School of Computer Sciences, Universiti Sains Malaysia

CST233 INFORMATION SECURITY & ASSURANCE

Semester II, 2019/2020

ASSIGNMENT 2

Lecturer: Assoc. Prof Dr Aman Jantan, PhD

Literature Review on Factors that Contribute to Lack of Secure Software Development in Agile Methods

Name: Muhammad Iqrar Amin

Matric: 140659

E-mail: iqraramin@student.usm.my

Table of Contents

List of Figures

List of Tables

Abstract

As the time passes, more and more organization are switching from traditional software development techniques to newer ones, where Agile Software Development sits at the top, but agile methods comes with the lack of security in their development process. In this review I have performed a systematic literature review on factors that contribute to the lack of secure software development in agile methods. I have complied in this study 15 factors which strongly effect the security in agile systems, after searching through various studies and research available online and offline. Due to the lack of studies on the topic of secure software development, agile software development and factors separating both of them, I was only able to compile 15 factors. The results of this review justify the need for new security models or alteration in the existing security models to qualify for integration of security in agile software development.

1 Introduction

. In the last few years, many companies has adapted agile methodologies for software development such as SCRUM [1], Extreme Programming (XP) [2] and Feature-Driven Development (FDD) [3]. These methods prioritize the delivery of software and services which integrates the developer productivity, product quality and customer satisfaction. Due to these aspects of agile methods it has started to take over the waterfall approaches. The agile methods are very frequent to requirement changes which makes it very feasible for customers as they can interact with system way earlier than other methods, but these methods do not practice software security engineering.

Since security is considered non-functional requirement in agile methods [4], there has always been the lack of security in agile methods. There are many research papers and studies that highlight the lack of security in agile methods and some also suggests the new ways of modeling the secure software models and revising the old practices into adapting the new ones.

This specific paper or rather literature review aims to answer the question: What are the factors that contribute to lack of secure software development in agile methods? As of now, there are some literature review that directly address this question, there is a review that highlights the challenges that developers face when developing secure software using agile approach [5] and a review on factors that influence the implementation of Secure Software Development practices [6], but not truly or totally answering the stated question. A literature review is carried out to identify, analyse, and interpret available information (relevant papers) by using appropriate methods.

In this paper, I will perform a systematic literature review on the factors that contribute to lack of security in agile software development. This study includes the analysis performed on information gathered from 35 different publications for the whole review and around 20 out 35 publications answers the research question. In Section 2, I have intruded briefly the concepts of agile software development and secure software development using SSE-CMM model as a reference, although there are several other security models, I personally like this model due to its completeness. In Section 3, I have discussed about the systematic review performed on the topic and the methodology/process used to carry put the successful review. Moving on to Section 4, I have analyzed the findings from all the publications and summarized them in tabular manner and discussed the factors in detail. Before the conclusion, I have discussed the limitations and potential impact of this study in Section 5. Finally, I have concluded my study/review in Section 6.

2 Background

This section of the review contains an overview of Secure Software Development (SSD) and overview of Agile Software Development (ASD) or Agile Software Development life cycle (ASDLC).

2.1 Overview Secure Software Development

Douglas A. Ashbaugh [7] define secure software development as: security measures in applications/software that minimize the likelihood of threats to the assets. Whereas, Goertzel et al. [8] define Secure software as: software that cannot be intentionally subverted or forced to fail; it remains correct and predictable in spite of intentional efforts to compromise that dependability.

There are several models defined/approved by ISO/IEC including Trusted CMM/Trusted Software Methodology (T-CMM, TSM), Systems Security Engineering Capability Maturity Model (SSE-CMM), Microsoft's Trustworthy Computing Security Development Lifecycle and so on [9]. But the most common and comprehensive one is Systems Security Engineering Capability Maturity Model (SSE-CMM) which is a model that can be utilized to access and advance the capabilities of an organization [9], it is depicted in Figure 2.1. The process is subdivided into three processes: Engineering Process, Assurance Process and Risk Process. The engineering process specifies security needs, provides security inputs, monitor and administer security postures and coordinate security. The assurance process verifies and validate security along with building assurance argument. Finally, the risk process access threats, vulnerabilities, impact and security risks in the system.

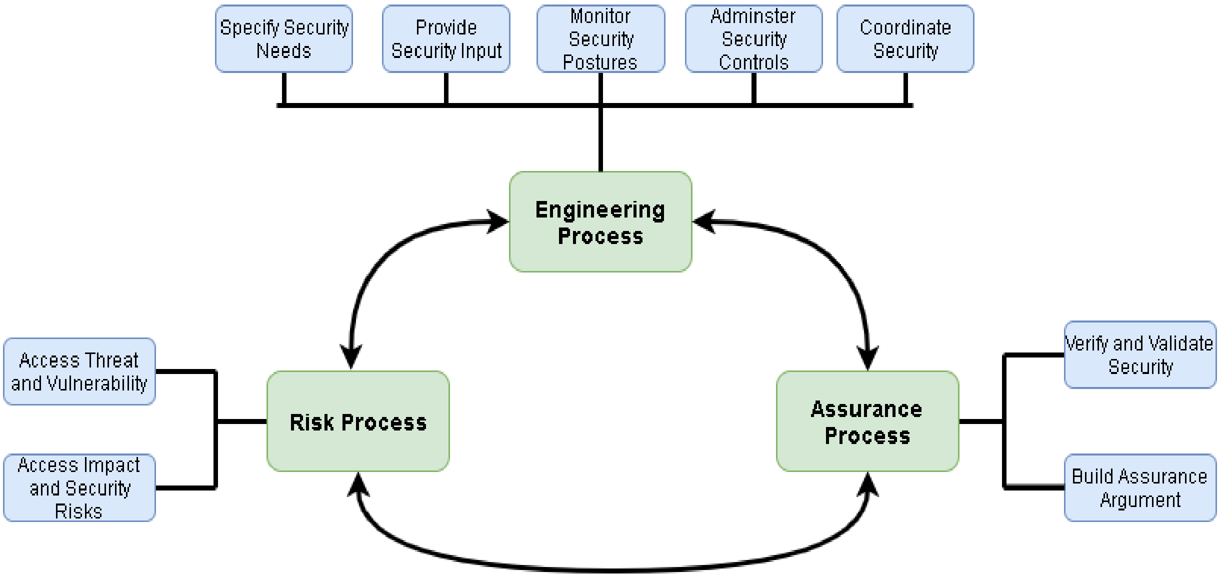

Specify Security Needs
Provide Security Input
Monitor Security Postures
Adminster Security Controls
Coordinate Security
Engineering Process
Access Threat and Vulnerability
Access Impact and Security Risks
Risk Process
Assurance Process
Verify and Validate Security
Build Assurance Argument

Figure 2.1: SSE-CMM process model for Security Engineering Process Area

2.2 Overview of Agile Software Development

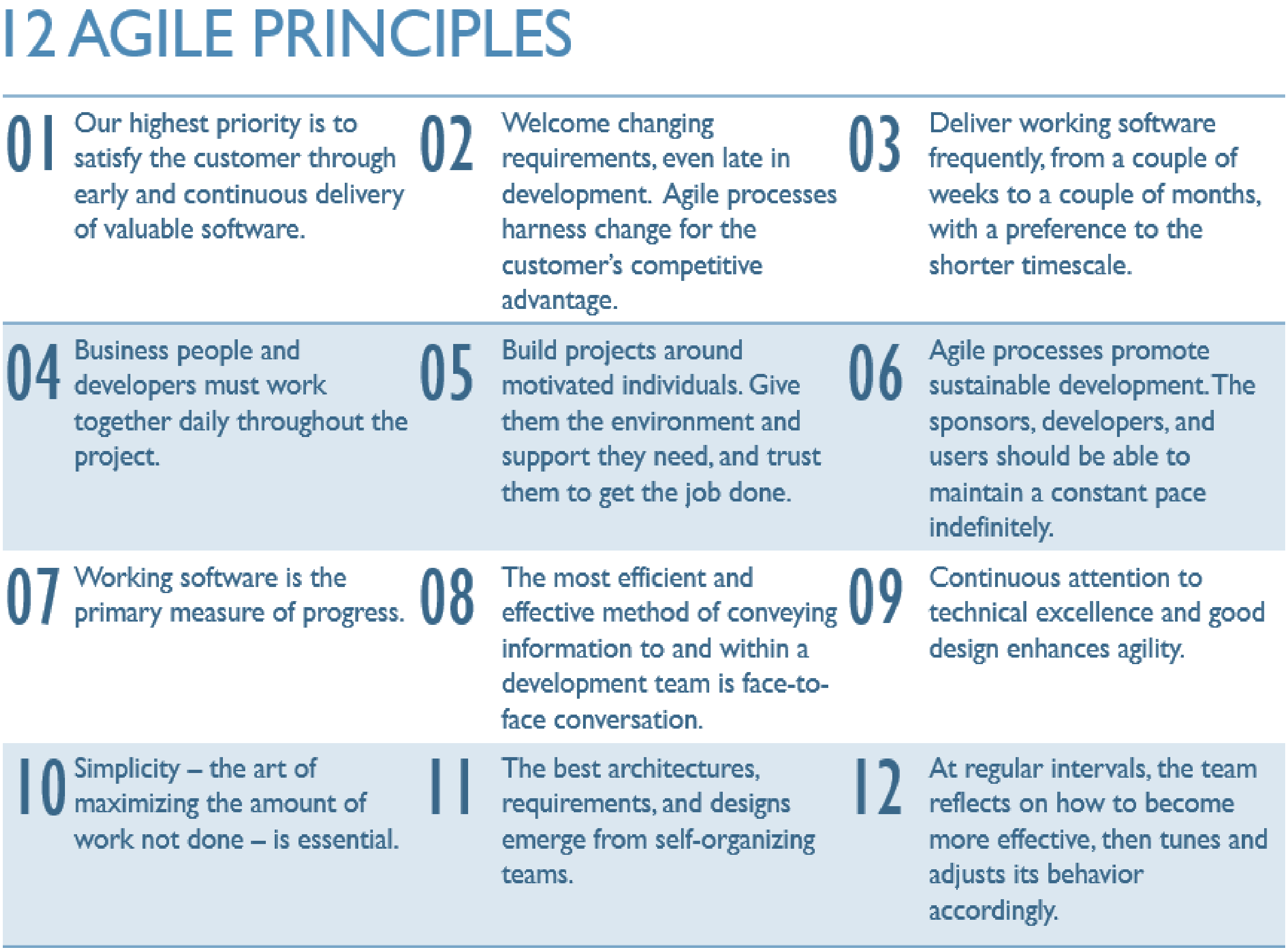
The idea of Agile Software Development was brought up by group of 17 developers in February 2001 at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, USA, when they were trying to find the common grounds between different software development models such as XP, SCRUM, DSDM, Waterfall, etc. [10]. The manifesto states that:

“*We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*

1. *Individuals and interactions over processes and tools*
2. *Working software over comprehensive documentation*
3. *Customer collaboration over contract negotiation*
4. *Responding to change over following a plan.*

That is, while there is value in the items on the right, we value the items on the left more”

They have also defined the 12 principles for the agile software development manifesto which are stated in Figure 2.2.



12 AGILE PRINCIPLES

01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

02 Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

04 Business people and developers must work together daily throughout the project.

05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

07 Working software is the primary measure of progress.

08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

09 Continuous attention to technical excellence and good design enhances agility.

10 Simplicity – the art of maximizing the amount of work not done – is essential.

11 The best architectures, requirements, and designs emerge from self-organizing teams.

12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 2.2: Twelve Principles supporting Agile manifesto

The principles and values of Agile Software Development has been utilized in other software development methodologies such as Dynamic Systems Development Method (DSDM) [11], Feature-Driven Development (FDD) [12] and Agile Scaling Model (ASM) [13]. Hence, the methods could produce shippable working software at different intervals in the development knows as iterations or cycles [13], which provide customer high flexibility in updating functional requirements in short time to develop wide variety of software such as Web Application [14].

Agile approach is often understood to be a development philosophy. But, the methods that have been used to implement the approach uses control theory mechanisms that acts as a guidance for teams to project completion [15]. First, it reduces the chance of project failure because it enables early detection of gaps between business expectations and developers understanding. Second, it enables discovery of customer needs rather than customer wishes since customers can observe demos of the product while being developed and can adapt the requirements based on their needs. Third, it enables early discovery of technical barriers since the developers experiment their ideas and use the experiment results to adapt the system architecture and work plan.

Agile software development life cycle, like other software development life cycles, consists of Plan, Design, Build, Test, and Review, but with multiple sprints instead of one single cycles. The figure showing the cycles in the Agile Methodology [16] is depicted in Figure 2.3.

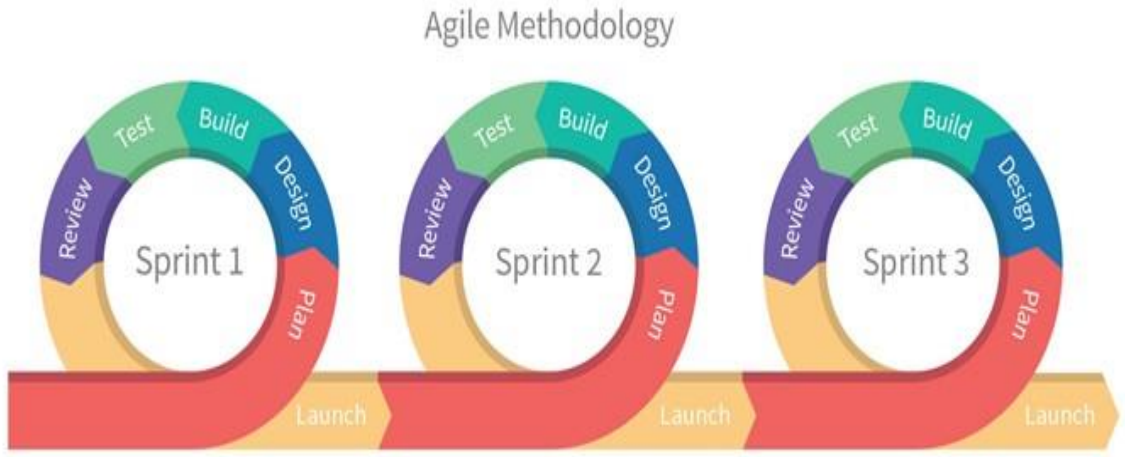


Figure 2.3: Agile Software Development Life Cycle

3 Review and Analysis Methodology

I carried out a systematic literal review to identify the factor that contribute to lack of security in agile software development. The details description of literature review and the methodology used for evaluation is provided below.

3.1 Systematic Literature Review (SLR)

I used the methods and strategies for the review proposed by Frâncila Weidt Neiva and Rodrigo Luis de Souza da Silva [17], and Kitchenham et al. [18]. Th following are the descriptions of the activities performed by me.

3.1.1 Systematic Literature Review Question

There are in total 2 research questions around which the whole SLR will circulate, the objective of this review is to identify the factor that contribute to lack of security in agile software development. Hence, I have derived the following research questions to support my research:

RQ1: What are the factors contributing to the lack of security in existing agile software development models?

3.1.2 Search Strategy

A Systematic Literature Review focuses on searching the scientific databases to get the desired information. The database utilized in this SLR search is listed below:

- IEEE Xplore
- ResearchGate
- Academy of Computing Machinery Digital Library (ACM)
- Science Direct
- Springer Link
- Others

After identifying the scientific databases, I processed with the keywords that are related to the research questions and will help to get research materials from the databases. I used Boolean operation such as “OR” and “AND” to widen the search and get most accurate studies to further do the review. The list of all the keywords are listed in the Table 3.1: Keywords used in search. There are two classification of keywords, Class 1, Class 2, and Class 3, I search all the possible combination of Class 1 AND Class 2 AND Class 3.

Table 3.1: Keywords used in search

Class	Keywords
1	“Secure Software Development” “Secure Software” “Security in Software Development” “Secure Software Development Life Cycle” “Secure Application Development” “Secure Web Application Development” “Information Security in Software Development”
2	“challenges” “lack of security” “factors effecting security”
3	“Agile Approach” “Agile Software Development” “Agile Software Development Life Cycle” “Agile Methods”

Based on the above list the following search string was generated to execute in the search engines:

((“Secure Software Development” OR “Secure Software” OR “Security in Software Development” OR “Secure Software Development Life Cycle” OR “Secure Application Development” OR “Secure Web Application Development” OR “Information Security in Software Development”) AND (“challenges” OR “lack of security” OR “factors effecting security”) AND (“Agile Approach” OR “Agile Software Development” OR “Agile Software Development Life Cycle” OR “Agile Methods”))

3.1.3 Reviewing the Search Results and Inclusion Decision

The studies were conducted on selected studies and research based on the inclusion and exclusion criteria. The inclusion criteria as the name suggests includes the research and studies on which the data extraction will be performed, while exclusion criteria includes the criteria which will be used to exclude the studies that are not related from the review. Table 3.2 shows the inclusion and exclusion criteria for selecting the appropriate studies.

Table 3.2: Inclusion and Exclusion Criteria

Type	Criteria
Inclusion Criteria	• Studies that implements SSD in agile systems • Studies that discuss about the issues and challenges in developing secure systems using agile approach
Exclusion Criteria	• Studies in languages other than English • Duplicated studies • Studies that restricts from accessing them, in other word full text is not available for them.

The search for the study was performed in four different stages depicted in Figure 3.1: Stages of the search performed in the review. In the very first stage there were around 1393 search results from all the database together, on stage 2 the duplicate results were eliminated resulting in 603 studies. In the third stage, the studies were selected based on appropriate title and abstract which reduced the count of studies from 603 to 223. Finally, a comprehensive study was carried out on the leftover studies and 35 studies were selected from them.

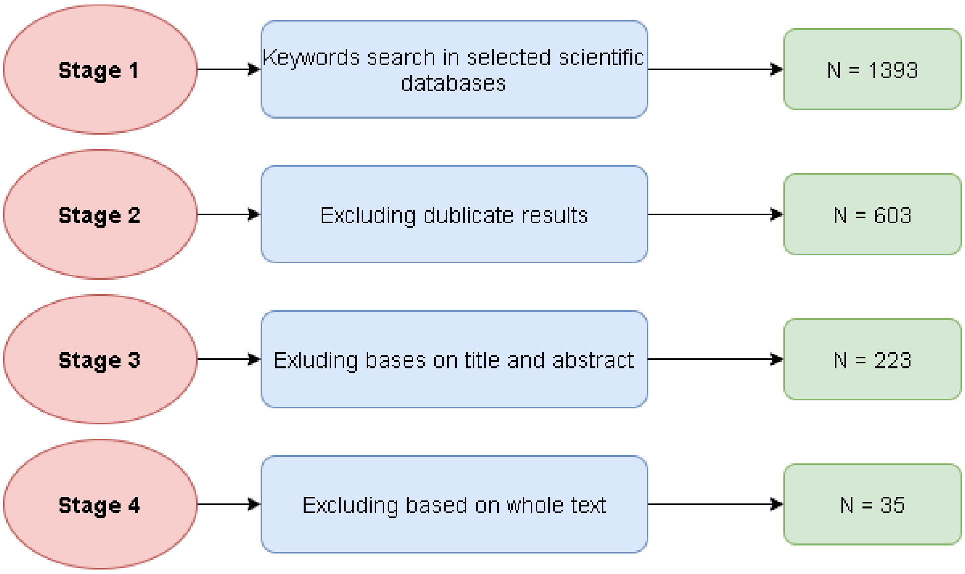

Stage 1
Keywords search in selected scientific databases
N = 1393
Stage 2
Excluding dublicate results
N = 603
Stage 3
Exluding bases on title and abstract
N = 223
Stage 4
Excluding based on whole text
N = 35

Figure 3.1: Stages of the search performed in the review.

Once the studies were handpicked after executing inclusion and exclusion criteria, a comprehensive Quality Assessment was conducted to ensure the qualitative property of the search/researched data. In this specific SLR, I have considered the following four QA questions and a marking scheme was implemented on them which was Complete (C) = 1, Partial (P) = 0.5, None (N) = 0. The following is the list of QA questions:

1. Does the study reflect the implementation of secure software development in agile development?
2. Does the study address the factors that contribute to the lack of secure software development in agile software development?
3. Does the study include any real-life experience of using secure software development in agile methods?
4. Does the study discuss on research process?

After performing QA on the studies, out of 223 studies only 35 were selected to perform comprehensive review on them and find out the factors that contribute to the lack of security in agile methods.

3.1.4 Data Extraction

After carefully selecting the studies for the review that data extraction strategy was applied to each study to gather required information from them. Table 3.3 shows the description of the components in the extraction form. After the data is extracted the comprehensive analysis was done on the gather data.

Table 3.3: Data Extraction Form

Data Extracted	Description
ID	Unique id for each preferred study
Author	Name of the author of the study
Title	Title of the respective study
Year	Year of Publication
Publisher	Publisher's name
Type	Study type (Journal. Research Report, Conference Proceedings, Book Section)
Domain	Research domain (for example Artificial Intelligence, Software Engineering)
Methodology	The technique used to write the study e.g. case study, systematic review, empirical study, interview to get data, observation
Factors	Information about the factors that are responsible for the lack of security in Agile methods.
SDLC	SDLC used in the study
Models	Secure Software Development Model used in the study

3.1.5 Results

In the decisive review, 35 studies have been selected and the distribution of the studies selected are shown in Table 3.4, most are from IEEE, Springer Link (28.5%), and ACM (14.4%).

Scientific Database	Stage 1	Stage 2	Stage 3	Stage 4	Weight
IEEE Xplore	112	68	20	10	28.5%
ResearchGate	69	30	10	4	11.4%
ACM	55	30	15	5	14.4%
Science Direct	28	21	8	2	5.8%
Springer Link	879	350	120	10	28.5%
Others	245	100	50	4	11.4%
Total	1393	603	223	35	100%

Table 3.4: Distribution of studies from various databases

4 Analysis of Finding and Discussion

In this section of the review, I will analyze my finding from the studies after in-depth and comprehensive analysis. Secure software development contains set of activities/practices that the organization must carry out in order to implement it in agile software development. But most of the developers find it very hard to implement it in even traditional software development techniques such as Waterfall, not to mention the difficulties to implement it in agile methods. This section answers the RQ1. After the systematic review of all the resources the 15 factors are generated that are the reasons for the lack of security in agile methods. The factors are listed in

Table 4.1: Factors that contribute to the lack of security in agile methods.

No.	**Factor**	**Source**
1	Security assessments require detailed documentation.	[19] [20]
2	Absence of risk assessment activities in agile development methods.	[21] [22] [23]
3	Excessive iterations and each includes the need for security related activities.	[24] [21] [22]
4	Absence of security in core principles (mentioned in Section 2.2) of agile methods.	[25]
5	Changes in the user requirements triggers the changes in the security requirement too.	[23]
6	Time taken by each iteration may not be feasible to carry out security activities.	[26] [27]
7	Lack of security awareness in customers.	[28] [26] [29] [23]
8	Lack of developer experience in secure software development.	[28] [26]
9	Security requirement as a non-functional requirement in agile methods.	[4]
10	Integration of security activities might violate the agility of the agile software development.	[30]
11	Incessant alteration in codes make the security activities troublesome to carry out.	[24] [23]
12	Misconception that focusing on security can hinder the progress of development in agile methods.	[31] [32]
13	Ineffectiveness of previous security models for software development on agile development.	[33] [34] [35] [32]
14	Additional cost of security assurance in agile methods.	[36]
15	Same individual performing the role of developer and security analyst can reduce the security.	[29] [37]

All the factors are crucial in the analysis of this review, the detailed description of these factors is discussed in the following section.

1. **Security assessments require detailed documentation:**

   According to [19], [20], six companies which were utilizing agile methods were interviewed to interrogate about the security of their software development, most of them were lacking security in their software development and the most common reason was the incapability of generating detailed documentation after each iteration of agile method.

2. **Absence of risk assessment activities in agile development methods:**

   Since Agile Software Development methods carries out development in successive iterations, these methods do not integrate risk assessment activities in them [21], [22], [23].

3. **Excessive iterations and each includes the need for security related activities:**

   Different authors [24], [21], [22], claims that some security practices needs to be repeated in each iteration of agile software development , to ensure secure software development. This discourages the developer and is especially difficult in the first place, resulting in lack of security.

4. **Absence of security in core principles of agile methods:**

   If we pay attention to the core principles stated on official site of agile manifesto [10], there is not a single reference to word “security” in the core principles of the system design and implementation, which clearly defines the lack of security in agile methods [25].

5. **Changes in the user requirements triggers the changes in the security requirement too:**

   According to Wayrynen et al. [23], changes in user requirements, which is one of the most significant and distinguishing feature in agile software development, and system design from time to time can introduce the change in security requirements too, which is often neglected or ignored by the developers. This possess a huge lack of security in agile methods.

6. **Time taken by each iteration may not be feasible to carry out security activities:**

   Since the time give to each iteration in agile software development is few weeks only, which makes difficult and sometimes impossible to implement security activities (e.g. risk assessment and security requirements elicitation) challenging because there is just not enough time [26], [27]. This is one of the major factors of absence of secure development in agile approach.

7. **Lack of security awareness in customers:**

   Woody [29], Wayrynen el at. [23], Alnatheer et al. [28] and Bartsch [26], all share the common point of view that there isn’t much awareness of security among the customer, and propose the annexation of security exerts in the teams of developer to raise awareness of security among clients and customers. This lack of awareness becomes the factor for lack of security in agile methods.

8. **Lack of developer experience in secure software development:**

   Along with the lack of awareness in customers, Alnatheer et al. [28] and Bartsch [26] also suggests that developers also lack the experience in secure software development and are not familiar enough with the activities and practices of SSD.

9. **Security requirement as a non-functional requirement in agile methods:**
   Security has always been the non-functional requirement in agile methodologies, since it is hard for an agile organization to pin it down in use case, which is the main feature in agile development [4]. Hence, lack strong security use cases prevents the security to be implemented in agile development.
10. **Integration of security activities might violate the agility of the agile software development:**
    Sharma el at. [30], while identifying and discussing different ways to integrate security in agile software development, highlights that by enforcing security in agile systems, it takes away the agility of agile development since the implementation of security at each development phase over and over again is tedious and discourage developers to focus on security.
11. **Incessant alteration in codes make the security activities troublesome to carry out:**
    According to [24], [23], changes in the codes for software being developed in agile methods is very common and frequent in each iteration as the user requirements changes and it can make the security activities (i.e. risk assessment, vulnerability assessment etc.) in the software development troublesome to carry out.
12. **Misconception that focusing on security can hinder the progress of development in agile methods:**
    Zurko [31] mentioned about the misconception of the developers that focusing on the security will hinder the progress of development in the aglie methods, although it is true to certain point but it does not mean that security should be completely ignored. Whereas, Siponena el at. [32] keeping the same view in mind as Zurko, introduced the techniques for integration of security in agile methods.
13. **Ineffectiveness of existing security models for software development on agile development:**
    The secure software development models in some of the famous existing models [33], [34] and [35] are outdated and are not adequate to be used in agile methods but are suitable for traditional methods of software development such as Waterfall. Siponena el at. [32] also pointed out that agile software needs the call for agile secure software development that offers agility to the developers to focus on security.
14. **Additional cost of security assurance in agile methods.:**
    The implementation of security in agile software development requires additional costs, although these costs are software dependent (web application, desktop application, mobile application etc.), there is still a need for more funds [36]. This factor in my opinion is quite negatable but debatable at the same time, hence I included it in the review, so the readers can decide on whether it is eligible or not.

15. **Same individual performing the role of developer and security analyst can reduce the security:**

    Both Woody [29] and Konglsi [37] firmly believe that there should be two separate individuals for different tasks of developing software and maintaining security in the system. Security should be handled by experts in security to ensure objectivity of the results. But this is often ignored by developers in agile development which plays as the factor of lack in security again in agile methods.

5 Impacts and Limitations of Review

This section discuss the impacts and limitations of this systematic literature review, the limitations include several different constraints. The impacts and limitations are discussed as follow.

5.1 Impact of review

This study delivers the literature review on the factors contributing to the secure development of software in agile methods. There is always a new research that could be formed from existing research [38]. Firstly, this review calls for the need of additional research on how to overcome these factors and introduce new models or update the existing one to make the secure software development as agile as agile software development. Secondly, this study will make it leisurelier for the new research to get information about the factors that they need to work on to improve. Lastly, this study can also act as reference for future research in “related work” section.

5.2 Limitations of the review

There are several limitations to this literature review, which includes absence of relevant studies, inaccuracy of data extraction, selection criteria, and low number of relevant papers. Some of them are common with other literature reviews [6] [39]. The description of the limitations is given below:

- **Absence of relevant studies:**
  The review is conducted on the results that came from the keywords that I used to search the databases, but there is a possibility that there are other useful papers that get ignored or didn’t share the keywords used, which could contain very useful information about the study that I conducted.
- **Inaccuracy of data extraction:**
  Since the review was conducted my myself only, with only one proofread of the data extracted from the sources, there is high chance of the inaccuracy of data that has been extracted.
- **Selection criteria:**
  The selection criteria which was used in this review, selects the resources based on their title and abstract, due to which there is chance that I missed some helpful and relevant papers in the process.
- **Low number of relevant papers:**
  Since this topic is quite new and even the concept of agile methods is rapidly adapted in the past few years, hence there just isn’t much studies about it. I was able to find only 20 studies about the issues/factors of absence of SSD in agile systems (not counting the 15 which involves the definitions and concepts).

6 Conclusion

As the software development in agile methods continues, there is constant argumentation about the security in it, and also the discussion on the difficulties when performing security in agile methods. Since agile methods prioritize the requirement changes and frequent deliveries, it often leave security in the shadows. In this review, I was able to widen my knowledge in secure software development and agile software development along with 15 factors that shows the lack of security in agile methods.

The results of this study are an evidence that developing secure software using the agile methods lack security in its core and it calls for research to develop more efficient models to adapt with the agility of agile methods. This review intends to contribute in adoption of agile development methods and the security assurance practices to enable developing secure software using the agile methods smoothly and seamlessly.

7 Acknowledgment

I would like of thank Assoc. Prof Dr Aman Jantan, for giving me such an amazing opportunity to work on this literature review study and enhance my Affective Domain skills along with Cognitive skills.

8 References

[1] K. Gaurav and B. Pradeep, "Impact of Agile Methodology on Software Development Process," *International Journal of Computer Technology and Electronics Engineering (IJCTEE),* vol. 2, pp. 2249-6343, 2012.

[2] L. Lowell and J. Ron., "Extreme Programming and Agile Software Development Methodologies," *IS Management,* vol. 2, no. 1, pp. 41-52, 2004.

[3] R. Lynn, "What is FDD in Agile?," planview, [Online]. Available: https://www.planview.com/resources/articles/fdd-agile/. [Accessed 03 June 2020].

[4] S. Vonnegut, "Six Steps to Secure Software Development in the Agile Era," Checkmarx, 20 April 2017. [Online]. Available: https://www.checkmarx.com/2017/04/20/six-steps-secure-software-development-agile-era/. [Accessed 02 June 2020].

[5] H. Oueslatil, M. M. Rahman and L. b. Othmane, "Literature Review of the Challenges of Developing Secure Software Using the Agile Approach," in *2015 10th International Conference on Availability, Reliability and Security*, 2015, pp. 540-547.

[6] S. L. Kanniah and M. N. Mahrin, "A Review on Factors Influencing Implementation of Secure Software Development Practices," *World Academy of Science, Engineering and Technology International Journal of Computer and Systems Engineering,* vol. 10, no. 8, pp. 3032-3037, 2016.

[7] D. A. Ashbaugh, Security Software Development: Assessing and Managing Security Risks, Auerbach Publications, 2009.

[8] G. K., M. Winograd, T. McKinley and H. L. H. P., "Security in the software lifecycle," August 2006. [Online]. Available: www.cert.org/books/secureswe/SecuritySL.pdf. [Accessed 01 June 2020].

[9] N. Davis, "Secure Software Development Life Cycle Processes," Software Engineering Institute Carnegie Mellon University, Pittsburgh, 2013.

[10] M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C.Martin, K. Schwaber, J. Sutherland and a. D. Thomas, "Manifesto for Agile Software Development," 2015. [Online]. Available: http://agilemanifesto.org/. [Accessed 01 May 2020].

[11] A. B. Consortium, "What is DSDM?," 2013. [Online]. Available: https://www.agilebusiness.org/page/whatisdsdm. [Accessed 01 June 2020].

[12] S. R. P. a. J. M. Felsing, A practical guide to feature-drivend evelopment, 1st ed, Prentice Hall, 2002.

[13] S. W. Ambler, "The agile scaling model (ASM): adapting agile methods for complex environments. IBM," December 2009. [Online]. Available: ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF. [Accessed 01 June 2020].

[14] C. Arsenault, "Agile Web Development - Comprehensive Overview," 3 March 2017. [Online]. Available: https://www.keycdn.com/blog/agile-web-development. [Accessed 01 June 2020].

[15] Maruping, L. M. Agarwal and V. Venkatesh, "A control theory perspective on agile methodology use and changing user requirements," in *Info. Sys. Research, vol. 20*, 2009, p. 377–399.

[16] B. Srivastava, "What is agile methodology? Disadvantage of waterfall model in software development.," 16 January 2017. [Online]. Available: https://bikeshsrivastava.blogspot.com/2017/01/part-43what-is-agile-methodology.html. [Accessed 01 June 2020].

[17] F. W. N. Silva and R. L. d. S. da, "Systematic Literature Review in Computer Science - A Practical Guide," Federal University of Juiz de Fora, November 2016.

[18] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software.," University of Durham Tech. Rep. EBSE-2007-01, Durham, UK, July 2007.

[19] M. G. Jaatun and P. H. Meland, "Special Issue on Software Security," *InternatIonal Journal of Secure Software engIneerIng*, vol. 1, no. 3, pp. 71-80, 2010.

[20] T. Nicolaysen, R. Sassoon, R. Sassoon and Martin Gilje Jaatun, "Agile Software development: the Straight and narrow path to Secure Software?," Oslo, Norway, 2010.

[21] G. Bostr¨om, J. W¨ayrynen, M. Bod´en, K. Beznosov and P. Kruchten, "Extending XP practices to support security requirements engineering," in *Proc. of the 2006 international workshop on Software engineering for secure systems*, Shanghai, China, May 2006, p. 11–18.

[22] X. Ge, R. F. Paige, F. A. Polack, H. Chivers and P. J. Brooke, "Agile development of secure web applications," in *Proc. of the 6th International Conference on Web Engineering, ser,* Palo Alto, CA, ICWE ’06, July 2006,, p. 305–312..

[23] J. Wayrynen, M. Boden and G. Bostrom, "Security engineering and extreme programming: An impossible marriage?," in *Proc. 4th Conference on Extreme Programming and Agile Methods*, Calgary, Canada: Springer, Aug. 2004, p. 117–128.

[24] K. Beznosov and P. Kruchten, "Towards agile security assurance," in *Proc. of the 2004 Workshop on New Security Paradigms, ser,* White Point Beach Resort, Canada,, NSPW 04, 2004.

[25] G. ". PAL, "Is agile's inattention to security to blame for software vulnerabilities?," GCN, 21 JUNE 2019. [Online]. Available: https://gcn.com/articles/2019/06/21/agile-security-best-practices.aspxGAURAV "GP" PAL. [Accessed 02 June 2020].

[26] S. Bartsch, "Practitioners’ perspectives on security in agile development," in *Proc. the Sixth International Conference on Availability,*, Vienna, Austria,, ARES ’11, 2011, p. 479–484.

[27] H. Keramati and S.-H. Mirian-Hosseinabadi, "Integrating software development security activities with agile methodologies,," in *Proc. IEEE/ACS International Conference on Computer Systems and Applications*, Doha, Qatar, 2008, p. 749 –754.

[28] A. Alnatheer, A. M. Gravell and D. Argles, "Agile security issues: A research study," [Online]. Available: http://esem2010.case.unibz.it/idoese/docs/alnatheer.pdf. [Accessed 02 June 2020].

[29] C. Woody, "Agile security - review of current research and pilot usage," Carnegie Mellon University, Tech. Rep, Nov. 2013. [Online]. Available: http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=70232. [Accessed 02 June 2020].

[30] A. Sharma and R. K. Bawa, "Identification and integration of security activities for secure agile development," Bharati Vidyapeeth's Institute of Computer Applications and Management, 2020.

[31] M. Zurko, Simon and R.T, "User-Centered Security," in *ACM New Security Paradigms Workshop*, Lake Arrowhead, CA., 1996.

[32] M. Siponena, R. Baskervilleb and T. Kuivalainena, "Integrating Security into Agile Development Methods," University of Oulu, Department of Information Processing Science, Oulun yliopisto, FINLAND, 2005.

[33] B. R., "The Developmental Duality of Information Systems Security," in *Journal of Management Systems. Vol. 4*, 1992, pp. 1-12.

[34] B. R., "Information Systems Security Design Methods: Implications for Information Systems Development," in *ACM Computing Surveys 25*, December 1993, pp. 375-414.

[35] M. Siponen and R. Baskerville, "A New Paradigm For Adding Security Into IS Development Methods," in *Advances in information security management & small systems security*, MA: Kluwer Academic Publishers., 2001.

[36] K. Rindell, S. Hyrynsalmi and V. Leppänen, "Busting a myth: Review of agile security engineering methods," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2015.

[37] V. Kongsli, "Towards agile security in web applications," in *Proc. 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications.*, Portland, OR, USA, Oct. 2006, p. 805–808.

[38] B. A. Kitchenham, D. Budgen and O. P. Brereton, "Using mapping studies as the basis for further research a participant-observer case study," *Information and Software Technology*, vol. 53, no. 6, p. 638 –651, 2011.

[39] T. Dyba and T. Dingsoyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50, no. 9, pp. 833-859, 2008.