Pontifical Catholic University of Rio Grande do Sul
TECHNICAL SCHOOL
POSTGRADUATE PROGRAM IN COMPUTER SCIENCE

SECURITY REQUIREMENTS IN AGILE SOFTWARE DEVELOPMENT: A SYSTEMATIC MAPPING STUDY

Leandro Ripoll Saldanha, Avelino Zorzo

Technical Report 087

July 2019

REQUISITOS DE SEGURANÇA NO DESENVOLVIMENTO ÁGIL DE SOFTWARE: UM ESTUDO DE MAPEAMENTO SISTEMÁTICO

RESUMO

Segurança é um quesito que deve ser levado em consideração por qualquer metodologia de desenvolvimento de software. Segurança da informação pode ser considerada uma preocupação antiga, desde o início de nossa civilização, e se torna cada vez mais relevante hoje em dia, levando-se em conta que estamos em plena era da informação. Desde a década de noventa o desenvolvimento de sistemas computacionais vem sendo fortemente afetado por metodologias ágeis tais como o SCRUM e o XP. Este tipo de metodologia foca em pequenas entregas de software utilizável em períodos menores de tempo. Estes perdíodos menores de tempo são conhecidos como iterações e a cada nova iteração chega-se mais próximo da entrega final, ou seja, do produto pronto ou versão final. A agilidade está justamente nas entregas mais rápidas e contínuas ao longo do ciclo de vida do projeto. O escopo vai sendo definido ao longo do tempo de projeto, enquanto o software é desenvolvido. As funcionalidades e requisitos de software são definidos por especialistas no negócio também conhecidos como “donos do produto”. Neste cenário, é possível que requsitos de segurança sejam subestimados ou mesmo desvalorizados por falta de conhecimento ou mesmo de experiência no assunto. O objetivo deste artigo é trazer o que vem sendo pesquisado na intersecção entre as áreas de segurança e métodos ágeis em projetos de desenvolvimento de software. Ao atingir este objetivo será possível entender como os métotodos ágeis pretendem lidar com requisitos de segurança.

Palavras-Chave: Requisitos de segurança, Desenvolvimento ágil, Ágil, Mapeamento Sistemático.

SECURITY REQUIREMENTS IN AGILE SOFTWARE DEVELOPMENT: A SYSTEMATIC MAPPING STUDY

ABSTRACT

Security is a concern that has to be considered in software development methodologies. Information security has been a major issue since the beginning of our civilization and it is even more relevant nowadays, considering we live in the information era. Since the nineties, computer-based systems development is being affected by agile methodologies such as SCRUM and XP. They focus on delivering a working piece of software in shorter periods called iterations, until reaching the final version. The agility comes from faster and continuous delivering along the project time, defining the scope while the software is being developed. As software features are pointed by a product owner, it is possible that requirements related to security could be undervalued in this scenario. This article aims to present what is being researched in the intersection between the agile and the security fields regarded to software development projects. By reaching this goal, it will be possible to understand how agile intends to deal with security requirements.

Keywords: Security requirements, Agile development, Agile, Systematic Mapping Study.

1. INTRODUCTION

The Agile Manifesto [12] has consolidated a set of practices that became relevant in software development projects [12] [26]. Agile is considered an adaptative methodology that includes conceptual simplicity. Such simplicity is opposed to the traditional cascade methodologies and focuses on delivering workable piece of software faster, within predefined iterations that count on intense customer's participation all project long. Requirement changes are welcomed through the process, being treated in a dynamic way [12] [20] [27].

The traditional sequential waterfall model tries to reach efficiency by preventing changes. Agility, on the other hand, tries to anticipate them [51]. Even the Project Management Body of Knowledge (PMBOK) has increased the importance of adopting agility by mentioning it as a methodology in the project life cycle. The sixth edition of PMBOK, which nowadays is the newest one, points out considerations for incorporating agile methodologies and such considerations are present in all knowledge areas inside this set of project management processes (recommended practices) [49].

Agile principles have attracted organizations' attention since agility addresses challenges such as communication, cooperation, requirements control and constant changes. Agile software development lightweight methods have become a current mainstream accepted by the software industry [65] [7] [17] [28] [31] [48] [52].

Agile provides tools to cope with requirement changes at any point of the project [27] [50] [52]. Agility, in all its known methodologies, is highly flexible, but on the other hand it raises concerns linked to the security requirements [63]. In spite of existing empirical evidences pointing to the success on dealing with functional requirements, there seems to be a careless approach regarded to security requirements [63].

Practitioners are trying to make agile better by incorporating security practices [1]. Specifically speaking of SCRUM, for instance, it does not indicate procedures to cope with security requirements [36]. Security problems are also reported in other agile methodologies, creating vulnerabilities that can be exploited by attackers [42].

In general, agile development is said to be disadvantageous regarding security requirements [12] [35]. Following the security and agility issues, as explained so far, a number of studies were started in the intersection between both areas, agility and security requirements [4] [5] [6] [8] [9] [17] [18] [25] [32] [33] [36] [37] [45] [58] [61] [68].

At the same time, there are concerns about the effect security requirements can cause over a software development project. Some of the reported symptoms include delays and costs increase [17]. It is a diagnosis that does not match with agile philosophy [11]. A distinct point is that the market is forcing non-specialists in security to develop software that requires complex security features. When they do not have the appropriate security process they fail on providing secure software [24] [54].

According to Highsmith [35] and Williams and Cockburn [67], one of the possibilities to be compliance with security guides is adopting the agile software development. It enables customers to define security requirements during the iterations and change them as frequently as they need [35] [67]. The agile community has adopted some practices to cope with security issues [30], such as abuser stories [45], extended agile practices [18], techniques to cope with the clash between security and agile development [17], abuser stories combined with attack trees [59] and others.

Considering the presented context, some authors believe that the existing research within the area is not enough to understand how security activities can be fully integrated in the agile methodology [22]. The present paper aims to research what is being discussed related to security and agility.

We noticed that the seminal researches related to the intersection between agile and security areas have been done at the end of the ninety decade. That means that security concerns in agile development were discussed almost at the same time as agility was born.

Within this work, we are interested on putting together relevant information, providing data source and delivering insights for future works on security requirements inside agile software development projects. We are especially interested on practices for coping with security requirements in agile software development projects.

In order to reach an understanding about the intersection between agile and security areas, some research questions were formulated with the intention of guiding the present work. The three research questions are presented below, but they will be explained in deeper details within Chapter 3.

RQ1. What is being researched related to agile and security requirements?

RQ2. What are the artifacts, tools and methodologies used by agile for dealing with security requirements?

RQ3. What is the adherence degree of agile when the subject is to develop secure software?

By answering these questions we intend to reach the research goal, which is to provide an overview of the intersection between agile and security areas. The chosen methodology for reaching the research goal is the Systematic Mapping Study (SMS). Such methodology will be thoroughly elicited in Chapter 3.

This work is organized as follows. Chapter 1 has presented the introduction, the initial concepts and the reasons that justify this research. The research goal, research questions and research methodology were also briefly introduced at this chapter. Chapter 2

provides important concepts regarding agility and security, setting the theoretical basis of this research, as well as the state of art on the relation between agility and software security. Chapter 3 goes deeper on presenting the adopted research methodology, the research goal and the research questions. Chapter 4 analyses the studied papers, the ones that were considered in the Systematic Mapping Study. Chapter 5 brings us the conclusions of this study, which means it synthesizes the database analysis in a map containing the main findings. Chapter (6) provides some ideas and suggestions for future work, based on the main findings of the present research (produced map).

2. AGILE AND SECURITY CONCEPTS

2.1 Agility

The Agile Manifesto [12] has defined agility in contrast with the older practices such as Waterfall. Instead of focusing on processes, tools and documentation, agile focus on people and their interactions, software rather than documentation and customer active participation [12].

There are twelve principles that guide the agile philosophy and distinguish it from the others. These principles are [12]:

- Customer satisfaction through early and continuous delivery of valuable software;
- Welcome change requirements, even late in development;
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale;
- Business people and developers must work together daily throughout the project;
- Build projects around motivated individuals, give them the environment and support they need and trust them to get the job done;
- Face-to-face conversation is the most efficient and effective method of conveying information to and within a development team;
- Working software is the primary measure of progress;
- Sustainable development – the sponsors, developers and users should be able to maintain a constant pace indefinitely;
- Continuous attention to technical excellence and good design enhances agility;
- Simplicity – the art of maximizing the amount of work not done is essential;
- Self-organizing teams;
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

There are different agile frameworks such as Extreme Programming (XP) [13], SCRUM [37] and Kanban [2]. Each one has its own characteristics but all of them are based on the Agile Manifesto [12].

SCRUM and XP are the most adopted agile methodologies [65] [52] [51]. Both present pre-scheduled events and deliver workable software in a short and predefined periods of time, as suggested in the agile principles. Advanced system users define the software features and set the priorities with the development team, in order to define which feature will be delivered first [17] [52].

As one of the most adopted agile framework [65] [51] [52], SCRUM has set some important concepts related to the agility and widely used within agile teams. It names User Stories the system requirements specified by the point of view of the users. The development team has to transform the Users Stories into software features [51].

SCRUM also names Sprint the amount of time within the development team delivers some Users Stories as workable software. It is a fixed period of time normally between two and four weeks. At the end of each Sprint, the development team has to deliver the Sprint Backlog ready for being tested by the users [51].

Product Backlog and Sprint Backlog are other concepts introduced by SCRUM. A Product Backlog is a set of all User Stories (product requirements) needed to develop the new product. The Sprint Backlog is a set of User Stories chosen to be delivered by the development team at the end of the current Sprint [51].

Another important concept created by SCRUM is the Definition of Done. It stands for a set of consistent criteria that define when an item can be considered done and ready to be accepted by the users [51].

SCRUM also relies on what is known as Ceremonies to keep the communication flowing through the team. There are short daily meetings amongst the development team, known simply as Daily Scrum (fifteen minutes). The Sprint Review is used to approve the delivered items and the Sprint Retrospective is used to discuss how things were done and what can be improved related to the way of working. At regular intervals of time the development team have to discuss on what is being done until now and how to increase the quality level from now on (continuous improvement) [51].

In order to keep the agile framework running, the Scrum Master is the guardian of the Scrum Framework and also helps the development team to resolve any conflict or impediment which is blocking the normal flow of the job. The Scrum Master is the responsible for conducting the SCRUM events, keeping the team walking in the agile line [51].

The Product Owner (PO) is another role that defines the features and business rules the software has to implement. Ultimately the PO defines the user stories [51]. It is important for the development team keeping the communication flowing not only amongst them but also between the team and the Product Owner. The following figure presents an outlook of the Scrum framework.

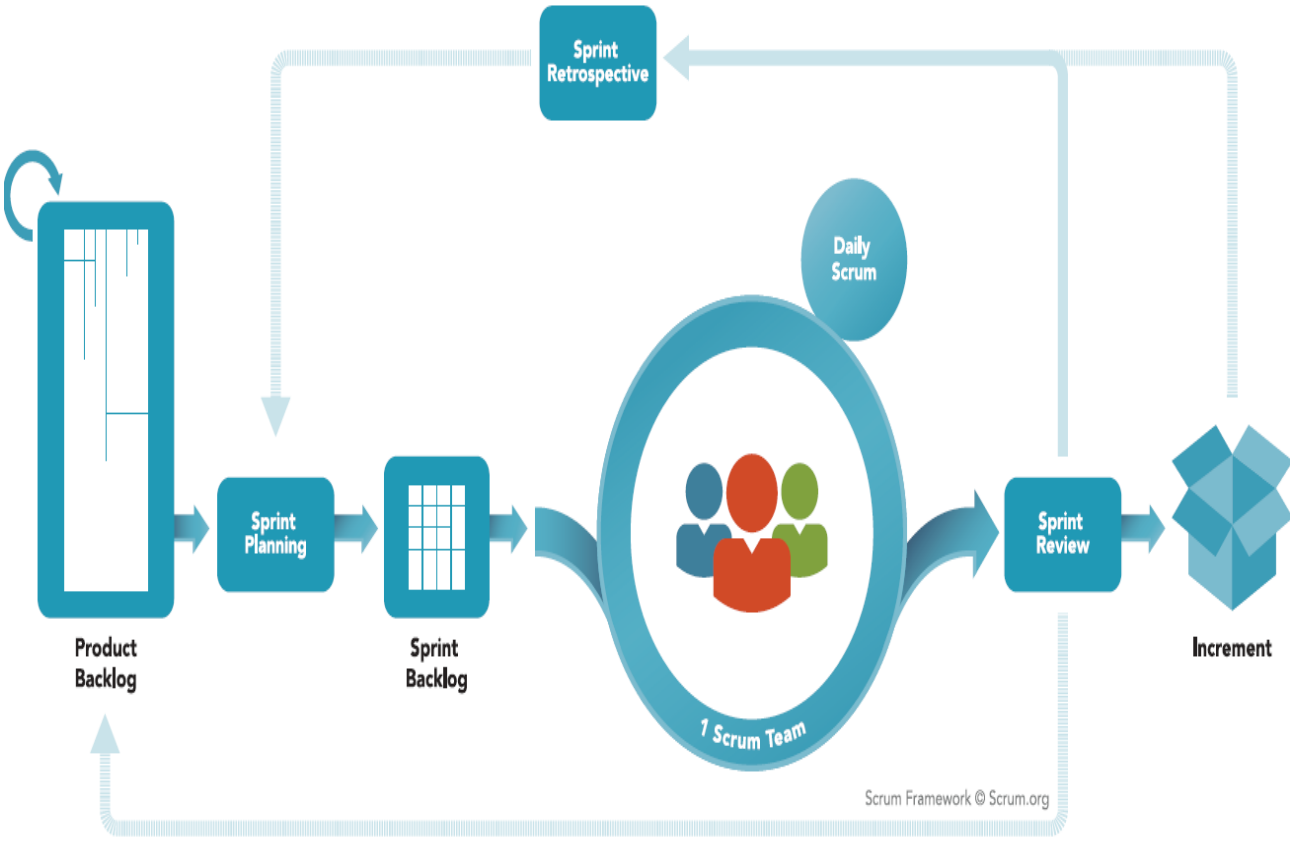

Figure 2.1 – The Scrum Framework [55].

There are a number of other agile frameworks, for example, Feature Driven Development (FDD) and Lean Software Development. Again, all of the existing agile frameworks are based on the principles of the Agile Manifesto [4] [12].

An important concept to be presented is the definition of abuse case. According to McDermott [41], an abuse case is a complete interaction between one or more actors and a system that can cause damage. It does not matter whether it is intentional or not, the results are always harmful to the system and the data involved [41].

Based on abuse cases, abuser stories are similar to the user stories used in agile frameworks such as SCRUM, but they focus on security [5]. One has to understand the abuse case scenarios in order to write good enough abuser stories, which will be transformed in security features within the system [5].

Section 3.2 will bring some important concepts related to security and security tests. At some point of the agile development, the security requirements need to be tested in order to be validated and considered done. SCRUM, for instance, proposes to test such requirements within a Sprint Review. Approval will be executed by the Product Owner.

Still considering SCRUM, it is clear to see the importance of the Done Definition, as mentioned before. A Product Owner will be able to test a security requirement only if they know what a security requirement is and how it can be considered successfully done.

2.2 Security

Ayalew and Kidane [4] believe software security is about developing secure software. It is related to the taken choices linked to technology, platform, language, way codes are written and so on [4]. The authors also highlight that there is a further step to take after the development process is ended, which stands for application protection.

Security Engineering is also defined as set of activities performed during the development process, which are intended to guarantee the delivering of secure software [4].

Microsoft has written down its experiences and best practices on software security in an integrated process called Secure Development Life-Cycle (SDL) [36]. Another example of security instructions is VAHTI, built up by the Finnish government in 2001. VAHTI is a Finnish acronym that stands for the Government Information Security Management Board. All the software suppliers have to be compliance with VAHTI for any kind of information that passes through the Finnish government agencies since 2014 [51] [53].

Following the same way, in 2016, European Union (EU) has published the so called General Data Protection Regulation (GDPR) [23]. That law sets the EU citizens have the right over their personal data, being valid all over EU and applied to any company that manipulates EU citizens' data. The newest version has just taken into its full effect within last May, 2018.

GDPR states that all EU citizens must have complete access to their personal data. They have to be able to consult, alter and even delete their information from a given application. Companies who are proved not to be compliance with GDPR are susceptible to fines that can reach 4 percent over their global revenues [23].

At this point, it is important to state what is considered a software vulnerability as well as other important concepts related to software security. Those theoretical bases will be taken as a guideline for this research.

Software vulnerabilities are weak points of the software and provide opportunities for the attackers to use the software in a non-proper way. In terms of risks, information security is a concern that deserves even more attention nowadays. If an organization ignores the risks they are more likely to suffer an attack as a result of such lack of security attention. One alternative to mitigate that kind of risk is adopting security techniques as Pentest [15].

Penetration Test (Pentest) is a way to simulate attacks by trying to reproduce real external situations of attack. It intends to reveal systems or even networks flaws by planning,

executing and documenting a series of attacks [15]. Pentest is a security test technique that is able to provide a good level of details related to a target system weaknesses. Even considering the Pentest technique, it is hard to define a pattern for all the activities within the tests domain. The experience of the tester is still relevant when it comes to apply the tests and documents their results [15]. Therefore, it is important having a pattern or even a model to guide the test activities and reduce the differences of the results derived from the different levels of expertise [15].

Within its Systematic Mapping Study, Bertoglio and Zorzo [15] have identified some used methodologies, frameworks and security models as shown here: OSSTMM (Open Source Security Testing Methodology Manual), ISSAF (Information Systems Security Assessment Framework), PTES (Penetration Testing Execution Standard), NIST Guidelines (National Institute of Standards and Technology) and OWASP Testing Guide. The following paragraphs will bring a brief explanation of each one of these methodologies.

OSSTMM is an international methodology that comes from ISECOM (Institute for Security and Open Methodologies). It considers all the security environment, including issues related to communication, physical channels and human factors. It is considered one of the most complete models of security tests but it is criticized for not considering cyclic evaluation of found vulnerabilities and diagrams reflecting the tests flaw [15].

ISSAF is a framework to model internal control requirements for security information. It is based on three areas: Planning and Preparing, Evaluation and Report, Cleaning and Destruction of the produced artifacts. It presents a vast documentation related to its structure and creates a connection between the activities and the used tools. As weak points, it can be mentioned the lack of instructions to elaborate the reports and not considering some hypothesis that could make the test procedures better [15].

The PTES brings details and instructions on how to execute required activities to test the security state of a system. It does not intend to be prescriptive, on the contrary it is a main stream to provide a general view on security evaluation. It was created by a community of security professionals and analysts, providing orientations and techniques of easy comprehension to be followed during a Pentest. As it is built by security experts, it may lack some points related to business aspects [15].

The NIST Guidelines basically follow four steps: Planning, Discovery, Atack and Report. It is considered the first one to introduce a more detailed process of report writing which indicates to write the found vulnerabilities and also the ones that were explored successfully and unsuccessfully [15].

OWASP is the result of the studies proceeded by the OWASP community. It is designed to make security software a reality, being focused on web application. The methodology is divides in three big phases: Pre-requirements and Scope of the tests, Testing Framework and Describing how the vulnerabilities are tested through code review and penetration tests [15].

Bertoglio and Zorzo [14] also recommend what they name Tramonto as a new strategy of security test recommendation. It can be used together with any of the above mentioned methodologies, being organized in five phases: Adapting, Verifying, Preparing, Executing, Finalizing.

Considering the presented scenario, the next chapter (3) will present the methodology adopted in this research as well as the research questions to achieve the set of goals.

3. METHODOLOGY

The present work aims to put together relevant information about what has been researched in the intersection between agile development and security requirements. By reaching such goal, we believe it will be possible to show the state of art in this field.

Still considering the main goal, which can also be described as delivering a subject overview and pointing to possibilities regarded to future research, this work has adopted a Systematic Mapping Study (SMS) as its research strategy.

It is clear this work represents an initial exploration related to security requirements and agile development. That is the reason the SMS strategy was chosen, for being considered suitable for this kind of research [38] [46] [47]. An SMS is considered a secondary study, since its research sources are based on other primary or even secondary studies [38] [46] [47].

Such methodology is highly recommended to get a general view of a specific topic, identifying the amount, the quality and the kind of research conducted in a research area [19] [38] [46] [47]. That is exactly what we intend to proceed in this work, providing an overview on security requirements in agile software development projects.

By using an SMS study, one can find out the structure, the reports and the results related to the researched topic. It is also possible to categorize the published works, thus creating a map of the main work performed until now. Such mapping is especially useful when there is a demand for more primary studies on a topic [19]. Figure 3.1 shows the process followed in order to reach this study goals.

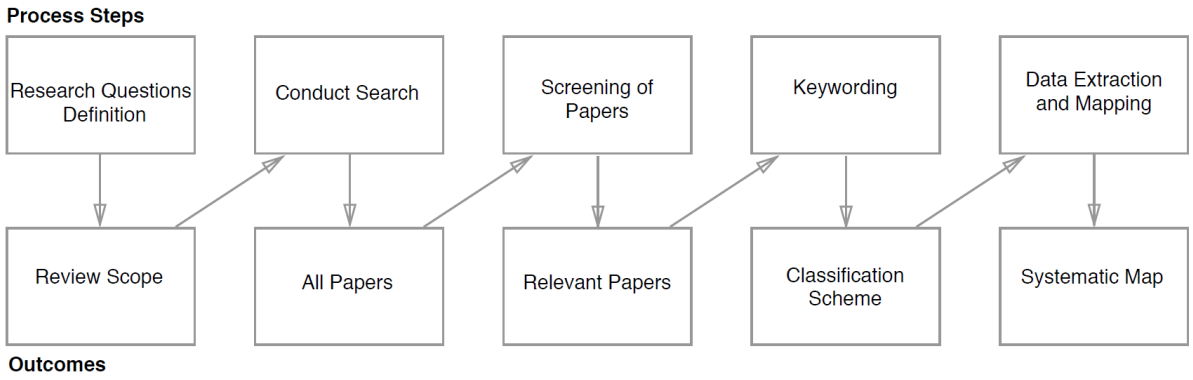

Figure 3.1 – SMS Process adapted from Petersen et al. [46] [21].

3.1 Research Questions

As an initial exploration conducted through a Systematic Mapping Study (SMS), three questions were formulated to guide us during this process. By answering these questions, we expect to reach our main goal. The three formulated questions are:

RQ1. What is being researched related to agile and security requirements?

RQ2. What are the artifacts, tools and methodologies used by agile for dealing with security requirements?

RQ3. What is the adherence degree of agile when the subject is to develop secure software?

On setting RQ1 we intend to raise the topics that are being researched related to both agile and security requirements areas. RQ2 intends to point out the main practices and tools used on dealing with security requirements inside agile environments. RQ3 intends to capture the opinions of the researchers, based on their studies, whether it is possible or not building secure software in an agile development environment. Beyond being possible or not, we intend to understand what is the compatibility degree between agile methodologies and secure software.

3.2 Strategy

We have built a search string, aiming to find the papers and studies related to the intersection between agile and security software development. The search string was set following the Kitchenham and Charters guidelines [39]. It is formed by the expected population, intervention and outcomes. We have intentionally chosen to ignore comparison and context structures, given the exploratory nature of this research. The search string is presented on Table 3.1

Based on the databases list proposed by Kitchenham and Charters [39] and considering the ones we have access, we have chosen to focus on IEEE, ACM and Science@Direct as the main research databases used in this work. These databases were also selected for supporting our search string and providing an easy and organized access to the aimed papers.

Table 3.1 – Search String

Population	(Software Engineering OR Software Development) AND
Intervention	(Agile Security OR Security Agile OR Agile Security Requirements OR Security Requirements Agile Development) AND
Outcome	(Practices OR Best Practices OR Framework OR Methodology)

Google academic was also used as a search engine to reach other research repositories. It was also used for finding the references used as a source for the main works as well as for double checking the search results.

We have also set four search criteria that acted as a filter for reducing the retrieved papers. The papers have to be available online, they have to approach security requirements in agile software development, they have to be dated between the years 1999 and 2018 and, also, they have to contain at minimum four pages. The search was applied in the entire text. Table 3.2 summarizes the presented search strategy.

Table 3.2 – Strategy Summarized

Databases and Search Engine	ACM Digital Library IEEExplore Science@Direct Google Academic
Criteria	Available online Embrace security requirements and agile From 1999 to 2018 4 pages minimum
Applied to	The ful text

Table 3.3 shows the retrieved papers per database and also the ones retrieved on Google Academic (the ones that were not found at IEEE, ACM and Science@Direct). The results represent the initial search, before applying the selection criteria.

Table 3.3 – Retrieved papers

Database	Papers
ACM Digital Library (http://dl.acm.org/)	7
IEEExplore (http://ieeexplore.ieee.org)	34
Science@Direct (http://www.sciencedirect.com/)	15
Others (via https://scholar.google.com.br/)	5
Total	61

The already presented selection criteria has excluded the studies that do not embrace security requirements and agile development together. The amount of papers after

applying the selection criteria is shown on Table 3.4, classified by the time they were published.

3.3 Screening

All the 61 retrieved papers were organized in a spreadsheet, making it possible to proceed with the screening process. The spreadsheet includes the columns as demonstrated below.

1. Database;
2. Title;
3. Year;
4. Author;
5. Abstract;
6. Keywords;
7. Duplicated (yes/no);
8. More than 3 pages (yes/no);
9. Security Requirements and Agile Development (yes/no).

After gathering all the information for the 61 papers, a filter in columns 3, 7, 8 and 9 was applied. The complete filter is shown in the following list.

- 3- Year >= 1999;
- 7- Duplicated = no;
- 8- More than 3 pages = yes;
- 9- Security Requirements and Agile Development = yes.

All articles were read (abstract, introduction and conclusion) in order to define which ones were suitable for the goals of the present research. Ultimately, in order to be considered adherent for this work, the article has to study security requirements inside agile software development. Amongst the results, some papers were chosen as the most adherent ones.

We focused on the abstract, introduction and conclusion sections. The redundant studies were removed and the references have provided us with other suitable papers. The citations of the chosen articles have also guided us on new papers to be read and evaluated.

References in the returned articles were also used, as well as the articles that have cited the retrieved articles. This procedure is known as snowballing, consisting on using the reference list and the citation list to provide other sources of research. According to Wohlin [69], snowballing consists on using the reference list of a paper or even the citations aiming to identify additional papers (new source of information). Google Academic was used to execute the snowballing process.

Our screening process has left 38 papers to be fully analyzed. Table 3.4 provides a glimpse on the distribution of these papers over the time.

Table 3.4 – Papers distribution over the time

Period	Quantity
1999 – 2002	02
2003 – 2006	11
2007 – 2010	05
2011 – 2014	14
2015 – 2018	06
TOTAL	38

All these papers were analyzed and categorized as presented in Chapter 4, on Table 4.1. At the end of the process, the three research questions were answered and the results have brought up some insights and ideas for further research.

3.4 Keywording

The 38 papers were read in order to identify their main contribution area. On doing that, it was possible to create categories in which the works were combined. That process described above is known as keywording [46].

Petersen et al. [46] have stated that keywording is a strategy used to decrease the amount of time demanded for developing a classification scheme, taking into the account the existing studies.

Beyond the abstract, the introduction and conclusion section of each paper were also read in order to guarantee a deeper and meaningful analysis. Chapter 4, on Figure 4.1, presents the classification scheme.

3.5 Mapping

By thoroughly analyzing the selected articles, we have decided to add three new columns to the previously mentioned spreadsheet. Those new columns were used to perform the data extraction and mapping process, as following.

- 10- Agile Framework;
- 11- Primary/Secondary (P/S);
- 12- Suggestion;

The column Agile Framework stands for the framework approached by the paper (Scrum, XP, etc). Primary/Secondary column indicates whether the study is a primary one or a secondary one. The last column points out the main suggested practices for dealing with security requirements in agile software development.

That new information introduced in the spreadsheet was the basis for the creation of the category scheme and consequently for the final map. Further details on the categories as well as the mapping process will be presented in Chapter 4.

In order to make it easier for the reader, a map was produced and presented at the end of Chapter 4. Figure 4.1 represents that map, summarizing the main findings of this research. Chapter 4 will bring the analysis of the selected articles.

4. DATA ANALYSIS

This chapter aims to discuss the selected articles. Following the SMS methodology, the results of each research will be briefly presented and linked to the proposed questions. The analysis will be presented grouped by research question, even though the discussion about RQ1 is always present.

- **RQ1 and RQ3:**

We start the analysis by presenting the main agile methodologies in use according to the analyzed studies. Table 4.1 shows an overview about the researched agile methodologies.

Table 4.1 – Main researched agile methodologies

Methodology	Quantity
SECURITY x SCRUM	11
SECURITY x XP (Extreme Programming)	09
SECURITY x FDD (Feature Driven Development)	03
SECURITY x AGILE IN GENERAL	15
TOTAL	38

The three main agile methodologies (XP, SCRUM and Kanban) [17] [18] are compatible with security development even inside the strictest scenarios. SDL and VAHTI are two examples of security development guides and it was found that agile can be adapted to fit that kind of guides just by introducing a few changes in the agile framework [17] [18].

The VAHTI guide mentioned before was approached in some of the studies. SCRUM was found suitable for VAHTI with some additions and modifications in its ceremonies, although it was said that the organization has to be flexible on changing SCRUM for supporting different types of projects [53]. Additional artifacts and roles added to the SCRUM provide a usable framework that is able to deal with security issues, being ready for further tailoring when needed [28].

VAHTI and agile were investigated in [51]. It is agreed that VAHTI is not directly compatible with agility, albeit some adjustments in agile are enough to develop secure software. It is important to have in mind the “definition of done" has to be unfolded in different aspects, including security [51].

Another research sets that agility, in general, is approximately fifty percent naturally compatible with security requirements treatment [31]. As shown in other articles, some adaptation is required [17] [7]. Security and quality requirements are not native considerations within any existing agile framework, tough many security practices are compatible with agile [31].

- **RQ1 and RQ2:**

A theoretical study [17] has pointed benefits on having many iterations until the final software. It also helps to improve security from the start to the end of the development process. Automated tests are recommended to increase security while the cost of rewriting was considered bigger than the added activities in the agile framework [17]. It is also mentioned that evaluation criteria and security guidelines agree on taking into account the security issues from the very beginning of the project until the end of the development. [34] [44].

In Baca et al. [6], the agile methodology was improved to create a security-enhanced software development process. It was shown, in an empirical study, that the modified agile method is particularly successful in high-need-security environments, for instance, banks [6]. Some practices may help to increase the security level such as having negative stories or even non-functional stories on the backlog, automated tests and bringing security to the definition of done [64].

The traditional security principles will endure, hence they may be taken in by agile methodologies, even though agile activities have to be extended [10]. Traditional and agile practices have been studied together in a combined way as mentioned by [16] and [17]. One important aspect is taking into account the stakeholder's information in order to make good decisions on security issues [10] [29].

Security activities should be taken into consideration by the development team, the product owner and also by the test team [7]. A risk management process is one of the security practices suggested by [30].

In terms of architecture, Chivers et al. [22] argue that an incremental security architecture is superior to a top-down one. The agile philosophy contributes to security since its requirements are refined along each new iteration, but the infrastructure team have to be included in the process of development [22].

Automated tests were discussed in a number of studies, one of them approached web-based systems should be tested in an automated way. The adopted architecture also should make such tests not only possible but even easier. It also highlights the importance of including infrastructure and security team since the beginning of the development project [62].

Another study presents a guide of recommendations aiming to enhance the security in agile projects. Amongst the suggested items are training the team on security issues, creating a role called security master, providing security skills to the testers, adopting evil stories, building a security backlog and playing a security poker game [63] [68].

Sindre and Opdahl [56] [57] have suggested eliciting security requirements by using the so called misuse cases. According to the authors, the use cases point to the desired behavior of a user. On the other hand, the misuse cases would point to the not desired

behavior, consisting on the security requirements that should be developed and tested along the project [56] [57].

Ayalew et al. [4] has raised one way for coping with security requirements, which consists on mixing together agile framework with beneficial procedures from the traditional waterfall security processes. That is agreed by Dybå et al. [27], who mentions that it is possible to combine agile project management with the traditional methods. However, pure agile methods were found not suitable for developing secure software. Invariably, other practices have to be added into agile framework in order to reach a better security level during the project development [4].

Related to SCRUM, it is suggested to add a security backlog to the already existing backlog. That is considered a SCRUM security practice [5]. Bartsch [9] argues that implicit security requirements, security expertise and security awareness are by far the most relevant items to be considered in order to improve security level in software development projects [9].

Appropriate security mechanisms such as an incremental risk analysis is considered equally relevant by Ge et al. [32], as a security practice within agile methodologies. A superior understanding on the security requirements is reached through the risk assessment over each agile iteration [32].

Extended agile activities are also proposed to SCRUM methodology [25]. For instance, a practice similar to the planning poker is raised and named as the protection poker. It can be used to create a list of security requirements, therefore its focus is security [68].

XP is another agile methodology that appeared in the evaluated studies. A research has indicated a modified XP methodology which they have called Extreme Security Engineering (XSE) as a way to secure agile software development. It has also shown that good enough security should be defined during the development process instead of being predefined even before the beginning of the project [16].

The authors of [66] corroborate [16] suggesting that XP should be modified and a security engineer may be added to the project team. Other attributes related to quality are similarly affected by agile methodologies and also require XP adaptations [18]. The native XP process is considered to be limited for supporting secure software. Based on this finding, there is an urgent demand for extending XP practices by creating new roles, practices and guidelines [33].

Feature Driven Development (FDD) also has come up as an agile methodology, linked to the security requirements. Once again, it was suggested a FDD changing in order to incorporate security processes [60]. The same conclusion was reached by another research. It was pointed out that security activities should be added, although there should be a balance between security level and the costs derived by the loss of the agility [37].

A parameter called Agility Reduction should be adjusted depending on the project and the context it is involved in [37].

The biggest challenge of developing secure software in an agile environment is much more related to people. One way to face such defiance is integrating security in the definition of done [63]. Other suggestions are making the security as a part of the acceptance criteria, creating technical stories, bringing security into functional requirements, automating tests and adding a security expert in the development team [63].

A specific security activity was frequently highlighted in different studies. It is mainly known as abuser stories [18]. Speaking about abuser stories, abuse cases were first discussed in 1999, when agile has started spreading over the software development market [41]. Abuse cases tries to anticipate on how attackers may abuse the system and are considered a light cost to pay in order to keep security requirements traceable [3] [40] [43] [45].

Abuser stories are considered a good way to foresee attack profiles, heading to security assurance [43]. One argument used to reinforce abuse cases practice is the simplicity. Abuser stories are easily comprehended and effortlessly absorbed by all the involved stakeholders [41].

- **RQ1, RQ2 and RQ3:**

The map shown in Figure 4.1 synthesizes the main conclusions within the studied papers. Some categories were created in order to group the reached conclusions. Part of the categories came directly from the columns 10-Agile Framework and 11-Primary/Secondary of the spreadsheet mentioned in Chapter 3. Some categories have emerged from the column 12-Suggestion content as it follows.

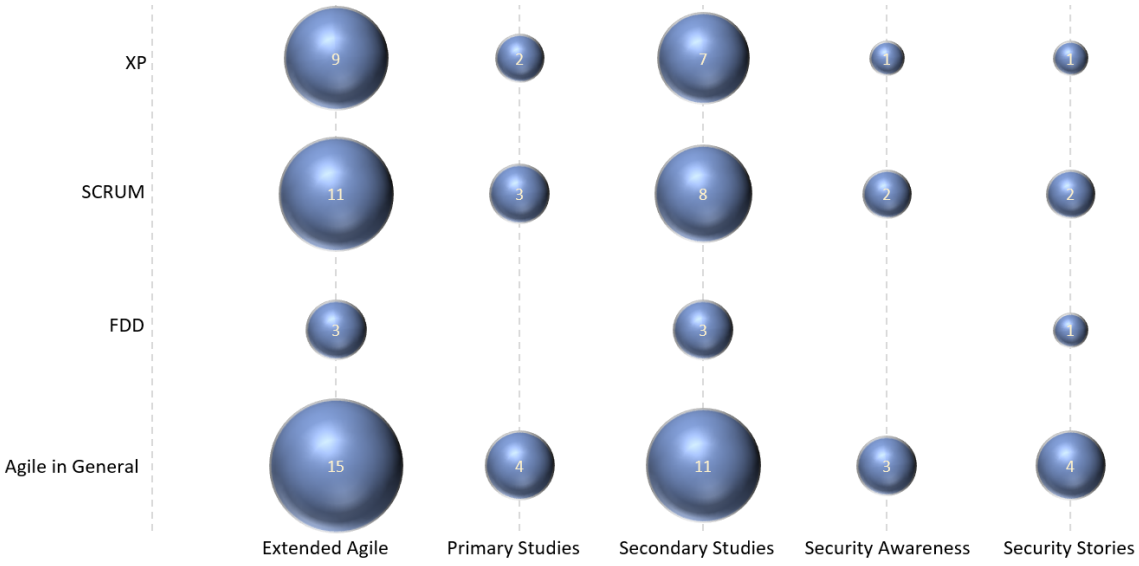

Figure 4.1 – Conclusion / Type versus Agile Methodologies

It is important to explain that Extended agile means any kind of process, activity or role introduced in the pure agile. Security awareness means any kind of training or even

security expert introduced into the development team. Security stories encloses all mentions to abuse cases, abuser stories or evil stories. After analyzing all the selected works, the next section will show the reached conclusions, answering the proposed research questions.

5. ANSWERING THE RESEARCH QUESTIONS

In this chapter we answer the proposed research questions. By answering these questions it will be possible to see some gaps we expect to research in future work.

RQ1. What is being researched related to agile and security requirements? Although there are many researches regarded to security requirements in agile software development, since the explosion of agility methodologies, there seems to be a lack of primary studies on how agile deals with security requirements.

The map presented on Figure 4.1 shows us that only nine works were classified as primary. The vast majority of the analyzed papers were considered as secondary, twenty nine out of thirty eight. The most researched agile methodologies were found to be SCRUM and XP, the first one being slightly superior in terms of adoption. It was also found that most of the papers have researched security related to agile in general, not linked to any agile framework in specific.

RQ2. What are the artifacts, tools and methodologies used by agile for dealing with security requirements? Independent on the agile methodology that is being used, the majority of the articles has recommended altering the framework in order to accommodate security activities and roles. Despite the necessity of introducing some changes, agility was found to be compatible with secure software development.

At the map on Figure 4.1, it is possible to see the articles that suggested modifying agile framework as the “Extend Agile" item. This item represents any kind of different practise, role or event introduced in the original agile frameworks (SCRUM, XP, FDD or even Agile in General).

Abuser stories, also called as evil stories, were also highlighted in many articles. In the map we have chosen the term “Security Story” to refer to abuse cases, abuse stories or evil stories. Considering the SCRUM framework as an example, user stories have to be tested as well as the security stories. The security tests were mentioned as a way to test the security stories in order to define whether they are ready (done) or not.

The importance of security training, security awareness and a security expert inside the development team are other items highly suggested. It is expressed as the “Security Awareness" item in the map. Some articles have even suggested including the infrastructure team since the very beginning of the project, including the user stories definition phase.

RQ3. What is the adherence degree of agile when the subject is to develop secure software? All articles found that agility is compatible, at some degree, with secure software development. As shown in Figure 4.1, some roles and activities have to be introduced in the agile framework in order to make it more secure.

There seem to be an agreement amongst the researchers that pure agile is not completely adherent to the security practices. What changes is the degree of adherence per-

ceived by the researchers, but some level of adaptation is always mentioned in the papers. Although some authors have suggested a modified agile framework, there is no consensus on what framework should be used.

The major limitation of the present study is the fact of being an exclusive secondary one, methodologically speaking. In spite of having such restriction, it was possible to obtain an overview of the area, given that the three proposed research questions were successfully answered. Chapter 6 presents some suggestions for possible future works. Our study also intends to be used as a data source for future research, since it summarizes many authors' point of view within the boundaries of agile development and security requirements areas.

6. CONCLUSION AND FUTURE RESEARCH

A modified agile framework seems to be the preferred way for coping with security requirements in agile software development projects. By introducing some practices, activities, events or even roles, the authors who defend such strategy claim to make agile more secure.

Many of these authors think agile frameworks should be altered in order to deal with security requirements, as it was already quoted. Some of these authors go ahead saying that an modified agile framework can be called as an extended agile framework.

Within the present context, an extended agile framework is an original agile framework with added practices for handling with security requirements. It is a possible future research suggesting some agile practices for building safer software in agile projects.

Another possible work is conducting a survey to list the most adopted practices and roles. By proceeding such research, a new secure agile framework might emerge. Such framework would be tested in actual projects in a primary study. In both cases, the results could also be crossed with the results of this SMS.

Simpler than creating a new agile framework, is suggesting some changes in a widely used agile framework, such as SCRUM or XP. Introducing or testing just a little set of practises might be enough to achieve a better understanding on the subject, pointing to another important findings.

By the present SMS results, it is also possible to state that security awareness and security stories are recommended practices for security issues in agile projects. It could be researched the influence of security awareness amongst the development team.

It is possible that just by introducing regular security training in the development team routine could leverage the security level of the released software. Software security level mediated by the security awareness (training) would be an interesting subject to be researched in this area.

Security stories are a different way of facing user stories, which is an artifact already used by agile frameworks such as Scrum. it could be useful taking a familiar concept as user stories and slightly change its focus for lightning security requirements in agile projects.

The intersection between software engineering (specifically speaking about agility) and software security provides many challenges and possibilities to be explored. As it seems to be a lack of primary studies in this area (agile and security), we recommend primary studies for future research, especially the qualitative and exploratory ones.

The items presented in Figure 4.1 could be a starting point for a new primary research. It could be adopted a modified agile methodology in a real case project. It would be better considering a range of different kinds of projects when applying some altered frame-

work adapted for dealing with security requirements. That would provide a bigger source of data that could increase the significance of the work.

By the reached results, we really think the intersection between agile and security areas is an exciting field to be researched. Software security in agile software development projects is a subject in vogue. It is even more relevant nowadays, when security is being brought into the light of the law, considering some recently released regulations such as the already mentioned GDPR and VAHTI, for instance.

REFERENCES

[1] Alsaqaf, W.; Daneva, M.; Wieringa, R. “Quality requirements in large-scale distributed agile projects–a systematic literature review”. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, 2017, pp. 219–234.

[2] Anderson, D. J. “Kanban: successful evolutionary change for your technology business”. Blue Hole Press, 2010, 262p.

[3] Asthana, V.; Tarandach, I.; ODonoghue, N.; Sullivan, B.; Saario, M. “Practical security stories and security tasks for agile development environments”, *Online, July*, 2012.

[4] Ayalew, T.; Kidane, T.; Carlsson, B. “Identification and evaluation of security activities in agile projects”. In: Nordic Conference on Secure IT Systems, 2013, pp. 139–153.

[5] Azham, Z.; Ghani, I.; Ithnin, N. “Security backlog in scrum security practices”. In: Software Engineering (MySEC), 2011 5th Malaysian Conference in, 2011, pp. 414–417.

[6] Baca, D.; Boldt, M.; Carlsson, B.; Jacobsson, A. “A novel security-enhanced agile software development process applied in an industrial setting”. In: Availability, Reliability and Security (ARES), 2015 10th International Conference on, 2015, pp. 11–19.

[7] Baca, D.; Carlsson, B. “Agile development with security engineering activities”. In: Proceedings of the 2011 International Conference on Software and Systems Process, 2011, pp. 149–158.

[8] Barbosa, D. A.; Sampaio, S. “Guide to the support for the enhancement of security measures in agile projects”. In: Agile Methods (WBMA), 2015 6th Brazilian Workshop on, 2015, pp. 25–31.

[9] Bartsch, S. “Practitioners’ perspectives on security in agile development”. In: Availability, Reliability and Security (ARES), 2011 Sixth International Conference on, 2011, pp. 479–484.

[10] Baskerville, R. “Agile security for information warfare: A call for research”, *ECIS 2004 Proceedings*, 2004, pp. 13.

[11] Beck, K. “Embracing change with extreme programming”, *Computer*, vol. 32–10, 1999, pp. 70–77.

[12] Beck, K.; Beedle, M.; Van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; et al.. “Manifesto for agile software development”. Source: http://agilemanifesto.org/, May 2018.

[13] Beck, K.; Gamma, E. “Extreme programming explained: embrace change”. addison-wesley professional, 2000, 189p.

[14] Bertoglio, D. D.; Zorzo, A. F. “Tramonto: Uma estratégia de recomendações para testes de penetração”, *XVI Simpósio Brasileiro de*, vol. 1315, 2016.

[15] Bertoglio, D. D.; Zorzo, A. F. “Overview and open issues on penetration test”, *Journal of the Brazilian Computer Society*, vol. 23–1, 2017, pp. 2.

[16] Beznosov, K. “Extreme security engineering: On employing xp practices to achieve’good enough security’without defining it”. In: First ACM Workshop on Business Driven Security Engineering (BizSec). Fairfax, VA, 2003.

[17] Beznosov, K.; Kruchten, P. “Towards agile security assurance”. In: Proceedings of the 2004 workshop on New security paradigms, 2004, pp. 47–54.

[18] Boström, G.; Wäyrynen, J.; Bodén, M.; Beznosov, K.; Kruchten, P. “Extending xp practices to support security requirements engineering”. In: Proceedings of the 2006 international workshop on Software engineering for secure systems, 2006, pp. 11–18.

[19] Budgen, D.; Turner, M.; Brereton, P.; Kitchenham, B. “Using mapping studies in software engineering”. In: Proceedings of PPIG, 2008, pp. 195–204.

[20] Cao, L.; Ramesh, B. “Agile requirements engineering practices: An empirical study”, *IEEE software*, vol. 25–1, 2008.

[21] Chanin, R.; de Sales, A. H. C.; PRIKLADNICKI, R.; POMPERMAIER, L. “A systematic mapping study on software startups education”. In: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018-EASE’18, 2018, Nova Zelândia., 2018.

[22] Chivers, H.; Paige, R. F.; Ge, X. “Agile security using an incremental security architecture”. In: International Conference on Extreme Programming and Agile Processes in Software Engineering, 2005, pp. 57–65.

[23] Coffman, E.; Galambos, J.; Martello, S.; Vigo, D. “Guide software development with data protection by design and by default”. Source: https://www.datatilsynet.no/en/regulations-and-tools/guidelines/data-protection-by-design-and-by-default/, Jun 2018.

[24] Cusumano, M. A.; Selby, R. W. “How microsoft builds software”, *Communications of the ACM*, vol. 40–6, 1997, pp. 53–61.

[25] de Andrade, M. R.; Queiroz, A. A.; de Queiroz, R. J. “Scrum-seg: Uma extensão da metodologia scrum para aplicações seguras”, *SI NFORME*, 2011, pp. 128.

[26] de Azevedo Santos, M.; de Souza Bermejo, P. H.; de Oliveira, M. S.; Tonelli, A. O. “Agile practices: An assessment of perception of value of professionals on the quality criteria in performance of projects”, *Journal of Software Engineering and Applications*, vol. 4–12, 2011, pp. 700.

[27] Dybå, T.; Dingsøyr, T. “Empirical studies of agile software development: A systematic review”, *Information and software technology*, vol. 50–9-10, 2008, pp. 833–859.

[28] Fitzgerald, B.; Stol, K.-J.; O’Sullivan, R.; O’Brien, D. “Scaling agile methods to regulated environments: An industry case study”. In: Proceedings of the 2013 International Conference on Software Engineering, 2013, pp. 863–872.

[29] Flechais, I.; Sasse, M. A.; Hailes, S. “Bringing security home: a process for developing secure and usable systems”. In: Proceedings of the 2003 workshop on New security paradigms, 2003, pp. 49–57.

[30] Franqueira, V. N.; Bakalova, Z.; Tun, T. T.; Daneva, M. “Towards agile security risk management in re and beyond.” In: EmpiRE, 2011, pp. 33–36.

[31] Ge, X.; Paige, R. F.; Polack, F.; Brooke, P. “Extreme programming security practices”. In: International Conference on Extreme Programming and Agile Processes in Software Engineering, 2007, pp. 226–230.

[32] Ge, X.; Paige, R. F.; Polack, F. A.; Chivers, H.; Brooke, P. J. “Agile development of secure web applications”. In: Proceedings of the 6th international conference on Web engineering, 2006, pp. 305–312.

[33] Ghani, I.; Yasin, I. “Software security engineering in extreme programming methodology: A systematic literature review.”, *Science International*, vol. 25–2, 2013.

[34] Goertzel, K.; Winograd, T.; McKinley, H.; Holley, P.; Hamilton, B. “Security in software life-cycle, making software development processes and software produced by them-more secure”, *Department of Homeland Security*, 2006, pp. 46.

[35] Highsmith, J. A.; Highsmith, J. “Agile software development ecosystems”. Addison-Wesley Professional, 2002, vol. 13.

[36] Howard, M.; Lipner, S. “The security development lifecycle”. Microsoft Press Redmond, 2006, vol. 8.

[37] Keramati, H.; Mirian-Hosseinabadi, S.-H. “Integrating software development security activities with agile methodologies”. In: Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on, 2008, pp. 749–754.

[38] Kitchenham, B. “What’s up with software metrics?–a preliminary mapping study”, *Journal of systems and software*, vol. 83–1, 2010, pp. 37–51.

[39] Kitchenham, B.; Charters, S. “Guidelines for performing systematic literature reviews in software engineering”, 2007.

[40] McDermott, J. “Abuse-case-based assurance arguments”. In: Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, 2001, pp. 366–374.

[41] McDermott, J.; Fox, C. “Using abuse case models for security requirements analysis”. In: Computer Security Applications Conference, 1999.(ACSAC’99) Proceedings. 15th Annual, 1999, pp. 55–64.

[42] Mead, N. R.; Allen, J. H.; Barnum, S.; Ellison, R. J.; McGraw, G. R. “Software security engineering: a guide for project managers”. Addison-Wesley Professional, 2004.

[43] Moore, A. P.; Ellison, R. J.; Linger, R. C. “Attack modeling for information security and survivability”, Technical Report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2001.

[44] Organisations, C. “Common criteria for information technology security evaluation (version 2.1)”, *Common Criteria Project Sponsoring Organisations*, 1999, pp. 1–3.

[45] Peeters, J. “Agile security requirements engineering”. In: Symposium on Requirements Engineering for Information Security, 2005.

[46] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. “Systematic mapping studies in software engineering.” In: EASE, 2008, pp. 68–77.

[47] Petersen, K.; Vakkalanka, S.; Kuzniarz, L. “Guidelines for conducting systematic mapping studies in software engineering: An update”, *Information and Software Technology*, vol. 64, 2015, pp. 1–18.

[48] Pietikäinen, P.; Röning, J. “Handbook of the secure agile software development life cycle”, *Univ. of Oulu*, 2014.

[49] PMI. “PMBOK® GUIDE Project Management Body of Knowledge 6th Edition”. Project Management Institute, 2017.

[50] Racheva, Z.; Daneva, M.; Sikkel, K. “Value creation by agile projects: Methodology or mystery?” In: International Conference on Product-Focused Software Process Improvement, 2009, pp. 141–155.

[51] Rindell, K.; Hyrynsalmi, S.; Leppänen, V. “Securing scrum for vahti.” In: SPLST, 2015, pp. 236–250.

[52] Rindell, K.; Hyrynsalmi, S.; Leppänen, V. “A comparison of security assurance support of agile software development methods”. In: Proceedings of the 16th International Conference on Computer Systems and Technologies, 2015, pp. 61–68.

[53] Rindell, K.; Hyrynsalmi, S.; Leppänen, V. “Case study of security development in an agile environment: building identity management for a government agency”. In: Availability, Reliability and Security (ARES), 2016 11th International Conference on, 2016, pp. 556–563.

[54] Schneier, B.; et al.. “Cryptanalysis of microsoft’s point-to-point tunneling protocol (pptp)”. In: Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998, pp. 132–141.

[55] SCRUM.ORG. “The home of scrum”. Source: https://www.scrum.org/, Jun 2018.

[56] Sindre, G.; Opdahl, A. L. “Eliciting security requirements by misuse case”. In: Proceedings of the 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-Pacific 2000), IEEE CS Press, 2000.

[57] Sindre, G.; Opdahl, A. L. “Eliciting security requirements with misuse cases”, *Requirements engineering*, vol. 10–1, 2005, pp. 34–44.

[58] Singhal, A.; Banati, H.; et al.. “Fisa-xp: an agile-based integration of security activities with extreme programming”, *ACM SIGSOFT Software Engineering Notes*, vol. 39–3, 2014, pp. 1–14.

[59] Singhal, A.; et al.. “Development of agile security framework using a hybrid technique for requirements elicitation”. In: *Advances in Computing, Communication and Control*, Springer, 2011, pp. 178–188.

[60] Siponen, M.; Baskerville, R.; Kuivalainen, T. “Integrating security into agile development methods”. In: System Sciences, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on, 2005, pp. 185a–185a.

[61] Sourcefire. “Agile security manifesto : 12 core principles for security for the real world”. Source: https://pt.scribd.com/document/255860042/Sourcefire-Agile-Security-Manifesto, Mar 2018.

[62] Tappenden, A.; Beatty, P.; Miller, J.; Geras, A.; Smith, M. “Agile security testing of web-based systems via httpunit”. In: Agile Conference, 2005. Proceedings, 2005, pp. 29–38.

[63] Terpstra, E.; Daneva, M.; Wang, C. “Agile practitioners’ understanding of security requirements: Insights from a grounded theory analysis”. In: 2017 IEEE 25th

International Requirements Engineering Conference Workshops (REW), 2017, pp. 439–442.

[64] Vähä-Sipilä, A. “Software security in agile product management”, 2011.

[65] VersionOne. “12th annual state of agile report”. Source: https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report, Jun 2018.

[66] Wäyrynen, J.; Bodén, M.; Boström, G. “Security engineering and extreme programming: An impossible marriage?” In: Conference on Extreme Programming and Agile Methods, 2004, pp. 117–128.

[67] Williams, L.; Cockburn, A. “Guest editors’ introduction: Agile software development: It’s about feedback and change”, *Computer*, vol. 36–6, 2003, pp. 39–43.

[68] Williams, L.; Meneely, A.; Shipley, G. “Protection poker: The new software security" game"”, *IEEE Security & Privacy*, vol. 8–3, 2010, pp. 14–20.

[69] Wohlin, C. “Guidelines for snowballing in systematic literature studies and a replication in software engineering”. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering, 2014, pp. 38.