschungsgemeinschaft., P. (2000). *Forum, qualitative social research*. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* (Vol. 8). Deutsche Forschungsgemeinschaft. Retrieved from http://www.qualitative-research.net/index.php/fqs/article/view/291/641

Dubois, E., & Mouratidis, H. (2010). Guest editorial: Security requirements engineering: Past, present and future. *Requirements Engineering*, *15*(1), 1–5. http://doi.org/10.1007/s00766-009-0094-8

Egusa, C., & Cohen, S. (2015). The Netherlands: A Look At The World's High-Tech Startup Capital. Retrieved April 26, 2017, from https://techcrunch.com/2015/07/05/the-netherlands-a-look-at-the-worlds-high-tech-startup-capital/

Estévez-Reyes, L. (2016). How the Internet of Things Affects the Premises in Geekonomics. *SoutheastCon*, 1–7. http://doi.org/10.1109/SECON.2016.7506689

European Commission. (2017). CE marking. Retrieved June 16, 2017, from https://ec.europa.eu/growth/single-market/ce-marking_en

European Commission. (2017). Notified bodies. Retrieved July 26, 2017, from https://ec.europa.eu/growth/single-market/goods/building-blocks/notified-bodies_en

European Commission. (2017). Protection of personal data. Retrieved June 25, 2017, from http://ec.europa.eu/justice/data-protection/

European Parliament. (2007). Council Directive 93/42/EEC. *Official Journal of the European Union*, (June 1993), 1–60. http://doi.org/2004R0726 - v.7 of 05.06.2013

Evans, D. (2011). The Internet of Things: How the Next Evolution of the Internet is Changing Everything. *CISCO White Paper*, (April), 1–11. http://doi.org/10.1109/IEEESTD.2007.373646

Fabian, B., Gurses, S., Heisel, M., Santen, T., & Schmidt, H. (2009). A comparison of security

requirements engineering methods. *Requirements Engineering*, *15*(1), 7–40. http://doi.org/10.1007/s00766-009-0092-x

FBI Cyber Division. (2014). *(U) Health Care Systems and Medical Devices at Risk for Increased Cyber Intrusions for Financial Gain* (Vol. 140408–9).

Federal Trade Commission. (2014a). FTC Approves Final Order Settling Charges Against TRENDnet, Inc. Retrieved June 6, 2017, from https://www.ftc.gov/news-events/press-releases/2014/02/ftc-approves-final-order-settling-charges-against-trendnet-inc

Federal Trade Commission. (2014b). *Privacy and Data Security at the Federal Trade Commission: Recent Developments*.

Firesmith, D. (2003). Engineering Security Requirements. *Journal of Object Technology*, *2*(1), 53–68. Retrieved from http://www.jot.fm/issues/issue_2003_01/column6/

Firesmith, D. (2004). Specifying Reusable Security Requirements. *Journal of Object Technology*, *3*(1), 61–75. Retrieved from http://www.jot.fm/issues/issue_2004_01/column6/

Firesmith, D. (2007). Engineering Safety- and Security-Related Requirements for Software-Intensive Systems : Tutorial Summary. *ICCBSS'2007 Conference Tutorial Software Engineering Institute Carnegie Mellon University Pittsburgh*, 1–100. Retrieved from http://resources.sei.cmu.edu/asset_files/Presentation/2007_017_001_22993.pdf

Flechais, I., & Sasse, M. A. (2007). Stakeholder involvement, motivation, responsibility, communication: How to design usable security in e-Science. *International Journal of Human Computer Studies*, *67*(4), 281–296. http://doi.org/10.1016/j.ijhcs.2007.10.002

Friedman, B., Kahn Jr., P. H., & Borning, A. (2006). Value Sensitive Design and Information Systems. *Human-Computer Interaction and Management Information Systems: Foundations*, 1–27. http://doi.org/10.1145/242485.242493

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. (2016). Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Transactions on Software Engineering*, *42*(6), 585–604. http://doi.org/10.1109/TSE.2015.2509970

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. http://doi.org/10.1016/j.future.2013.01.010

Gürgens, S., Rudolph, C., Maña, A., & Nadjm-Tehrani, S. (2010). Security engineering for embedded systems. *Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems - S&D4RCES '10*, 1. http://doi.org/10.1145/1868433.1868443

Haley, C. B., Laney, R., Moffett, J. D., & Nuseibeh, B. (2006). Arguing Satisfaction of Security Requirements. In *Integrating security and software engineering: advances and future vision.* (pp. 15–42). http://doi.org/10.4018/978-1-59904-147-6.ch002

Haley, C. B., Laney, R., Moffett, J. D., & Nuseibeh, B. (2008). Security Requirements Engineering: A Framework for Representation and Analysis. *IEEE Transactions on Software Engineering*, *34*(1), 133–153. http://doi.org/10.1109/TSE.2007.70754

Hautala, L. (2017). Smart toy flaws make hacking kids' info child's play. Retrieved June 6, 2017, from https://www.cnet.com/news/cloudpets-iot-smart-toy-flaws-hacking-kids-info-children-

cybersecurity/

Holmström, J., & Sawyer, S. (2011). Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements. *European Journal of Information Systems*, *20*(1), 34–47. http://doi.org/10.1057/ejis.2010.51

Hoo, S., Sudbury, A. W., & Jaquith, A. R. (2001). Tangible ROI through Secure Software Engineering. *Secure Business Quarterly*, *1*(2). Retrieved from p:%5Cknowledge%5Cpapers%5Csbq_rosi_software_engineering.pdf

Jadoul, M. (2015). The IoT: The next step in internet evolution. Retrieved July 12, 2017, from https://insight.nokia.com/iot-next-step-internet-evolution

Jøsang, A., Ødegaard, M., & Oftedal, E. (2015). Cybersecurity Through Secure Software Development. *IFIP International Federation for Information Processing*, 53–63. http://doi.org/10.1007/978-3-319-18500-2

Keoh, S. L., Kumar, S. S., & Tschofenig, H. (2014). Securing the Internet of Things: A Standardization Perspective. *IEEE Internet of Things Journal*, *1*(3), 265–275. http://doi.org/10.1109/JIOT.2014.2323395

Kim, J., & Lee, J. W. (2014). OpenIoT: An open service framework for the Internet of Things. *2014 IEEE World Forum on Internet of Things, WF-IoT*, 89–93. http://doi.org/10.1109/WF-IoT.2014.6803126

Kocher, P., Lee, R., McGraw, G., & Raghunathan, A. (2004). Security as a new dimension in embedded system design. *Proceedings of the 41st Annual Design Automation Conference*, 753–760. http://doi.org/10.1145/996566.996771

Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. (2016). Learning Internet-of-Things Security "Hands-On." *IEEE Security & Privacy*, *14*(January), 37–46. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7397713&isnumber=7397706

Krebs, B. (2016). Hacked Cameras, DVRs Powered Today's Massive Internet Outage. Retrieved December 10, 2016, from https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/

LaRossa, R. (2005). Grounded Theory Methods and Qualitative Family Research. *Journal of Marrige and Family*, *67*(November), 837–857. http://doi.org/10.1111/j.1741-3737.2005.00179.x

Laudon, K. C., & Laudon, J. P. (2014). Ethical and Social Issues in Information Systems. In S. Wall (Ed.), *Management Information Systems* (Thirteen, pp. 1070–1077). Harlow: Pearson Prentice Hall. http://doi.org/http://dx.doi.org/10.1016/B0-12-227240-4/00108-8

Li, S., Tryfonas, T., & Li, H. (2016). The Internet of Things: a security point of view. *Internet Research*, *26*(2), 337–359. http://doi.org/10.1108/IntR-07-2014-0173

Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, *17*(2), 243–259. http://doi.org/10.1007/s10796-014-9492-7

March, B. S. T., & Storey, V. C. (2008). Design Science in the Information Systems. *MIS Quaterly*, *32*(4), 725–730.

Markowsky, L., & Markowsky, G. (2015). Scanning for vulnerable devices in the Internet of Things. *Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data*

Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015, *1*, 463–467. http://doi.org/10.1109/IDAACS.2015.7340779

Mavropoulos, O., Mouratidis, H., Fish, A., Panaousis, E., & Kalloniatis, C. (2016). APPARATUS: Reasoning About Security Requirements in the Internet of Things. *Advanced Information Systems Engineering Workshops*, 221–254. http://doi.org/10.1007/978-3-319-13776-6_9

Mazhelis, O., Warma, H., Leminen, S., Ahokangas, P., Pussinen, P., Rajahonka, M., … Myllykoski, J. (2013). Internet-of-Things Market, Value Networks, and Business Models: State of the Art Report. *Computer Science and Information Systems Reports*, 1–93. Retrieved from http://blog.prossess.com/wp-content/uploads/2013/10/IoT-SOTA-Report-2013.pdf

McCue, T. (2015). $117 Billion Market For Internet of Things In Healthcare By 2020. Retrieved August 1, 2017, from https://www.forbes.com/sites/tjmccue/2015/04/22/117-billion-market-for-internet-of-things-in-healthcare-by-2020/#2172fc7769d9

Mead, N. R., & Stehney, T. (2005). Security Quality Requirements Engineering (SQUARE) Methodology. *ACM SIGSOFT Software Engineering Notes*, *30*(4), 1. http://doi.org/10.1145/1082983.1083214

Mead, N. R., Viswanathan, V., & Padmanabhan, D. (2008). Incorporating Security Requirements Engineering into the Dynamic Systems Development Method. *2008 32nd Annual IEEE International Computer Software and Applications Conference*, 949–954. http://doi.org/10.1109/COMPSAC.2008.85

Mellado, D., Blanco, C., Sánchez, L. E., & Fernández-Medina, E. (2010). A systematic review of security requirements engineering. *Computer Standards and Interfaces*, *32*(4), 153–165. http://doi.org/10.1016/j.csi.2010.01.006

Mellado, D., Fernandez-Medina, E., & Piattini, M. (2006). Applying a Security Requirements Engineering Process. *Computer Security – ESORICS*, *4189*, 192–206. Retrieved from https://link.springer.com/chapter/10.1007/11863908_13

Miles, M. B., Huberman, M. A., & Saldana, J. (2014). *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications, Inc.

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. *Computer Communications*, *89–90*, 5–16. http://doi.org/10.1016/j.comcom.2016.03.015

Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). *IEEE Internet of Things*, (1), 86. http://doi.org/10.5120/19787-1571

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. http://doi.org/10.1016/j.adhoc.2012.02.016

Mittelstadt, B. (2017). Designing the Health-Related Internet of Things: Ethical Principles and Guidelines. *Information*, 1–25. http://doi.org/10.3390/info8030077

Nass, S. J., Levit, L. A., & Gostin, L. O. (2009). *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*. *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*. The National Academies Press. Retrieved from http://www.ncbi.nlm.nih.gov/books/NBK9571/ accessed 01/08/2016

Nederlandse Vereniging van Ziekenhuizen. (2016). Bewerkersovereenkomst. Retrieved June 5, 2017, from https://www.nvz-ziekenhuizen.nl/onderwerpen/bewerkersovereenkomst

NEN. (2017). Nieuwe versie NEN 7510 “Informatiebeveiliging in de zorg” gepubliceerd voor commentaar. Retrieved June 25, 2017, from https://www.nen.nl/NEN-Shop/Nieuwsberichten-Zorg-Welzijn/Nieuwe-versie-NEN-7510-Informatiebeveiliging-in-de-zorg-gepubliceerd-voor-commentaar.htm

Orlikowski, W. J. (1993). CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Radical Changes in Systems Development. *MIS Quarterly*, *17*(3), 309–340. http://doi.org/10.2307/249774

OWASP. (2011). *Internet of Things Top Ten*. Retrieved from http://www.mycisco.net/web/ES/assets/executives/pdf/Internet_of_Things_IoT_IBSG_0411FINAL.pdf

Oxford Dictionaries. (2017). perception - definition of perception in English. Retrieved August 11, 2017, from https://en.oxforddictionaries.com/definition/perception

Päivärinta, T., & Smolander, K. (2015). Theorizing about software development practices. *Science of Computer Programming*, *101*, 124–135. http://doi.org/10.1016/j.scico.2014.11.012

Paja, E., Dalpiaz, F., & Giorgini, P. (2014). STS-Tool: Security Requirements Engineering for Socio-Technical Systems. In M. Heisel, W. Joosen, J. Lopez, & F. Martinelli (Eds.), *Engineering Secure Future Internet Services and Systems: Current Research* (pp. 65–96). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-07452-8_3

Pandit, N. R. (1996). The creation of theory: A recent application of the grounded theory method. *The Qualitative Report*, *2*(4), 1–15. http://doi.org/10.1186/gb-2006-7-9-r80

Philips Research. (2004). Philips Research Eindhoven (Headquarters). Retrieved April 25, 2017, from http://www.philips.com/a-w/research/locations/eindhoven.html

Ramachandran, M. (2015). Software Security Requirements Engineering: State of the Art. In H. Jahankhani, A. Carlile, B. Akhgar, A. Taal, A. G. Hessami, & A. Hosseinian-Far (Eds.), *Global Security, Safety and Sustainability: Tomorrow’s Challenges of Cyber Security: 10th International Conference, ICGS3 2015, London, UK, September 15-17, 2015. Proceedings* (pp. 313–322). Springer International Publishing. http://doi.org/10.1007/978-3-319-23276-8_28

Robertson, S., & Robertson, J. (2013). *Mastering the Requirements Process: Getting Requirements Right*.

Sadeghi, A. (2016). Games without Frontiers: Whither Information Security and Privacy? *IEEE Security & Privacy*, (February), 3–5.

Schilling, M. A. (2005). *Strategic Management of Technological Innovation* (Fourth Edi). New York: McGraw-Hill/Irwin.

Souag, A., Salinesi, C., Mazo, R., & Comyn-Wattiau, I. (2015). A Security Ontology for Security Requirements Elicitation. In F. Piessens, J. Caballero, & N. Bielova (Eds.), *Engineering Secure Software and Systems: 7th International Symposium, ESSoS 2015, Milan, Italy, March 4-6, 2015. Proceedings* (pp. 157–177). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-15618-7_13

Tarkoma, S., & Katasnov, A. (2011). Internet of Things Strategic Research Agenda. *Finnish Stra- Tegic Centre for Science, Technology and Innovation*, 40. Retrieved from http://books.google.com/books?hl=en&lr=&id=Eug-RvslW30C&oi=fnd&pg=PA9&dq=Internet+of+Things+Strategic+Research+Agenda&ots=3SxaDKhzvq&sig=w2FKkLgGm6wya1vQRc9naG985PM

Thielman, S., & Johnston, C. (2016). Major cyber attack disrupts internet service across Europe and US. Retrieved December 10, 2016, from https://www.theguardian.com/technology/2016/oct/21/ddos-attack-dyn-internet-denial-service

Tondel, I. A., Jaatun, M. G., & Meland, P. H. (2008). Security Requirements for the Rest of Us: A Survey. *IEEE Software*, *25*(1), 20–27. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4420063&isnumber=4420053

Tryfonas, T., Kiountouzis, A., & Poulymenakou, A. (2001). Embedding security practices in contemporary information systems development approaches. *Information Management & Computer Security*, *9*(4), 183–197.

van de Poel, I., & Royakkers, L. (2011). *Ethics, Technology, and Engineering: An Introduction*. http://doi.org/10.1128/AAC.03728-14

van der Meulen, R., & Rivera, J. (2015). Gartner Says By 2020, a Quarter Billion Connected Vehicles Will Enable New In-Vehicle Services and Automated Driving Capabilities. Retrieved December 10, 2016, from http://www.gartner.com/newsroom/id/2970017

Vaughn, R. B., Henning, R., & Fox, K. (2002). An empirical study of industrial security-engineering practices. *Journal of Systems and Software*, *61*(3), 225–232. http://doi.org/10.1016/S0164-1212(01)00150-9

Verschuren, P., & Doorewaard, H. (2010). *Designing a Research Project* (Second edi). Eleven International Publishing.

Verschuren, P., & Hartog, R. (2005). Evaluation in Design-Oriented Research. *Quality and Quantity*, *39*(6), 733–762. http://doi.org/10.1007/s11135-005-3150-6

Viega, J. (2005). Building security requirements with CLASP. *ACM SIGSOFT Software Engineering Notes*, *30*, 1. http://doi.org/10.1145/1082983.1083207

Werlinger, R., Hawkey, K., & Beznosov, K. (2009). An integrated view of human, organizational, and technological challenges of IT security management. *Information Management & Computer Security*, *17*(1), 4–19. http://doi.org/10.1108/09685220910944722

White, G. (2009). Strategic, Tactical, & Operational Management Security Model. *Journal of Computer Information Systems*, 71–76.

Williams, P. A. H., & McCauley, V. (2017). Always connected: The security challenges of the healthcare Internet of Things. *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, 30–35. http://doi.org/10.1109/WF-IoT.2016.7845455

Wortman, P. A., Tehranipoor, F., Karimian, N., & Chandy, J. A. (2017). Proposing a modeling framework for minimizing security vulnerabilities in IoT systems in the healthcare domain. *2017 IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2017*, 185–188. http://doi.org/10.1109/BHI.2017.7897236

Yandron, D. (2016). Fisher-Price smart bear allowed hacking of children’s biographical data.

Retrieved June 1, 2017, from https://www.theguardian.com/technology/2016/feb/02/fisher-price-mattel-smart-toy-bear-data-hack-technology

Yin, R. K. (2014). *Case Study Research Design and Methods* (Applied So). SAGA Publications. http://doi.org/10.1097/FCH.0b013e31822dda9e

Zetter, K. (2017). Medical Devices That Are Vulnerable to Life-Threatening Hacks. Retrieved July 12, 2017, from https://www.wired.com/2015/11/medical-devices-that-are-vulnerable-to-life-threatening-hacks/

Zitnik, S., Jankovic, M., Petrovcic, K., & Bajec, M. (2016). Architecture of Standard-based, Interoperable and Extensible IoT Platform. *IEEE*, 4.

Appendix A: Factors influencing developers' security practices: concepts and findings (complete version)

Categories	Concepts	Data from Company1	Data from Company2	Data from Company3
Context	Setting where application operates.	▪ Patients are monitoring from home, and the information is sent to the hospital.	▪ Application for monitoring patients in a hospital.	▪ Application for monitoring patients 24 hours in a hospital.
	Stage of development	▪ The application is fully developed (upscaling).	▪ Prototype phase close to launch a pilot product. ▪ As a prototype, the application does not need all functionalities.	▪ Testing feasibility of a prototype application. ▪ Working on converting the device in a small version.
Conditions for addressing security	Developers' perception of security	▪ Security is considered as a selling point.	▪ Security is considered as a feature of integrating the app with ICT.	▪ The company is satisfied if customers feel secure using the application. ▪ Security is interpreted differently in different countries; thus, rules and regulations also differ.
	Rules and regulation	▪ Regulations for e-Health products that process personally identifiable data. Personal Data Protection Act and NEM standard 7510 Information security in care. ▪ The company is held responsible for ensuring privacy and security of patients' data.	▪ Analyzing rules and regulations for different part of the application	▪ Medical devices need to comply with existing rules and regulations to be sold in Europe (CE Marking) and in other countries. ▪ Medical devices need to comply with standards to be sold in the European market. ▪ Companies should provide right evidence to notified bodies that the product

		▪ External parties processing and (scientific) analyzing hospitals/patients' data should sign an agreement with the hospital. Requirements and responsibilities in the field of information security and privacy must be properly regulated and recorded. (NVZ).		meets the required standards. ▪ Risk assessment of the product as part of the European regulation. ▪ The company is the legal manufacturer, and they will submit information to notified body to get the certifications.
	Hospital security knowledge	▪ Some hospitals required specific security requirements for applications that are integrated with hospital information system (3 categories of security requirements: SR for parties involved, for the Internet or application service providers, for the application). ▪ Hospitals have security officers to verify that applications meet the hospitals' security requirements.	▪ Hospital's ICT should approve if you want to send information outside the hospital. ▪ IT department ask is it possible for a hacker to come through the device and access the hospital's network	▪ Hospital has specific departments to check all devices that are used in the hospital which includes prototype, clinical trials, etc.
	Development approach	▪ Development cycles (per quarter) and road map to organize activities. ▪ Focus on providing new functionalities and improving the application/product.	▪ Iterative and incremental product development process focusses on delivering functionalities quickly. ▪ Functionalities and development are prioritized	▪ Multiples parties to develop the technology (component based development) focus on delivering core functionality.

			based on stakeholders' needs.	▪ In-house concept and integration of the application.
Strategies for dealing with security	Involving reliable stakeholders	▪ Incorporate security expertise (certified security officer) as part of the company. ▪ Involving responsible third parties that follow CE rules and regulations for medical data (e.g. hosting company).	▪ For the integration with ICT company expect to work with a third-party expert on security or a well-known brand that generates trust.	▪ Involve multiples parties with different expertise which include compliance and regulatory consulting companies, external testing, notified bodies. ▪ Involve partners with experience in healthcare
	Incorporating SR when it's required	▪ SR are included in the roadmap, and addressed, according to priority. ▪ SR are prioritized by law, customer, financial consequences. ▪ Engaging in negotiations to gain space (time) for adding the required security requirements.	▪ The security issues of integrating the device with ICT will be tackled as a final phase in the development process. ▪ The device is not integrated with ICT and data is not sent out of the hospital until device's functionalities satisfy customer's necessities.	▪ The company will follow all the rules and regulations for IoT medical applications when the application arrives at that point (expected in 2020).
	Adjusting technology features	▪ Balance hardware and software of the application. ▪ Building the system in accordance with rules and regulations.	▪ Balance hardware and software of the application. ▪ Very output/measurements of the device. ▪ Application does not transmit patient's data.	▪ Ensuring right data is transmitted to the right receiver. ▪ Apply cryptography for sending data. ▪ The application does not take patient's information.
Intervening conditions	Developers' perception	▪ Security requirements constrain usability, and customers feel the constraint.	▪ Failures in security make difficult to recover customers' trust in the product (damage of reputation).	▪ Small incidents can stop the whole business.

			▪ Lack of expertise makes difficult to address the security part by yourself.	
	Customers' necessities	▪ Different necessities between the customer (the person who use the product/buy the product) and security officers. ▪ Customers place less importance on security.	▪ Customers (nurse) has to like our product. Thus, first, focus on satisfying customers' necessity. ▪ Nurse or specialists are usually not concerned with (ICT) security.	▪ Our customers (doctor, nurses, parents) have to trust on the product. ▪ The focus is to gain acceptance of medical professionals.
	Challenges facing during the development	▪ Rules and regulations differ per country. ▪ Rules and regulations change raptly. ▪ Not all required requirements can be integrated into the app.	▪ The product should be protected against bacteria, and it must not escalate other staff within the hospital. ▪ Product's features should not harm patients' private information or place's hygiene. ▪ Bounded rationality regarding all possible threats and situations.	▪ The product has to be safe enough to place on the patient. ▪ Achieve the required battery life performance. ▪ Manage whole team and all partners while protecting the companies' intellectual property.
	Company's experience	▪ Commercializing the application in the European market for more than six years.	▪ Working on the product/application development for more than one year.	▪ Working on the application development for more than one year. But the founder and co-founder have a long experience in medical field and technology development.