SOFTWARE SECURITY ENGINEERING IN EXTREME PROGRAMMING METHODOLOGY: A SYSTEMATIC LITERATURE REVIEW

Imran Ghani
(Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
imran@utm.my)
Izzaty Yasin
(Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
izatyyasin@gmail.com)

ABSTRACT: *Agile methodology such as Extreme Programming (XP) has gained enough recognition as efficient development process by delivering software fast even under the time constrains. However, like other agile methods including Scrum, Feature Driven Development (FDD), DSDM and, XP has also been criticized because of unavailability of security element in its twelve practices. In order to have a deeper look into the matter and discover more about the reality, we conducted a systematic literature review (SLR) and studied the literature and software solutions between 2000 to 2012. Our findings highlight that the in its current form the XP model partially support integrating Software Security with its twelve practices. Although, there are a few researches on this topics but the detailed information about their usage and outcome is not yet published. Thus we conclude that the existing twelve practices of XP are not enough hence security based practices in XP need to be proposed.*

Keywords: Extreme Programming, XP, Software Security, Agile Methodologies

INTRODUCTION

Agile methodologies have had an important impact on software development practices in recent years [17]. A significant amount of positive feedback has been noted from the organizations [31,47] that practice agile methods. Their statements [7,11] suggest that agile methods help during the software development process by emphasizing rapid development. This, along with an ability to quickly respond to changes in requirements, leads to a high degree of customer satisfaction. Agile methods are more flexible and help to reduce iterations. However, they need to follow several rules related to the agile manifesto, including those concerning less documentation and team member interactions, which provide for appropriate communication with customers and other users.

On the other hand, some researchers and practitioners [7,52,43,6] have noted that, in software development, rapid development and changing requirements are not the only issues. They highlighted another critical software problem – software security [1]. In other words, the rapid development of software that is secure. Unfortunately, agile methodologies such as Scrum, FDD, DSDM and XP do not suggest or include security elements in their models. In general, the exclusion of security elements from the agile development process creates vulnerable software. This leads to reiteration in order to make the software secure, which affects the project timeline, significantly raises costs, and negatively affects customer satisfaction, which ultimately diminishes the notion of such a methodology being "agile."

In this paper, we explore both points of view in detail. We also present our own point of view on the existing agile models, methods and systems that have integrated security into XP practices. The twelve practices of XP are as follows: the planning game, small releases, system metaphor, simple design, continuous testing, refactoring, pair programming, collective code ownership, continuous integration, 40-hour work week, on-site customer, coding standards.

In order to perform a more detailed examination, this study aims to discover whether or not integration between software security and XP can be performed. The results of this study will enable researchers and practitioners to understand and identify how software security can be improved, particularly inside XP practices, while maintaining "agility."

The objectives of this paper are:

1. To review the possible software security issues that are raised while using XP practices.
2. To identify whether it is feasible to integrate security elements into XP as a whole.

Detailed discussion regarding these objectives is provided in the following sections, which are organized according to the [46] guidelines. [Section 2] presents the review process based on our defined research questions. [Section 3] explains the results. [Section 4] presents the discussion. The final section presents the conclusion and discusses possible future research.

1 Review Process

In order to perform a literature review, we decided to use SLR [12], since it helps to identify, classify, assess and understand the entirety of searched contents according to research topics. Furthermore, it is able to provide answers to the defined research questions. The next section provides further elaboration regarding the research questions defined to perform SLR.

1.1 Research Questions

In order to identify and understand the specific findings based on the topics in papers, [40] suggested five criteria known as Population, Intervention, Comparison, Outcomes, and Context (PICOC). Therefore, we use the same criteria to structure the research question as shown in Table 1.

The following research questions are defined based on PICOC.

[Q1] Have the security elements been discussed for XP and how much research has mentioned them?

Table 1: PICOC criteria and an explanation of each criterion

Criteria	*Meaning*
Population	Who or What?
Intervention	How?
Comparison	Compared to what / what is the alternative?
Outcomes	What are we trying to accomplish, improve, effect?
Context	Under what circumstances?

Table 2: PICOC for Question 1

Criteria	*Scope*
Population	Finding the number of papers that mention security inside XP
Intervention	XP and Security
Comparison	None
Outcomes	Understanding how security elements can be adopted inside an XP model
Context	Awareness

As stated in Table 2, Q1 is concerned with the number of papers found that mention security adoption inside an XP model. This question was defined as such in order to provide an understanding regarding state-of-the-art research. As the primary focus is the number of related articles, this question usually does not include a comparison (C).

[Q2] How much research mentioned security elements that could help in this study?

Table 3: PICOC for Question 2

Criteria	*Scope*
Population	Finding the amount of research mentioning security elements that could help in this study.
Intervention	Software security and XP
Comparison	None
Outcomes	Suggest security elements that could help reduce the vulnerabilities in an XP model
Context	Security

In Q2 (in Table 3), the aim is to find existing papers mentioning security elements that could help in this study. In addition to the classical limitations of XP related to security, issues such as conflict regarding SE standards with the security engineering standards [52] were considered. The findings of this research may help to highlight possible methods for reducing the existing issues in software security in other agile processes.

[Q3] Are there any particular models or frameworks that relate to secure XP methodologies?

Table 4: PICOC for Question 3

Criteria	*Scope*
Population	Finding the existing model or framework for Secured Extreme Programming methodologies.
Intervention	Secure XP methodology
Comparison	Existing secure XP model article
Outcomes	Suggest improving the existing secure XP model with a better one
Context	Availability of suitable secure agile model

According to Q3 (in Table 4), the scope of population involves finding any existing Secure XP model. It is necessary to answer this research question in order to discover whether or not a secure XP model exists. If an answer is discovered, then improvements can be made based on the existing model. These improvements, including those made to the development phase, roles, and tools, will provide more scalability to a secure agile model.

[Q4] What is the level of acceptance in a software development environment in regards to the XP model?

Table 5: PICOC for Question 4

Criteria	*Scope*
Population	Acceptance of the XP model in a real-world software development environment.
Intervention	XP methodology
Comparison	None
Outcomes	Suggest how to measure the conformity XP model that may be accepted by real life software development environment
Context	Testing, compatibility

The answer to Q4 (in Table 5) is necessary in order to measure the conformity of an XP model in a real-world software development environment. The answers to this question may help to prepare any future case studies which might be necessary to test the compatibility of security elements in real-world agile activities.

1.2 Data Sources

In this study, several data sources have been used as suggested by Emam Hossain (2009). Table 6 shows a list of the electronic database sources that have been used to discover the answers to the research questions. Among others, we included Google Scholar as an electronic database source since it is free and papers are easy to download when compared to the other sources.

Table 6: Electronic Sources

Source	URL
Springer	http://www.springerlink.com
ScienceDirect	http://www.sciencedirect.com

IEEE Xplore	http://ieeexplore.ieee.org
Google Scholar	http://scholar.google.com
Subject Matter websites	https://www.isc2.org/csslp-whitepaper.aspx http://www.asapm.org
Thesis	http://www.featuredrivendevelopment.com/ node/699

1.3 Search String

The search strings that were used in electronic database sources are as follows:
(Agile Methodologies OR Extreme Programming OR XP) AND *(Software Security OR Secure Web Applications OR Threat)*
However, in order to search the information about Secured Extreme Programming (XP), the search string was modified to:
(Extreme Programming OR XP) AND *(Software Security OR Secured OR Secure)*
In this study, the searches were performed based on a specific time period between the years 2000-2012. The range of research was limited to journals and conference proceedings. However, a few websites and related books were also considered and studied.

1.4 Selection Criteria

There were two types of filtering criteria defined for finding related sources – inclusion and exclusion. The inclusion criteria primarily targeted agile methodologies and focused on security elements of the XP practices. The exclusion criteria focused on the general concept of agility. It is important to note that the inclusion and exclusion criteria are not used simply to find papers randomly, but rather to identify appropriate research about the related topic.

2 RESULT

2.1 Findings in Appendix A

The Table 'Appendix A' shows the results of the research sources that have been found during SLR. After evaluating the papers, we could identify only 45 papers based on the inclusion criterion. The papers that were excluded did not present enough information or contained incomplete information about relevant studies. Furthermore, they did not realize the defined research questions.

2.2 A Year-wise Published Papers

Next, a review on these identified publications has been conducted. The year and number of collected publications is shown in the graph below (Figure 1).

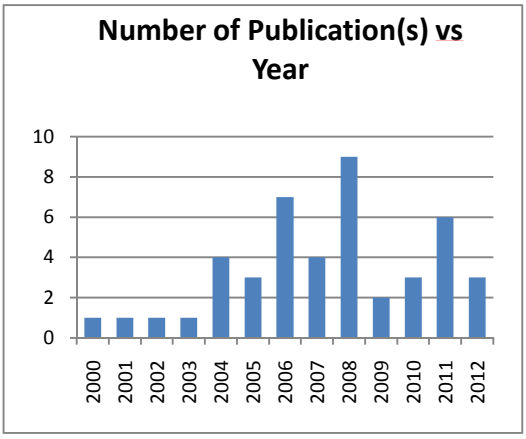

Figure 1: Number of Publications verses Year

Figure 1 illustrates that the sources were selected from the years 2000-2012. The minimum number of related publications was found from 2000 until 2003, with only one publication in each year. The year 2008 has the highest number of publications (nine). However, on average, 3.23 related publications were found per year.

2.3 Relevant Conference or Journal

Table 7 shows the list of journals and conferences papers found using qualitative analysis. Papers from ICS have the highest number of recorded collections. Book reviews follow with the second highest number. The mode is the number that is repeated more often than any other, so one out of 22 publications is the mode.
Each source has a different type of publication such as journals, proceedings and books. Seven of them are listed under journal type; six of them are book; two are from accessed type (websites) and one is from thesis type. The rest are proceedings. In order to find the right source for secure software development in XP, ICS was the most suitable choice due to the security issues being primarily related to society.

Source	Acronym	Type	Number of Publications
IEEE Software	Software	Proceeding	3
IEEE Computer Society	ICS	Proceeding	11
IEEE Symposium Computer and Information	IEEE Symposium	Proceeding	1
Lecture Note Computer Science	LNCS	Proceeding	2
Lecture Notes in Business Information Processing	LNBIP	Journal	1
IET Software	IET	Journal	1
Association for Computing Machinery	ACM	Proceeding	2

Conference of the South African Institute of Computer Scientists and Information Technologists	SAICSIT	Proceeding	1
IEEE Transaction on Dependable and Secure Computing	TDSC	Journal	1
Software Production WS		Proceeding	
Information and Software Technology	IST	Journal	3
The Journal of Systems and Software	JOSS	Journal	2
IEEE International Conference on Systems, Man, and Cybernetics	IEEE SMC	Proceeding	1
IEEE Transactions on Software Engineering	TSE	Proceeding	1
Information Systems	IS	Journal	1
Information Security Technical Report	ISTR	Journal	1
IEEE Computer and Reliability Societies		Proceeding	1
Information Technology Journal	ITJ	Journal	1
Final Project for CSCIE	CSCIE	Thesis	1
The Open Web Application Security Project	OWASP	Proceeding	1
Malaysian Conference in Software Engineering	MySEC	Proceeding	1
Book Review	Book	Book	6
Website		Accessed	2

Table 7: Journal and Conference Papers Found

3 DISCUSSION

According to the research questions that have been defined in [Section 3], this section discusses the findings based on the selected sources as shown in Table 7 above.

Q1: Has software security been proposed for XP and how much research makes mention of it?

In order to find an answer to Q1, we studied five papers: S44, S30, S36, S29 and S45. Some of those papers, such as S30, S29 and S45 explain the development of secure software using XP practices, partially discussing the issues that occur during the integration of security into the XP model.

Based on this study, S44 came to a conclusion based primarily on theoretical analysis that made mention of perspective among each author in regards to issues concerning secure forms of XP. On the contrary, a detailed discussion on the importance of integrating security into XP practices is presented by S30 in a paper titled "Improved Extreme Programming Methodology with inbuilt security." This paper displays an improved XP framework that is more structured and adopts security elements into XP practices. In order to prevent vulnerabilities at an early phase, risk assessment has been performed in an iterative fashion, as suggested by S1.

In addition, there are several papers that lead security on agile methodologies, such as S7, S15 and S21. The majority of papers, such as S2, S4, S6, S8, S9, S12, S13, S18, S20, S22, S23, S24, S28, S32, S34, S35, S39 and S40 are only about the detailed introduction of agile and XP.

Q2: How much research mentioned security elements that could help in this study?

Based on Question 2, around 18 papers were found that mentioned security elements that could help in securing the XP model. These included S1, S3, S5, S10, S11, S15, S16, S19, S27, S31, S37, S42, and S43. Most of them focus on issues that occur during the SE development process that introduce vulnerabilities into the system, such as SQL injection, Cross-site Scripting (XSS), attack trees and unsecured architecture design.

Other cases, such as S29, S30, S36 and S45 claim that security elements cannot be suitably integrated into the XP process. For example, XP does not prioritize design documentation and its detailed interface specifications are not compatible with the security engineering standard.

However, to get a better understanding of what software security all about, the explanation in S1, S42 helps to highlight the issues that occur during each development phase. Furthermore, it discusses solutions on how to prevent vulnerabilities and threats during the development process.

Q3: Are there any particular models or frameworks that relate to secure XP methodologies?

In regards to Question 3, only one recent paper—S30, published at the 2011 IEEE conference—presents a proposed framework for secure web application development in an XP model. However, according to this paper the framework was still in progress. Based on S30, the creation of a new XP framework suitable for security standards is necessary to ensure user satisfaction. This method aims to prevent and detect vulnerabilities during the initial phase by implementing misuse cases during the requirement and design stage, while performing iterative risk assessment.

However, this secure XP model based on S30 can be used as a guideline for future research in terms of improving the security issues in XP during each phase of development.

Q4: What is the level of acceptance for the XP model in a real-world software development environment?

Information regarding the level of acceptance for the XP model in an SE environment can be found in case studies S9, S24, and S41. These papers claim that XP practices are not enough to cover all the development processes. This is because some practices are suitable for favorable environments at a company with the necessary cultural aspects. However, some may encounter hostile cultural aspects during implementation.

S14, S36 and S39 primarily mention the adoption of XP among undergraduate students. In addition, S36 makes a recommendation for making XP more secure based on a case study conducted during a classroom discussion. This case study begins by suggesting ideas for how XP can be made more secure after students have applied the twelve practices to the XP process in their group projects.

Meanwhile, S17, S26 and S38 propose the acceptance of agile in real-world SE teams. S3 and S31 discuss the acceptance of software security in the life cycle and focus on preventing issues related to real systems. A similar Malaysian case study is discussed in S25.

Based on the discussion and the answered research questions, it can be concluded that it is feasible to integrate security elements in the XP process that will bring benefits to the development teams. In doing so, vulnerabilities will be reduced during the system development process, instead of being corrected during the testing or maintenance phases.

Table 8: Number of papers based on the questions

Question	Number of related papers (Y)	Number of unrelated papers (N)
[Q1] Have the security elements been discussed for XP and how much research have they been mentioned in?	6	39
[Q2] How much research mentioned security elements that could help in this study?	18	27
[Q3] Are there any particular models or frameworks that relate to secure XP methodologies?	2	43
[Q4] What is the level of acceptance in a software development environment regarding the XP model?	3	42

Table 8 shows the total number of papers based on the research questions. Referring to Appendix A, these collected papers were categorized according to four questions that were formed to locate related papers.

As depicted, Q2 holds the highest number of papers (18). Q3 held the lowest number of papers (2). The majority of papers found were not related to Q2, as these primarily discussed basic XP security, and only partially proposed security issues. However, these unrelated papers helped to provide a deeper understanding and provided fresh ideas for future research.

4 Conclusion and Future Research

Based on this SLR, we can conclude that some researchers do agree that XP is compatible with software security engineering. However, as such research is still at the beginning stages, specific ideas have yet to be proposed.

Some researchers, however, do not agree that security elements could be fully integrated into XP. It is believed that such a risky endeavor could affect the entire software process. For example, the XP development process does not encourage intensive documentation, whereas security artifacts such as misuse cases, attack trees, attack patterns and secure architecture design are necessary for risk assessment. These concerns become even more important if specific activities, such as design documentation and architecture, are not clear and necessitate re-iteration. This means that detailed design artifacts are important for software engineering standards, for a solid understanding of the flow of a system process, and for vulnerability detection during implementation.

Even so, there have been some attempts to integrate XP and security and reduce issues that occur during the process. We noticed that, if software is created with security qualities but no proper security management, the software development teams would experience terrible results. On the other hand, the software system may be developed and delivered on-time but not equipped with security elements.

Based on the findings of this research, we have come to the conclusion that the existing XP process has limitations in supporting secure software engineering practices. Pair--programming, coding standards, and refactoring help secure coding during the implementation phase. However, as it is not equipped with security elements overall, this does not provide any guidelines towards secure software engineering. Based on these findings there is a need to propose security practices, roles and guidelines in order to extend existing agile XP practices.

ACKNOWLEDGEMENTS

This research has been supported by the research project grant Vot No. 4S028, approved by Ministry of Science, Technology and Innovation (MOSTI), Malaysia.

REFERENCES

[1]Allen, J.H., Barnum, S., Ellison, R.J., McGraw, G., Mead, N.R.,Software Security Engineering: A Guide for Project Managers, 1st Edition. Pearson Education, Inc (2008)

[2] Alshayeb, M., Li, W.A.,Empirical study of relationships among extreme programming engineering activities, Information and Software Technology 48 (2006), 1068–1072.

[3] Ampatzoglou, A., Stamelos, I.,Software engineering research for computer games: A systematic review, Journal Information and Software Technology (2010), 888–901.

[4] Ann, E.K.S., McGraw, G.,Interview: Software Security In The Real World, IEEE Computer Society (2010), 47-53.

[5]Arisholm, E.,Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise, IEEE Transactions on Software Engineering (2007), Vol. 33.

[6] Azham, Z., Ghani, I., Ithnin, N.,Security Backlog in Scrum Security Practices, 5th MySEC (Malaysian Conference in Software Engineering)(2011).

[7] Azim, A.S., Amir, S.S., Shams, F.,Embedding Architectural Practies into Extreme Programming, 19th Australian Conference on Software Engineering (2008), 310-319.

[8] Bartsch, S.,Practitioners' Perspectives on Security in Agile Development, Sixth International Conference on Availability, Reliability and Security (2011), 479-484.

[9] Beck, K.,Extreme programming explained: Embrace change, 2nd. edition, Boston: Addison-Wesley, London.

[10] Bin, X., Xiaohu, Y., Zhijun, H.,Extreme Programming In Reducing The Rework Of Requirement Change, IEEE (2004), 1567-1570.

[11] Breivold, H.P., Sundmark, D., Wallin, P., Larsson, S.,What Does Research Say About Agile and Architecture, Fifth International Conference on Software Engineering Advances (2010), 32-37.

[12] Brereton, P., Kitchenham, B., Budgen, D., Turner M., Khalil M.,Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software (2007), 571–583.

[13] Cao, L., Mohan, K., Xu, P., Ramesh, B.,How Extreme does Extreme Programming Have to be? Adapting XP Practices to Large-scale Projects, Proceedings of the 37th Hawaii International Conference on System Sciences (2004), 1-10.

[14] Chess, B., Arkin, B.,Software Security in Practice, The IEEE Computer And Reliability Societies (2011), 89-92.

[15] Dalton, M., Kannan, H., Kozyrakis, C.,Raksha: A Flexible Information Flow Architecture for Software Security, ACM: San Diego (2007), California, USA, 482-493.

[16] Dudziak, T.,eXtreme Programming: An Overview, Methods and tools for software production WS 1999/2000, 1-28.

[17] Dyba, T., Dingsoyr, T.,Empirical studies of agile software development: A systematic review, Information and Software Technology (2008), 833–859.

[18] Dyba, T., Dingsoyr, T.,What Do We Know about Agile Software Development?, International Conference on Software Quality (2009), 6-9.

[19] Endicott-Popovsky, B.,Adopting Extreme Programming on a Graduate Student Project, IEEE Workshop on Information Assurance and Security (2005), 454-455.

[20] Erdogan, G., Hakon, P.M., Mathieson, D.,Security Testing in Agile Web Application Development – A Case Study Using the EAST Methodology, XP 2010, LNBIP 48 (2010), 14-27.

[21] Ge, X., Paige, R.F., Polack, F., Brooke, P.,Extreme Programming Security Practices, XP 2007, LNCS 4536, 226-230.

[22] Halkidis, S.T., Tsantalis, N.,Architectural Risk Analysis of Software Systems Based on Security Patterns, IEEE Transactions On Dependable And Secure Computing (2008), Vol. 5, 129-142.

[23] Hansson, C., Dittrich, Y., Gustafsson, B., Zarnak, S.,How agile are industrial software development practices?, The Journal of Systems and Software (2006), 1295–1311.

[24] Hossain, E., Ali, M.B., Paik, H.Y.,Using Scrum in Global Software Development: A Systematic Literature Review, Fourth IEEE International Conference on Global Software Engineering (2009), 175-184.

[25] Hunt, J.,Agile Software Construction, 1st Edition. Springer_Verlag, London.

[26] Jalali, S., Wohlin, C.,Agile Practices in Global Software Engineering – A Systematic Map, International Conference on Global Software Engineering (2010), 45-54.

[27] Kainerstorfer, M., Sametinger, J., Wiesauer, A.,Software Security for Small Development Teams – A Case Study, ACM: Ho Chi Minh City, Vietnam (2011), 305-310.

[28] Keramati, H., Hassan, S.M.H.,Integrating Software Development Security Activities with Agile Methodologies, p.749-754.

[29] Khramtchenko, S.,Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments, Thesis: Final Paper for CSCIE-275: Software Architecture and Engineering Course. Harvard University, US.

[29] Kumar, P.G.,Build your Project using Extreme Programming, Available from http://www.asapm.org. [Assesed on Sept. 19, 2012]

[30] Livermore, J.A.,What Elements of XP Are Being Adopted by Industry Practitioners?, IEEE Software (2006), 149-152.

[31] Liza, A.A., Andrew, M.G., Gary, B.W.,Emergence of Agile Methods: Perceptions from Software Practitioners in Malaysia, IEEE AgileIndia (2012), 30-39.

[32] Mchugh, O., Conboy, K., Lang, M.,Agile Practices: The Impact on Trust in Software Project Teams, Articles on Computer Sciences (2012), 71-76.

[33] Mouratidisa, H., Giorginib, P.,Attack Testing (SAT)—testing the security of information systems at design time, Journal of Information Systems (2007), 1166–1183.

[34] Muller, M.M. "A preliminary study on the impact of a pair design phase on pair programming and solo programming, Information and Software Technology 48 (2006), 335–344.

[35] Musa, B.S., Norita, M.N., Hassan, M.S.,Secure E-commerce Web Development Framework, Information Technology Journal 10 (2011), 4,769-778.

[36] Musa, B.S., Norwawi, N.M., Hasan, M.S., Yetim, K.S.,Improved Extreme Programming Methodology with Inbuilt Security, IEEE Symposium on Computers & Informatics (2011), 674-679.

[37] Paul, M. Software Security: Being Secure in an Insecure World. Available from https://www.isc2.org/csslp-whitepaper.aspx. [Assesed on Ogos 22, 2012].

[38] Paulk, M.C. "Extreme Programming from a CMM Perspective, IEEE Software (2001), 19-26.

[39] Payne, J.,Integrating Application Security into Software Development, IEEE Computer Society (2010), 6-9.

[40] Pearman, G., Goodwill, J.,Pro. NET 2.0 Extreme Programming, Greg Pearman and James Goodwill, US.

[41] Petticrew, M., Roberts, H.,Systematic Reviews in the Social Sciences: A Practical Guide, Oxford (UK): Blackwell.

[42] Qumer, A., Henderson-Sellers, B.,An evaluation of the degree of agility in six agile methods and its applicability for method engineering, Conference on Information and Software Technology (Henderson-Sellers2008), 280–295.

[43] Richard, G.E.,Getting Students to Think About How Agile Processes Can Be Made More Secure, 21st Conference on Software Engineering Education and Training (2008), 51-58.

[44] Saglietti, F., Oster, N., Pinte, F.,White and grey-box verification and validation approaches for safety- and security-critical software systems, Information security technical report (2008), 10 –16.

[45] Salo, O., Abrahamsson, P.,Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum, IET Software (2008), 58–64.

[46] Sedek, K.A., Sulaiman, S., Omar, M.A.,A systematic literature review of interoperable architecture for e-government portals, Malaysian Conference in Software Engineering (2011), 82-87.

[47] Slaten, K.M., Droujkova, M., Berenson, S.B., Williams, L., Layman, L.,Undergraduate Student Perceptions of Pair Programming and Agile Software Methodologies: Verifying a Model of Social Interaction, Proceedings of the Agile Development Conference.

[48] Theunissen, M.W.H., Derrick, G., Kourie, Watson, B.W.,Standards and Agile Software Development, Proceedings of SAICSIT (2003), 178-188.

[49] Tolfo, C., Sidnei, R.W.,The influence of organizational culture on the adoption of extreme programming, The Journal of Systems and Software 81 (2008), 1955-1967.

[50] Viega, J., McGraw, G.,Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley, London.

[51] Vries, S.D.,Security Testing Web Applications throughout Automated Software Tests, OWASP Europe Conferences (2006), 1-15.

[52] Wäyrynen, J., Bodén, M., Boström, G.,Security Engineering and eXtreme Programming: An Impossible Marriage?, Forum on Stockholm University/Royal Institute of Technology (2004), 117–128.

APPENDIX A. SYSTEMATIC REVIEWS ON EXTREME PROGRAMMING

ID	Author	Year	Title	Type	Q1	Q2	Q3	Q4
S1	Allen et al.	2008	Software Security Engineering: A Guide for Project Managers	Book	N	Y	N	N
S2	Alshayeb and Li	2006	An empirical study of relationships among extreme programming engineering activities	Journal	N	N	N	N
S3	Ann and McGraw	2010	Interview: Software Security In The Real World	Proceeding	N	Y	N	N
S4	Arisholm	2007	Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise	Proceeding	N	N	N	N
S5	Azham et al.	2011	Security Backlog in Scrum Security Practices	Proceeding	N	Y	N	N
S6	Azim et al.	2008	Embedding Architectural Practices into Extreme Programming	Proceeding	N	N	N	N
S7	Bartsch	2011	Practitioners' Perspectives on Security in Agile Development	Proceeding	N	Y	N	N
S8	Bin et al.	2004	Extreme Programming In Reducing The Rework Of Requirement Change	Proceeding	N	N	N	N
S9	Cao et al.	2004	How Extreme does Extreme Programming Have to be? Adapting XP Practices to Large-scale Projects	Proceeding	Y	N	N	Y
S10	Chess and Arkin	2011	Software Security in Practice	Proceeding	N	Y	N	N
S11	Dalton et al.	2007	Raksha: A Flexible Information Flow Architecture for Software Security	Proceeding	N	Y	N	N
S12	Dudziak	2000	eXtreme Programming: An Overview	Book	N	N	N	N
S13	Dyba and Dingsoyr	2009	What Do We Know about Agile Software Development?	Proceeding	N	N	N	N
S14	Endicott-Popovsky	2005	Adopting Extreme Programming on a Graduate Student Project	Proceeding	N	N	N	N

S15	Erdogan et al.	2010	Security Testing in Agile Web Application Development – A Case Study Using the EAST Methodology	Journal	N	Y	N	N
S16	Halkidis and Tsantalis	2008	Architectural Risk Analysis of Software Systems Based on Security Patterns	Journal	N	Y	N	N
S17	Hansson et al.	2006	How agile are industrial software development practices?	Journal	N	N	N	N
S18	Hunt	2006	Agile Software Construction	Book	N	N	N	N
S19	Kainerstorfer et al.	2011	Software Security for Small Development Teams – A Case Study	Proceeding	N	Y	N	N
S20	Kent Beck	2005	Embracing Change with Extreme Programming	Book	N	N	N	N
S21	Keramati and Hassan	2008	Integrating Software Development Security Activities with Agile Methodologies	Proceeding	N	N	N	N
S22	Khramtchenko	2004	Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments	Thesis	N	N	N	N
S23	Kumar	2009	Build your Project using Extreme Programming	Accessed	N	N	N	N
S24	Livermore	2006	What Elements of XP Are Being Adopted by Industry Practitioners?	Proceeding	N	N	N	Y
S25	Liza et al.	2012	Emergence of Agile Methods: Perceptions from Software Practitioners in Malaysia	Proceeding	N	N	N	N
S26	Mchugh et al.	2012	Agile Practices: The Impact on Trust in Software Project Teams	Proceeding	N	N	N	N
S27	Mouratidisa and Giorginib	2007	Attack Testing (SAT)—testing the security of information systems at design time	Journal	N	Y	N	N
S28	Muller	2006	A preliminary study on the impact of a pair design phase on pair programming and solo programming	Journal	N	N	N	N
S29	Musa	2011	Secure E-commerce Web Development Framework	Journal	Y	Y	Y	N
S30	Musa et al.	2011	Improved Extreme Programming Methodology with Inbuilt Security	Proceeding	Y	Y	Y	N
S31	Paul	2012	Software Security: Being Secure in an Insecure World	Accessed	N	Y	N	N
S32	Paulk	2001	Extreme Programming from a CMM Perspective	Proceeding	N	N	N	N

S33	Payne	2010	Integrating Application Security into Software Development	Proceeding	N	N	N	N
S34	Pearman and Goodwill	2006	Pro. NET 2.0 Extreme Programming	Book	N	N	N	N
S35	Qumer and Henderson-Sellers	2008	An evaluation of the degree of agility in six agile methods and its applicability for method engineering	Journal	N	N	N	N
S36	Richard	2008	Getting Students to Think About How Agile Processes Can Be Made More Secure	Proceeding	Y	Y	N	Y
S37	Saglietti et al.	2008	White and grey-box verification and validation approaches for safety- and security-critical software systems	Journal	N	Y	N	N
S38	Salo and Abrahamsson	2008	Agile methods in European embedded software development organizations: a survey on the actual use and usefulness of Extreme Programming and Scrum	Journal	N	N	N	N
S39	Slaten et al.	2005	Undergraduate Student Perceptions of Pair Programming and Agile Software Methodologies: Verifying a Model of Social Interaction	Proceeding	N	N	N	N
S40	Theunissen et al.	2003	Standards and Agile Software Development	Proceeding	N	N	N	N
S41	Tolfo and Sidnei	2008	The influence of organizational culture on the adoption of extreme programming	Journal	N	N	N	Y
S42	Viega and McGraw	2002	Building Secure Software: How to Avoid Security Problems the Right Way	Book	N	Y	N	N
S43	Vries	2006	Security Testing Web Applications throughout Automated Software Tests	Proceeding	N	Y	N	N
S44	Wäyrynen et al	2004	Security Engineering and Extreme Programming: An Impossible Marriage?	Proceeding	Y	N	N	N
S45	Xiaocheng Ge et al.	2007	Extreme Programming Security Practices	Proceeding	Y	Y	N	N