SECURE SOFTWARE DEVELOPMENT PRACTICE SELECTION MODEL

SRI LAKSHMI A/P KANNIAH

UNIVERSITI TEKNOLOGI MALAYSIA

SECURE SOFTWARE DEVELOPMENT PRACTICE SELECTION MODEL

SRI LAKSHMI A/P KANNIAH

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia

DECEMBER 2020

DEDICATION

This thesis is dedicated to my parents, especially my father, for his encouragement and motivation to take up this challenge. It is also dedicated to my husband and son for their love and support.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Prof. Madya Dr. Mohd Naz'ri bin Mahrin, for encouragement, guidance, critics and friendship. I am also very thankful to my co-supervisor Dr. Jayaletchumi a/p T. Sambantha Moorthy for her guidance, advice and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

I would also like to express my gratitude to my close friend Dr. Mariayee Doraisamy and Dr. Nur Azaliah binti Abu Bakar for their insightful comments and encouragement. Their contribution has predominantly influenced my focus and motivation for this research. I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my husband and my son for their love, patience, sacrifice and continuous support to complete this research work. Also I would like to acknowledge and thank my family members for their support.

I profusely thank the staffs at Department of Advanced Informatics, and co-researchers at UTM who have helped me in many ways to complete this research. I would also like to extend my gratitude to Jabatan Perkhidmatan Awam (JPA), Malaysia for the opportunity and funding my study. My sincere appreciation is extended to MAMPU, Malaysia for their cooperation during data collection and evaluation.

ABSTRACT

Developing secure software is critical for organizations as highly-sensitive and confidential data are transacted through online applications. Insecure software can lead to loss of revenue and damage to business reputation. Although numerous methods, models and standards in regards to secure software development have been established, implementation of the whole model is quite challenging as it involves cost, skill, and time. Moreover, lack of knowledge and guidance on selection of suitable secure development practices becomes a challenge for project managers. On that account, this thesis developed a model which aims to guide the project managers to select secure software development practices based on the factors fulfilled by the project. Initially, a systematic literature review (SLR) was conducted, and as a result 18 influential factors were identified. To strengthen and enhance these findings, semistructured interviews were conducted with 21 software development experts from eight IT departments in Malaysian public sector, and 18 influential factors emerged from the interviews. The findings from both the SLR and interviews were consolidated, and analysed using the grounded theory techniques. As a result, 20 influential factors were finalized and grouped into four main categories that influenced software development outcomes: institutional context, software project content, people and action, and development processes. To assess the fulfilment of each factor, assessment criteria to determine the fulfilment of the factors were identified using secondary data analysis method. Subsequently, secure development practices which were suitable for the Malaysian public sector were identified through a survey, and as a result 24 practices were identified. The identified factors, assessment criteria, and practices were validated using the Delphi method, involving ten experts. In addition, the experts mapped the influential factors to each secure software development practice. As a result of the Delphi method which involved three phases, the lists of validated factors and assessment criteria were produced. Additionally, a list of practices mapped with the related influential factors was produced. The validated elements were used to formulate the Secure Software Development Practice Selection Model. The proposed model was finally evaluated using a multiple case study method that involved four software development projects in the Malaysian public sector. The project managers were provided with questionnaire to assess the fulfilment of factors, and identify practices that can be incorporated in their software development project. Thus, with the proposed Secure Software Development Practice Selection Model, suitable secure software development practices can be effectively identified by assessing the influential factors fulfilled by the software project. Furthermore, the average System Usability Scale score obtained for all agencies was 70.7; thus Secure Software Development Practice Selection Model was perceived to have ‘good’ usability which corresponds to the adjective scale. In sum, there are four significant contributions of this research: a validated list of factors influencing secure software development, a list of assessment criteria for the factors, mapping of secure software development practices with the influential factors, and evaluated Secure Software Development Practice Selection Model.

ABSTRAK

Membangunkan perisian yang selamat adalah penting bagi organisasi kerana data yang sangat sensitif dan sulit ditransaksi menerusi aplikasi atas talian. Perisian yang tidak selamat boleh menyebabkan kehilangan hasil dan kemudaratan kepada reputasi perniagaan. Walaupun banyak kaedah, model dan piawaian dalam hal pembangunan perisian yang selamat telah diwujudkan, pelaksanaan keseluruhan model agak mencabar kerana melibatkan kos, kemahiran dan masa. Selain itu, kekurangan pengetahuan dan panduan mengenai pemilihan amalan pembangunan selamat yang sesuai menjadi cabaran kepada pengurus projek. Oleh itu, kajian ini membangunkan model bagi tujuan untuk membimbing pengurus projek memilih amalan pembangunan perisian yang selamat berdasarkan faktor-faktor yang dipenuhi oleh projek. Pada mulanya, kajian literatur sistematik (SLR) dijalankan dan hasilnya 18 faktor berpengaruh dikenal pasti. Bagi mengukuhkan dan meningkatkan dapatan ini, temu bual separa berstruktur dilakukan dengan 21 pakar pembangunan perisian dari lapan jabatan teknologi maklumat di sektor awam Malaysia dan 18 faktor yang mempengaruhi pelaksanaan amalan pembangunan perisian yang selamat telah dikenal pasti. Penemuan dari SLR dan temu bual digabungkan dan dianalisis menggunakan teknik *grounded theory*. Susulan ini, 20 faktor telah dimuktamadkan dan dikelompokkan menjadi empat kategori utama yang mempengaruhi hasil pembangunan perisian: konteks institusi, kandungan projek perisian, pengguna dan tindakan, dan proses pembangunan sistem. Untuk menilai pencapaian setiap faktor, kriteria penilaian telah dikenal pasti menggunakan kaedah analisis data sekunder. Selanjutnya, amalan pembangunan selamat yang sesuai untuk sektor awam Malaysia dikenal pasti menerusi kaedah tinjauan dan hasilnya, 24 amalan dikenal pasti sesuai. Faktor, kriteria penilaian dan amalan yang dikenal pasti disahkan menggunakan kaedah Delphi, yang melibatkan sepuluh orang pakar. Selain itu, para pakar memetakan faktor-faktor yang mempengaruhi setiap amalan pembangunan perisian yang selamat. Hasil daripada kaedah Delphi yang melibatkan tiga fasa, senarai faktor yang disahkan dan kriteria penilaian dihasilkan. Selain itu, senarai amalan yang dipetakan dengan faktor-faktor berpengaruh yang berkaitan telah dihasilkan. Unsur-unsur yang disahkan digunakan untuk membangunkan *Secure Software Development Practice Selection Model*. Model yang dicadangkan akhirnya dinilai menggunakan kaedah kajian kes yang melibatkan empat projek pembangunan perisian di sektor awam Malaysia. Pengurus projek diberikan soal selidik untuk menilai pencapaian faktor dan mengenal pasti amalan yang boleh dipraktikkan dalam projek pembangunan perisian mereka. Oleh itu, dengan *Secure Software Development Practice Selection Model* yang dicadangkan, amalan pembangunan perisian selamat yang sesuai dapat dikenal pasti dengan berkesan dengan menilai faktor-faktor berpengaruh yang dicapai oleh sesuatu projek perisian. Tambahan pula, skor purata yang diperoleh melalui *System Usability Scale* untuk semua agensi adalah 70.7; Oleh itu, *Secure Software Development Practice Selection Model* dianggap mempunyai tahap kegunaan yang baik. Ringkasnya, terdapat empat sumbangan penting dalam kajian ini; senarai faktor yang disahkan yang mempengaruhi pelaksanaan amalan pembangunan perisian selamat, senarai kriteria penilaian faktor, pemetaan amalan pembangunan perisian yang selamat kepada faktor yang berpengaruh, dan *Secure Software Development Practice Selection Model* yang telah dinilai.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACM	-	Association for Computing Machinery
CLASP	-	Comprehensive Lightweight Application Security Process
CV	-	Coefficient of Variation
MAMPU	-	Malaysian Administrative Modernization and Management Planning Unit
SDLC	-	Software Development Lifecycle
SLR		Systematic Literature Review
SPSS	-	Statistical Package for Social Science
SSD	-	Secure Software Development
SWEBOK	-	Software Engineering Body of Knowledge

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Chapter Overview

The aim of this study is to develop a model to select suitable secure software development practices for Malaysian Public Sector (MPS). This chapter presents the overview of this study. The first section of this chapter explains the background of the research problem, followed by the problem statement, research questions, objectives, and scope of the research. This explanation is continued by the significance of this research and provides a brief description on key terms applied throughout the thesis. The final section explains the outline of the thesis and overall chapter summary.

1.2 Problem Background

The advancement of internet and e-commerce have instilled revolutionary changes in peoples' lifestyle and living standards. Organizations are moving towards digitalizing services using a range of information and communication technologies. Both private and public organizations have transformed the way they run their daily operations and marketing activities from manual to the use of websites (Deepa & Thilagam, 2016; MAMPU, 2016). As more services go online, security becomes the biggest challenge in both public and private sector. Lack of security in the government services will affect the citizen's trust negatively because citizen's data can be compromised by irresponsible or unauthorized parties. Online applications has become a target of hackers due to strict vigilance on networks through firewalls and intrusion detection systems (Shuaibu, Norwawi, Selamat, & Al-Alwani, 2013). Many security incidents had been reported recently (MyCERT, 2019). Particularly, Cyber999 had recorded an increase of 44.56% in intrusion incidents reported in 2016 compared to 2015 (Kassim & Abdullah, 2017). Subsequently, 10699 cybersecurity

incidents were reported in 2018, representing 34% increase compared to year 2017. Such incidents reported to Cyber999 consist of account compromises (including email, social media and server accounts) and web defacements. Furthermore, most web defacements reported mainly exploited known vulnerabilities, for instance in the Content Management System or CMS that runs on web servers such as Joomla or Word Press.

Web applications are even more vulnerable compared to commercial applications due to the reason that web applications are available on internet (Brown & Paller, 2008). Present findings indicated that SQL injection and the exploitation of known vulnerabilities in a server are the trendy approaches used by attackers to compromise websites (MyCERT, 2019). Poorly constructed software systems and systems causes vulnerabilities in the system that can be exploited by malicious users and violate one or more software security properties (Shuaibu et al., 2013). Generally, security is the accountability of technical staffs who maintains antivirus, firewalls and intrusion detection systems. To prevent attackers, system administrators need to update security patches and apply best practices for web application. However, Cybersecurity Malaysia has stated that web defacements or web vandalism caused by vulnerable applications or unpatched servers are still rising (Cybersecurity, 2013). Furthermore, in 2016, National Institute of Standards and Technology (NIST) reported that most of the vulnerabilities are introduced during the design and architecture phase of software development and proper mitigations could have been taken to overcome the weaknesses (Black, Badger, Guttman, & Fong, 2016).

In 2016, National Institute of Standards and Technology (NIST) reported that most of the vulnerabilities are introduced during the design and architecture phase of software development and proper mitigations could have been taken to overcome the weaknesses (Black et al., 2016). Researches had indicated that the number and severity of vulnerabilities in online applications can be reduced by including security into development phases (Kainerstorfer, Sametinger, & Wiesauer, 2011). Scholars have used various methods and techniques such as security requirements engineering, security patterns and use cases to integrate security into software development life cycle (Lipner, 2004; Mellado, Fernández-Medina, & Piattini, 2007; Nunes, Belchior,

& Albuquerque, 2010). Microsoft Security Development Life cycle (SDL) , OWASP's Comprehensive, Lightweight Application Security Process (CLASP) and McGraw' Touchpoints are acknowledged as major players that provide an widespread set of activities covering a broad spectrum of the development lifecycle (De Win, Scandariato, Buyens, Grégoire, & Joosen, 2009). While these models cover the entire software development phase, efforts have been taken by some researchers to integrate security in a particular phase of software development such as requirement, design and implementation phase. It is believed that security must be tackled during the early phases of software development mainly during the requirement engineering (Mellado et al., 2007; P Salini & Kanmani, 2012). Various techniques such as threat modelling, use cases, misuse cases and abuser stories have been used to facilitate the management of security requirements engineering in software development life cycle (Mellado, Blanco, Sánchez, & Fernández-Medina, 2010). Meanwhile, UML and patterns are used in modelling secure designs (Abramov, Sturm, & Shoval, 2012; Eduardo B Fernandez, 2004).

Although various models have been introduced in efforts to produce secure software, many software development companies are still reluctant to use security development models. Project manager criticized that existing secure development processes for being too costly and complex (Geer, 2010). For example, a survey conducted by Oram (2017) pointed out acceptance and implementation of security practices in a software development process is insufficiently in place, and a majority of respondents highlighted that they want to perform the practice but cannot do it at all. Another study conducted in Finland highlights that only a small set of security activities are actively implemented (Rindell, Ruohonen, & Hyrynsalmi, 2018). In Malaysia, the implementation of secure software development is still in the early planning (Mohamed, Baharom, Deraman, Yahya, & Mohd, 2016). The awareness and readiness of the software developer to include the security practices in the software development process are still low even though there are many online or web applications are developed and introduced to the public day by day. This has become evident with vulnerabilities issues found on some of the Malaysian Public Sector online or web applications (Jaafar, 2017; Mohamed et al., 2016; Shuaibu et al., 2013). These scenarios highlight that the software development projects lack proper implementation of secure software practices.

It is found that lack of proper implementation of secure software practices is due to lack of knowledge in selecting suitable security practices (P. J. Morrison, 2017) which led the project managers only consider security requirements implicitly and let the security requirements undocumented, without any proper notations during software development process (Mohamed et al. (2016). Additionally, the project managers tend to ignore references and security guidelines on handling security practices issues. Despite the existence of many secure software development models (Howard & Lipner, 2009; OWASP, 2016) and guidelines, project managers find it difficult to select suitable practices for their projects due to lack of knowledge and guidance (P. J. Morrison, 2017). Selecting suitable practices are influenced by several factors such as inadequate development time (Jing, Lipford, & Bill, 2011), lack of skills or expertise (Hellström & Moberg, 2019; Mohamed et al., 2016) and improper team size (Jakeri & Hassan, 2018). Besides this, implementation of secure development models and practices in the industry requires security engineers or security experts to be part of the development team which poses a great challenge to small development teams involved in rapid development (Riaz, Slankas, King, & Williams, 2014). Assessment of these factors is necessary in order to assist projects managers to select suitable secure software development practices for their projects. However, literature on factors that influences the selection of secure software development practices is still lacking.

Background of the research shows security is an important element that need to be included in the software development especially online or web applications. Despite various efforts to reduce security problems, barriers in practical implementation are still exist due to many reasons. Lack of knowledge in security factors and practices by the software developers also has led to security vulnerabilities in online or web applications during the development (Yahya et al., 2019). According to Fraser, Campara, Fanning, McGraw, and Sullivan (2014), human awareness on security factors and practices can be the most cost- effective way to manage security. Thus, there is need to explore more in detail the security practices and factors for the implementation of secure software development during the software development process. This details will be useful in guiding and assisting software project managers in selecting suitable secure software development practices for their projects.

1.3 Problem Statement

Vulnerabilities are introduced in the online applications because developers fail to include security during the phases of software development. Despite the comprehensive guidelines from existing secure software development models and frameworks, implementation of secure development practices during software development is still lacking. Besides this, implementation of secure software development practices is also influenced by several factors such as development time, skills or expertise, top management support, automated tool support, team size and others. However, project managers find it difficult to select suitable practices for their projects due to lack of knowledge and guidance in assessing factors influencing the selection of secure software development practices. Therefore, assessment of factors is necessary in order to guide projects managers to select suitable secure software development practices for their projects. Thus, there is a need to add to the knowledge on the secure software development by guiding the project team to select suitable secure development practices that can be applied in their projects through assessment of related factors. In order to address the problem, this research propose to develop a model by incorporating practices involving factors into secure software development to facilitate selection of suitable security practices.

1.4 Research Goal

The goal of this research is to propose Secure Software Development Practice Selection Model. The research solution will act as a foundation and guide for software project managers in an organization to analyze and select a set of secure development practices by assessing the factors fulfilled by the organization. Hence, to achieve this goal, a set of research questions have been designed, as listed below:

a) What are the factors and its assessment criteria that influence the selection of secure software development practices?

b) What are the secure software development practices that are suitable for Malaysian Public Sector?

c) How are the factors, assessment criteria and practices validated and mapped?

d) How a suitable Secure Software Development Practice Selection Model can be proposed using the above findings?

1.5 Research Objectives

The objectives of this study are derived as below:

a) To identify factors and its assessment criteria that influence selection of secure software development practices.

b) To identify secure software development practices for Malaysian Public Sector.

c) To validate influential factors, assessment criteria and mapping of influential factors with secure software development practices.

d) To propose Secure Software Development Practice Selection Model.

e) To evaluate the proposed Secure Software Development Practice Selection Model.

1.6 Scope of the Study

The scope of this study is encompassed of secure software development factors, assessment criteria and practices. The following section delivers a detailed explanation of these scopes.

(a) Secure Software Development Factors

Secure software development is systematic process to reduce security vulnerabilities in the software being developed. This research focuses on identifying factors that influence secure software development practices during software development lifecycle from the project perspective. The factors are derived using Systematic Literature Review (SLR) and a semi structured interview method. The respondents who are involved in the interview were selected from Malaysian Public Sector only.

(b) Comprehensive Lightweight Application Security Process Model

The software security practices that are used in this study are adopted from the Comprehensive Lightweight Application Security Process model (CLASP). CLASP provides a detail process and presented with five high level perspectives. It is designed in order to embed security features especially during the software development life cycle.

(c) Malaysian Public Sector

Since software security problem is also a common problem faced in Malaysian Public Sector, respondents and experts involved in this study were selected from Malaysian Public Sector. Furthermore, possible factors that influence the selection of secure software development practices vary among private and public sector. Thus, focus of this study is on software development process at public sector.

1.7 Contribution and Significance of the Study

This research adds to the significant knowledge in the software engineering domain, especially on the software security and secure software development domain. The contribution of this study is as follows:

a) The first contribution of this research was the identification of 20 influential factors that affects the implementation of secure software development practices and 71 criteria to assess the achievement of the factors. Each factor and its assessment criteria were described accordingly.

b) The second contribution of this research was identification of secure software development practices for the Malaysian Public Sector. The practices were identified based on practitioner's agreement level on the importance of the practices.

c) The third contribution of this research was mapping of each secure software development practice to the factor that influences the implementation of that particular practice. Identification of factors influencing each practices is significant in selecting suitable practices to be implemented in a software project.

d) The fourth contribution of this research was the development of the Secure Software Development Practice Selection Model.

e) The fifth contribution of this research was the evaluated proposed model using case study method.

Additionally this study contributes to the area of knowledge in Software Engineering Body of Knowledge (SWEBOK) under Chapter 13, Computing Foundation, Subsection 17, Secure Software Development and Maintenance and specifically under subsection 17.5, (Society, Bourque, & Fairley, 2014). Currently, the security practices in the software development are not fully implemented by organizations, especially in public sectors like Malaysia. This study suggests the use of factors on selecting security practices in software development phases by the project managers and software developers. Thus, government agencies of Malaysia can reduce vulnerabilities during software development and produce secured online or web applications.

1.8 Glossary

(a) Software Project

A software project can be defined as a temporary endeavor or undertaken tasks related to Information Technology to create a product or process such as software project development. This study defines software project as an ICT project with a focus on application development.

(b) Secure Software Development

Secure software development is defined as the set of activities performed to develop, maintain, and deliver a secure software solution.

(c) Assessment Criteria

Assessment criteria in this study refer to questions or statement used to identify the existence of the factor in the project.

(d) Software Security Practices

Software security practices are software development practices implemented by project managers and developers to prevent security vulnerabilities in the software produced.

(e) Secure Software Development Factors

Secure software development factors refer to a circumstance or that contributes that influences the implementation of the secure software development practices during software development lifecycle.

1.9 Thesis Outline

This thesis consists of nine chapters. This chapter (Chapter 1) has briefly outlined the background of this study and the research problem and objectives. Below are the detailed explanations of Chapter 2 to Chapter 9 of this thesis.

(a) Chapter 2: Literature Review

Chapter 2 provides a comprehensive review of related studies in existing body of literature. The chapter is organized according to definitions, state of the art on secure development models, factors and criteria that influences secure development. Besides this, justification on selections of the methodologies in this study is also discussed here.

(b) Chapter 3: Research Methodology

Chapter 3 discusses the phases of the research design and methodology in detail. Explanation of the research phases includes related activities and deliverables. This chapter also discusses the research instruments and the evaluation criteria which were adopted in this work.

(c) Chapter 4: Identification of Factors and Assessment Criteria that Influence Selection of Secure Software Development Practices

Chapter 4 illustrates the data collection process using Systematic Literature Review to identify the factors that influence secure software development from state of the art perspective. Subsequently, this chapter also delivers the results from the structured interview session conducted among the experience software developers in Malaysian Public Sector. It highlights their practice, opinions, and experiences in implementing secure development practices in their projects. As a result of the structured interview, a set of factors that influence secure software development from the practitioner's perspective is identified. The identified factors from SLR and interview were consolidated to determine factors that influence the selection of secure software development practices which is the first objective of this study.

(d) Chapter 5: Identification of Secure Software Development Practices for Malaysian Public Service Organization

This chapter describes the identification of secure development practices that were important for Malaysian Public Sector. It illustrates the data collection process and presents the results of the survey conducted which fulfils the third objective of this study.

(e) Chapter 6: Validation of Factors, Assessment Criteria and Mapped Practices with Factors

This chapter explains the validation process of the factors and assessment criteria using Delphi method. The validated factors were further mapped to the secure development practices using the same method.

(f) Chapter 7: Formulation of Secure Software Development Practice Selection Model

This chapter describes the conceptual model of the Secure Software Development Practice Selection Model.

(g) Chapter 8: Evaluation of Secure Software Development Practice Selection Model

This chapter reports the evaluation outcomes of the proposed model. The evaluation phase is divided into two stages: investigation of the effectiveness of the model in identifying secure software development practices and the usability of the model. The software project managers involved in these two stages of evaluation are based on selected software projects.

(h) Chapter 9: Discussion and Conclusion

This chapter reflects back on the dissertation as a whole, to examine whether or not the research questions and research objectives have been answered. Next, this chapter highlights the contribution of this study. Finally, the limitations and the future directions of this study are addressed.

1.10 Chapter Summary

To conclude, this chapter provides an explanation of the current issue in this secure software development implementation and the need for this research to be carried out as the background of this study. The problem statement addresses the motivation in choosing the research topic and the research gap were identified. Subsequently, the research questions and objectives for this study were developed and presented. The research scope was also identified and explained in this chapter. This chapter also described the significance of this study and how it contributes to the state of knowledge in the software security especially in the domain of secure software development.

REFERENCES

Abdullah, S. F., Yusof, M. M., & Jambari, D. I. (2016). Model pengurusan risiko perancangan sistem maklumat di sektor awam. *Jurnal Pengurusan (UKM Journal of Management), 48*.

Abramov, J., Anson, O., Dahan, M., Shoval, P., & Sturm, A. (2012). A methodology for integrating access control policies within database development. *Computers & Security, 31*(3), 299-314. doi:10.1016/j.cose.2012.01.004.

Abramov, J., Sturm, A., & Shoval, P. (2012). Evaluation of the Pattern-based method for Secure Development (PbSD): A controlled experiment. *Information and Software Technology, 54*(9), 1029-1043. doi:10.1016/j.infsof.2012.04.001.

Adebiyi, A., Arreymbi, J., & Imafidon, C. (2012). *Applicability of neural networks to software security.* Paper presented at the UKSim 14th International Conference on Computer Modelling and Simulation (UKSim).

Adebiyi, A., Arreymbi, J., & Imafidon, C. (2013). Security Assessment of Software Design using Neural Network. *arXiv preprint arXiv:1303.2017*.

Adler, M., & Ziglio, E. (1996). *Gazing into the oracle: The Delphi method and its application to social policy and public health*: Jessica Kingsley Publishers.

Ahmad, Z., Asif, M., Shahid, M., & Rauf, A. (2015). Implementation of Secure Software Design and their Impact on Application. *International Journal of Computer Applications, 120*(10).

Al-Ahmad, W., & Al-Kaabi, R. (2008). *An extended security framework for e-government.* Paper presented at the IEEE International Conference on Intelligence and Security Informatics, 2008 (ISI 2008).

Alam, S. M. S., Singh, S., & Khan, S. A. (2016). A Strategy Oriented Process Model for Software Security. *International Journal of Engineering and Management Research (IJEMR), 6*(6), 137-142.

Alebrahim, A., & Heisel, M. (2014). Towards Developing Secure Software Using Problem-Oriented Security Patterns. In *Availability, Reliability, and Security in Information Systems* (pp. 45-62): Springer.

Ali, I., Asif, M., Shahbaz, M., Khalid, A., Rehman, M., & Guergachi, A. (2018). Text categorization approach for secure design pattern selection using software requirement specification. *IEEE Access, 6*, 73928-73939.

Alkussayer, A., & Allen, W. (2009). The ISDF Framework: Integrating Security Patterns and Best Practices. In J. Park, J. Zhan, C. Lee, G. Wang, T.-h. Kim, & S.-S. Yeo (Eds.), *Advances in Information Security and Its Application* (Vol. 36, pp. 17-28): Springer Berlin Heidelberg.

Alnatheer, M., Chan, T., & Nelson, K. (2012). *Understanding And Measuring Information Security Culture.* Paper presented at the PACIS.

Alqudah, M. K., Razali, R., & Alqudah, M. K. (2019). Agile Methods Selection Model: A Grounded Theory Study. *International Journal of Advanced Computer Science and Applications, 10*(7), 357-366.

Anwar, F., & Razali, R. (2016). Stakeholders selection model for software requirements elicitation. *American Journal of Applied Sciences, 13*(6), 726-738.

Apvrille, A., & Pourzandi, M. (2005a). Secure software development by example. *IEEE security and privacy, 3*(4), 10-17.

Apvrille, A., & Pourzandi, M. (2005b). Secure software development by example. *IEEE security & privacy, 3*(4), 10-17.

Assal, H., & Chiasson, S. (2018). *Security in the Software Development Lifecycle.* Paper presented at the Fourteenth Symposium on Usable Privacy and Security ({SOUPS} 2018).

Babchuk, W. A. (1996). *Glaser or Strauss? Grounded theory and adult education.* Paper presented at the Proceedings of the 15th Annual Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education.

Baca, D., Petersen, K., Carlsson, B., & Lundberg, L. (2009, 16-19 March 2009). *Static Code Analysis to Detect Software Security Vulnerabilities - Does Experience Matter?* Paper presented at the International Conference on Availability, Reliability and Security, 2009. ARES '09.

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction, 24*(6), 574-594.

Bartsch, S. (2011, 22-26 Aug. 2011). *Practitioners' Perspectives on Security in Agile Development.* Paper presented at the Sixth International Conference on Availability, Reliability and Security, 2011. ARES 2011.

Beretta, R. (1996). A critical review of the Delphi technique. *Nurse researcher, 3*(4), 79-89.

BKCASE. (2019). The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.0, . Retrieved from www.sebokwiki.org.

Black, P., Badger, M., Guttman, B., & Fong, E. (2016). *Dramatically Reducing Software Vulnerabilities: Report to the White House Office of Science and Technology Policy* ((No. NIST Internal or Interagency Report (NISTIR) 8151 (Draft)). National Institute of Standards and Technology.).

Bonver, E., & Cohen, M. (2008). Developing and Retaining a Security Testing Mindset. *Security & Privacy, IEEE, 6*(5), 82-85. doi:10.1109/MSP.2008.115

Braun, V., Clarke, V., & Terry, G. (2012). Thematic analysis. *APA handbook of research methods in psychology, 2*, 57-71.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry, 189*(194), 4-7.

Brown, M., & Paller, A. (2008). Secure software development: Why the development world awoke to the challenge. *Information Security Technical Report, 13*(1), 40-43. doi:http://dx.doi.org/10.1016/j.istr.2008.03.001

Bukhari, Z., Yahaya, J., & Deraman, A. (2018). A Conceptual Framework for Metrics Selection: SMeS. *International Journal on Advanced Science, Engineering and Information Technology, 8*(6), 2294-2300.

Byers, D., & Shahmehri, N. (2007, 10-13 April 2007). *Design of a Process for Software Security.* Paper presented at the The Second International Conference on Availability, Reliability and Security, 2007. ARES 2007.

Chakravarti, A., Vasanta, B., Krishnan, A., & Dubash, R. (1998). Modified Delphi methodology for technology forecasting case study of electronics and information technology in India. *Technological Forecasting and Social Change, 58*(1-2), 155-165.

Chess, B., & Arkin, B. (2011). Software security in practice. *Security & Privacy, IEEE, 9*(2), 89-92.

Church, R. M. (2002). The effective use of secondary data. *Learning and motivation, 33*(1), 32-45.

Colesky, M., Futcher, L., & Van Niekerk, J. (2013). *Design patterns for secure software development: demonstrating security through the MVC pattern.* Paper presented at the 15th Annual Conference on WWW Applications, Cape Town.

Colley, J. (2010). Why Secure Coding is not Enough: Professionals' Perspective. In *ISSE 2009 Securing Electronic Business Processes* (pp. 302-311): Springer.

Corbin, J., & Strauss, A. (2008). Basics of qualitative research: techniques and procedures for developing grounded theory. 2008. In: Sage Publications, Inc.

Corbin, J. M., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology, 13*(1), 3-21.

Creswell, J. W. (2012). *Educational research: Planning, conducting and evaluating quantitative and qualitative research* (4th Edition ed.): Pearson, 2012.

Crisp, J., Pelletier, D., Duffield, C., Adams, A., & Nagy, S. (1997). The Delphi method? *Nursing Research, 46*(2), 116-118.

Cruzes, D. S., & Dyba, T. (2011). *Recommended steps for thematic synthesis in software engineering.* Paper presented at the International Symposium on Empirical Software Engineering and Measurement (ESEM), 2011

Cybersecurity, M. (2013, 6 August 2013). e-Security Bulletin. *e-Security Bulleti, 34(Quarter 1/2013)*. Retrieved from https://www.cybersecurity.my/en/knowledge_banks/esecurity_bulletin/main/detail/2338/index.html

Czinkota, M. R., & Ronkainen, I. A. (1997). International business and trade in the next decade: Report from a Delphi study. *Journal of International Business Studies, 28*(4), 827-844.

Dajani, J. S., Sincoff, M. Z., & Talley, W. K. (1979). Stability and agreement criteria for the termination of Delphi studies. *Technological Forecasting and Social Change, 13*(1), 83-90.

Daud, M. I. (2010). *Secure software development model: A guide for secure software life cycle.* Paper presented at the Proceedings of the international MultiConference of Engineers and Computer Scientists.

Davis, N. (2013). Secure software development life cycle processes. *Software Engineering Institute CMU*.

Day, J., & Bobeva, M. (2005). A generic toolkit for the successful management of Delphi studies. *The Electronic Journal of Business Research Methodology, 3*(2), 103-116.

De Sousa, J. M. E. (2004). *Definition and analysis of critical success factors for ERP implementation projects*: Universitat Politècnica de Catalunya.

De Win, B., Scandariato, R., Buyens, K., Grégoire, J., & Joosen, W. (2009). On the secure software development process: CLASP, SDL and Touchpoints compared. *Information and Software Technology, 51*(7), 1152-1171. doi:http://dx.doi.org/10.1016/j.infsof.2008.01.010

Deepa, G., & Thilagam, P. S. (2016). Securing web applications from injection and logic vulnerabilities: Approaches and challenges. *Information and Software Technology, 74*, 160-180.

Delbecq, A. L., Van de Ven, A. H., & Gustafson, D. H. (1975). *Group techniques for program planning: A guide to nominal group and Delphi processes*: Scott, Foresman Glenview, IL.

Delone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: a ten-year update. *Journal of management information systems, 19*(4), 9-30.

Deschene, M. (2016). *Embracing security in all phases of the software development life cycle: A Delphi study.* Capella University,

Diamant, J. (2011). Resilient security architecture: A complementary approach to reducing vulnerabilities. *IEEE security & privacy, 9*(4), 80-84.

Dianxiang, X., Manghui, T., Sanford, M., Thomas, L., Woodraska, D., & Weifeng, X. (2012). Automated Security Test Generation with Formal Threat Models. *Dependable and Secure Computing, IEEE Transactions on, 9*(4), 526-540. doi:10.1109/TDSC.2012.24

Díaz, G., & Bermejo, J. R. (2013). Static analysis of source code security: Assessment of tools against SAMATE tests. *Information and Software Technology, 55*(8), 1462-1476. doi:http://dx.doi.org/10.1016/j.infsof.2013.02.005

EdgeScan. (2018). 2018 Vulnerability Statistics Report. Retrieved from https://www.edgescan.com/wp-content/uploads/2018/05/edgescan-stats-report-2018.pdf

Elahi, G., Yu, E., Li, T., & Liu, L. (2011). *Security requirements engineering in the wild: A survey of common practices.* Paper presented at the 2011 IEEE 35th Annual Computer Software and Applications Conference.

English, J. M., & Kernan, G. L. (1976). The prediction of air travel and aircraft technology to the year 2000 using the Delphi method. *Transportation research, 10*(1), 1-8.

Essafi, M., Labed, L., & Ben Ghezala, H. (2007). *S2D-Prom: A strategy oriented process model for secure software development.* Paper presented at the International Conference on Software Engineering Advances, 2007. ICSEA 2007.

Fernandez, E. B. (2004). *A Methodology for Secure Software Design.* Paper presented at the Software Engineering Research and Practice.

Fernandez, E. B., & Larrondo-Petrie, M. M. (2010, 5-8 Jan. 2010). *Designing Secure SCADA Systems Using Security Patterns.* Paper presented at the 43rd Hawaii International Conference on System Sciences (HICSS), 2010

Flechais, I., Mascolo, C., & Sasse, M. A. (2007). Integrating security and usability into the requirements and design process. *International Journal of Electronic Security and Digital Forensics, 1*(1), 12-26.

Fraser, S. D., Campara, D., Fanning, M. C., McGraw, G., & Sullivan, K. (2014). *Privacy and security in a networked world*. Paper presented at the Proceedings of the companion publication of the 2014 ACM SIGPLAN conference on Systems, Programming, and Applications: Software for Humanity, Portland, Oregon, USA.

Fuchs, A., & Rudolph, C. (2012, 14-16 Dec. 2012). *Security Engineering Based on Structured Formal Reasoning.* Paper presented at the ASE/IEEE International Conference on BioMedical Computing (BioMedCom), 2012.

Futcher, L., & Solms, R. v. (2008). *Guidelines for secure software development*. Paper presented at the Proceedings of the 2008 annual research conference of the South

African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology, Wilderness, South Africa.

Gamlo, A., & Bamasak, O. (2009, 9-12 Nov. 2009). *Towards securing e-transactions in e-government systems of Saudi Arabia.* Paper presented at the International Conference for Internet Technology and Secured Transactions, 2009. ICITST 2009.

Geer, D. (2010). Are companies actually using secure development life cycles? *Computer, 43*(6), 12-16.

Gibson, J. M. (1998). Using the Delphi technique to identify the content and context of nurses' continuing professional development needs. *Journal of clinical nursing, 7*(5), 451-459.

Gilliam, D. P. (2004). *Security risks: management and mitigation in the software life cycle.* Paper presented at the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004.

Glaser, B. G., Strauss, A. L., & Strutzel, E. (1968). The discovery of grounded theory; strategies for qualitative research. *Nursing Research, 17*(4), 364.

Glenn Wurster, P. C. v. O. (2008). The Developer is the Enemy. *NSPW'08*.

Glisson, W. B., & Welland, R. (2005, 31 Oct.-2 Nov. 2005). *Web development evolution: the assimilation of Web engineering security.* Paper presented at the Web Congress, 2005. LA-WEB 2005. Third Latin American.

Goertzel, K. M., & Winograd, T. (2008). Enhancing the development life cycle to produce secure software. *Technology Analysis Center (IATAC), USA, October*.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The qualitative report, 8*(4), 597-606.

Guan, H., Chen, W., Liu, L., & Yang, H. (2011). Environment-Driven Threats Elicitation for Web Applications. In J. O'Shea, N. Nguyen, K. Crockett, R. Howlett, & L. Jain (Eds.), *Agent and Multi-Agent Systems: Technologies and Applications* (Vol. 6682, pp. 291-300): Springer Berlin Heidelberg.

Hadavi, M. A., Sangchi, H. M., Hamishagi, V. S., & Shirazi, H. (2008, 4-7 March 2008). *Software Security; A Vulnerability Activity Revisit.* Paper presented at the Third

International Conference on Availability, Reliability and Security, 2008. ARES 08.

Haidar, G. G., & Bakar, A. Z. A. (2012). E-Government Success In Malaysia Through Government Portal And Website Assessment. *International Journal of Computer Science Issues (IJCSI), 9*(5).

Hanafizadeh, P., & Ravasan, A. Z. (2011). A McKinsey 7S model-based framework for ERP readiness assessment. *International Journal of Enterprise Information Systems (IJEIS), 7*(4), 23-63.

Hasson, F., Keeney, S., & McKenna, H. (2000). Research guidelines for the Delphi survey technique. *Journal of advanced nursing, 32*(4), 1008-1015.

Heath, H., & Cowley, S. (2004). Developing a grounded theory approach: a comparison of Glaser and Strauss. *International journal of nursing studies, 41*(2), 141-150.

Hein, D., & Saiedian, H. (2009). Secure Software Engineering: Learning from the Past to Address Future Challenges. *Information Security Journal: A Global Perspective, 18*(1), 8-25. doi:10.1080/19393550802623206

Hellström, J., & Moberg, A. (2019). A Lightweight Secure Development Process for Developers.

Hertzog, M. A. (2008). Considerations in determining sample size for pilot studies. *Research in nursing & health, 31*(2), 180-191.

Howard, M., & Lipner, S. (2009). *The security development lifecycle* (Vol. 11): Microsoft Press.

Hsu, C.-C., & Sandford, B. A. (2007). The Delphi technique: making sense of consensus. *Practical assessment, research & evaluation, 12*(10), 1-8.

Hussain, S., Erwin, H., & Dunne, P. (2011). *Threat modeling using formal methods: A new approach to develop secure web applications.* Paper presented at the 7th International Conference on Emerging Technologies (ICET), 2011.

Hussain, S., Kamal, A., Ahmad, S., Rasool, G., & Iqbal, S. (2014). Threat modelling methodologies: a survey. *Sci. Int.(Lahore), 26*(4), 1607-1609.

Infosec. (2013). Introduction to Secure Software Development Life Cycle. Retrieved from https://resources.infosecinstitute.com/intro-secure-software-development-life-cycle/#

Institute, P. M. (2004). *A Guide To The Project Management Body Of Knowledge (PMBOK Guides)*. Newtown Square, Pa: Project Management Institute.

Islam, S., & Dong, W. (2008). *Human factors in software security risk management.* Paper presented at the Proceedings of the first international workshop on Leadership and management in software architecture.

ISO. (2013). Information technology -- Security techniques -- Information security management systems -- Requirements. In. https://www.iso.org/.

Jaafar, S. b. A. R. N. b. (2017). Ops Bendera Analysis. *e-Security, Vol: 43 - (2/2017)*.

Jadhav, A. S., & Sonar, R. M. (2011). Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software, 84*(8), 1394-1407.

Jain, S., & Ingle, M. (2012, 5-7 Sept. 2012). *Techno-management view of Secured Software Development.* Paper presented at the 2012 CSI Sixth International Conference on Software Engineering (CONSEG).

Jakeri, M. M., & Hassan, M. F. (2018). *A Review of Factors Influencing the Implementation of Secure Framework for in-House Web Application Development in Malaysian Public Sector.* Paper presented at the 2018 IEEE Conference on Application, Information and Network Security (AINS).

Jing, X., Lipford, H. R., & Bill, C. (2011, 18-22 Sept. 2011). *Why do programmers make security errors?* Paper presented at the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).

Jing Xie, H. R. L., Bill Chu. (2012). Evaluating Interactive Support for Secure Programming. *CHI'12*.

Jinhua, L., & Jing, L. (2010). *Model Checking Security Vulnerabilities in Software Design.* Paper presented at the 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM).

Johnson, R. B. (1997). Examining the validity structure of qualitative research. *Education, 118*(2), 282.

Jones, R. L., & Rastogi, A. (2004). Secure Coding: Building Security into the Software Development Life Cycle. *Information Systems Security, 13*(5), 29-39. doi:10.1201/1086/44797.13.5.20041101/84907.5.

Kainerstorfer, M., Sametinger, J., & Wiesauer, A. (2011). *Software security for small development teams: a case study.* Paper presented at the Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services.

Kakkar, M., & Jain, S. (2016). *Feature selection in software defect prediction: A comparative study.* Paper presented at the 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence).

Kalaian, S. A., & Kasim, R. M. (2012). Terminating sequential Delphi survey data collection. *Practical assessment, research & evaluation, 17*(5).

Karpati, P., Sindre, G., & Opdahl, A. L. (2011). *Characterising and analysing security requirements modelling initiatives.* Paper presented at the 2011 6th International Conference on Availability, Reliability and Security, ARES 2011, Vienna.

Kassim, S. R. M., & Abdullah, K. (2017). e-Security Bulletin *42 (1/2017)*, 17-19. Retrieved from http://www.cybersecurity.my/en/knowledge_banks/esecurity_bulletin/main/detail/2338/index.html

Kasunic, M. (2005). *Designing an effective survey* ((No. CMU/SEI-2005-HB-004). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.

Kaur, D., & Kaur, P. (2016). Empirical analysis of web attacks. *Procedia Computer Science, 78*, 298-306.

Keeney, S., Hasson, F., & McKenna, H. P. (2001). A critical review of the Delphi technique as a research methodology for nursing. *International journal of nursing studies, 38*(2), 195-200.

Khan, K. S., Ter Riet, G., Glanville, J., Sowden, A. J., & Kleijnen, J. (2001). *Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews*: NHS Centre for Reviews and Dissemination.

Khan, M. U. A., & Zulkernine, M. (2009). On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software. 353-358. doi:10.1109/compsac.2009.206

Khidzir, N. Z., Mohamed, A., & Arshad, N. H. (2013). ICT outsourcing information security risk factors: an exploratory analysis of threat risks factor for critical project characteristics. *Journal of Industrial and Intelligent Information Vol, 1*(4).

Keele, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Vol. 5). Technical report, Ver. 2.3 EBSE Technical Report. EBSE.

Kitchenham, B., Linkman, S., & Law, D. (1997). DESMET: A methodology for evaluating software engineering methods and tools. *Computing & Control Engineering Journal, 8*(3), 120-126.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering–A systematic literature review. *Information and Software Technology, 51*(1), 7-15.

Kleidermacher, D., & Wolf, M. (2008, 26-30 Oct. 2008). *Using static analysis to improve communications infrastructure.* Paper presented at the IEEE/AIAA 27th Digital Avionics Systems Conference, 2008. DASC 2008.

Kleidermacher, D. N. (2008, 12-13 May 2008). *Integrating Static Analysis into a Secure Software Development Process.* Paper presented at the 2008 IEEE Conference onTechnologies for Homeland Security.

Knauss, E., Houmb, S., Schneider, K., Islam, S., & Jürjens, J. (2011). Supporting Requirements Engineers in Recognising Security Issues. In D. Berry & X. Franch (Eds.), *Requirements Engineering: Foundation for Software Quality* (Vol. 6606, pp. 4-18): Springer Berlin Heidelberg.

Kortum, P. T., & Bangor, A. (2013). Usability ratings for everyday products measured with the System Usability Scale. *International Journal of Human-Computer Interaction, 29*(2), 67-76.

Leech, N. L., & Onwuegbuzie, A. J. (2007). An array of qualitative data analysis tools: A call for data analysis triangulation. *School psychology quarterly, 22*(4), 557.

Linstone, H. A., & Turoff, M. (1975). *The delphi method*: Addison-Wesley Reading, MA.

Lipner, S. (2004). *The trustworthy computing security development lifecycle.* Paper presented at the Computer Security Applications Conference, 2004. 20th Annual.

Lipner, S. (2010). Security development lifecycle. *Datenschutz und Datensicherheit - DuD, 34*(3), 135-137. doi:10.1007/s11623-010-0021-7

Lummus*, R. R., Vokurka, R. J., & Duclos, L. K. (2005). Delphi study on supply chain flexibility. *International journal of production research, 43*(13), 2687-2708.

Lynn Futcher, R. v. S. (2007). SecSDM: A Model for Integrating Security into the

Software Development Life Cycle. *IFIP International Federation for Information Processing*.

Ma, Z., Wagner, C., Bonitz, A., Bleier, T., Woitsch, R., & Nichterl, M. (2012). Model-driven secure development lifecycle. *International Journal of Security and Its Applications, 6*(2), 443-448. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84864766736&partnerID=40&md5=03be0a32c073ef8fe6f4337f3c9ff475

Majeed, M., & Quadri, S. (2017). Secure Software Development Process: A Survey. *International Journal of Innovations & Advancement in Computer Science (IJIACS), 6*(11).

MAMPU. (2016). Pelan Strategik ICT Sektor Awam 2016-2020. Retrieved from https://www.mampu.gov.my/images/agensikerajaan/perkhidmatan/The-Malaysian-Public-Sector-ICT-Strategic-Plan-2016_2020.pdf

Masrom, M., Lim, E. A., & Din, S. (2013). Security and Quality Issues in Trusting E-Government Service Delivery. *Managing Trust in Cyberspace*, 197.

Mathison, S. (1988). Why triangulate? *Educational researcher, 17*(2), 13-17.

McGraw, G. (2006). *Software security: building security in* (Vol. 1): Addison-Wesley Professional.

McKenna, H. P. (1994). The Delphi technique: a worthwhile research approach for nursing? *Journal of advanced nursing, 19*(6), 1221-1225.

McLeod, L., & MacDonell, S. G. (2011). Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR), 43*(4), 24.

Mead, N. R., Allen, J. H., Barnum, S. J., Ellison, R. J., & McGraw, G. (2004). *Software Security Engineering: A Guide for Project Managers*: Addison-Wesley Professional.

Mead, N. R., & McGraw, G. (2005). A Portal for Software Security. *Security & Privacy, IEEE, 3*(4), 75-79. doi:10.1109/MSP.2005.88

Mead, N. R., & Stehney, T. (2005). *Security quality requirements engineering (SQUARE) methodology* (Vol. 30): ACM.

Meland, P. H., & Jensen, J. (2008). *Secure software design in practice.* Paper presented at the Third International Conference on Availability, Reliability and Security, 2008. ARES 08. .

Mellado, D., Blanco, C., Sánchez, L. E., & Fernández-Medina, E. (2010). A systematic review of security requirements engineering. *Computer standards & interfaces, 32*(4), 153-165.

Mellado, D., Fernández-Medina, E., & Piattini, M. (2007). A common criteria based security requirements engineering process for the development of secure information systems. *Computer standards & interfaces, 29*(2), 244-253.

Michael Kainerstorfer, J. S., Andreas Wiesauer. (2011). Software Security for Small Development Teams – A Case Study. *WAS2011*.

Microsoft. (2010, 4 November 2010). Simplified Implementation of the Microsoft SDL. Retrieved from https://www.microsoft.com/en-us/securityengineering/sdl/

Mockel, C., & Abdallah, A. E. (2010, 23-25 Aug. 2010). *Threat modeling approaches and tools for securing architectural designs of an e-banking application.* Paper presented at the 2010 Sixth International Conference on Information Assurance and Security (IAS)

Mohamed, S. F. P., Baharom, F., Deraman, A., Yahya, J., & Mohd, H. (2016). An Exploratory Study on Secure Software Practices Among Software Practitioners in Malaysia. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8*(8), 39-45.

Mohammad, A., & Abushariah, M. (2017). *Secure software engineering: Evaluation of emerging trends.* Paper presented at the 2017 8th International Conference on Information Technology (ICIT).

Morrison, P. (2015). *A security practices evaluation framework.* Paper presented at the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.

Morrison, P. J. (2017). *A Security Practices Evaluation Framework*. Unpublished Doctoral dissertation. North Carolina State University.

Mougoue, E. (2016). What is the secure SDLC and why should I care? Retrieved from https://www.synopsys.com/blogs/software-security/secure-sdlc/

MyCERT. (2019). *Malaysia Threat Landscape 2018 - Based on Incidents Reported To*

CyberSecurity Malaysia. Retrieved from https://www.mycert.org.my/portal/-publicationdoc?id=270d8ee0-cdd1-49fb-827d-f8fca7752155.

Myers, M. D. (2013). *Qualitative research in business and management*: Sage.

Nazir, S., Anwar, S., Khan, S. A., Shahzad, S., Ali, M., Amin, R., . . . Cosmas, J. (2014). Software component selection based on quality criteria using the analytic network process. *Abstract and Applied Analysis, 2014*.

Nazir, S., Khan, M. A., Anwar, S., Khan, H., & Nazir, M. (2012). *A novel fuzzy logic based software component selection modeling.* Paper presented at the 2012 International Conference on Information Science and Applications.

Nguyen, J., & Dupuis, M. (2019). *Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations.* Paper presented at the Proceedings of the 20th Annual SIG Conference on Information Technology Education.

Nunes, F. J. B., Belchior, A. D., & Albuquerque, A. B. (2010). *Security engineering approach to support software security.* Paper presented at the 2010 6th World Congress on Services (SERVICES-1).

Okoli, C., & Pawlowski, S. D. (2004). The Delphi method as a research tool: an example, design considerations and applications. *Information & management, 42*(1), 15-29.

Okubo, T., Kaiya, H., & Yoshioka, N. (2012, 16-20 July 2012). *Mutual Refinement of Security Requirements and Architecture Using Twin Peaks Model.* Paper presented at the IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), 2012

Okubo, T., & Tanaka, H. (2008a). *Web security patterns for analysis and design*. Paper presented at the Proceedings of the 15th Conference on Pattern Languages of Programs, Nashville, Tennessee, USA.

Okubo, T., & Tanaka, H. (2008b). *Web security patterns for analysis and design.* Paper presented at the Proceedings of the 15th Conference on Pattern Languages of Programs.

Onut, S., & Efendigil, T. (2010). A theorical model design for ERP software selection process under the constraints of cost and quality: A fuzzy approach. *Journal of Intelligent & Fuzzy Systems, 21*(6), 365-378.

Onwuegbuzie, A. J., Leech, N. L., & Collins, K. M. (2012). Qualitative analysis techniques for the review of the literature. *The qualitative report, 17*(28), 1.

Oram, A. (2017). The alarming state of secure coding neglect : A survey reveals a deep divide between developer aspirations for security and organizational practices. Retrieved from https://www.oreilly.com/ideas/the-alarming-state-of-secure-coding-neglect

OWASP. (2016, 10 August 2016). Comprehensive Lightweight Application Secrity Process. Version 1.2. Retrieved from https://www.owasp.org/images/9/9f/Us_owasp-clasp-v12-for-print-lulu.pdf

OWASP. (2017). Top 10-2017 The Ten Most Critical Web Application Security Risks. *URL: owasp. org/images/7/72/OWASP_Top_10-2017_% 28en, 29*.

Pavlidis, M., Mouratidis, H., Panaousis, E., & Argyropoulos, N. (2017). *Selecting security mechanisms in secure tropos.* Paper presented at the International Conference on Trust and Privacy in Digital Business.

Payne, J. (2010). Integrating Application Security into Software Development. *IT Professional, 12*(2), 6-9. doi:10.1109/MITP.2010.58

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic mapping studies in software engineering.* Paper presented at the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12.

Pfleeger, C. P., & Pfleeger, S. L. (2002). *Security in Computing*: Prentice Hall Professional Technical Reference.

Powell, C. (2003). The Delphi technique: myths and realities. *Journal of advanced nursing, 41*(4), 376-382.

Prescott, P. A., & Soeken, K. L. (1989). The potential uses of pilot work. *Nursing Research, 38*(1), 60.

Raghavan, V. V., & Zhang, X. (2009). *Building security in during information systems development.* Paper presented at the 15th Americas Conference on Information Systems 2009, AMCIS 2009, San Francisco, CA.

Riaz, M., Slankas, J., King, J., & Williams, L. (2014). *Using templates to elicit implied security requirements from functional requirements-a controlled experiment.*

Paper presented at the Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.

Rindell, K., Ruohonen, J., & Hyrynsalmi, S. (2018). *Surveying Secure Software Development Practices in Finland.* Paper presented at the Proceedings of the 13th International Conference on Availability, Reliability and Security.

Rowe, G., & Wright, G. (1999). The Delphi technique as a forecasting tool: issues and analysis. *International journal of forecasting, 15*(4), 353-375.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131-164.

Sajesh, V. (2018). Forecasting using Delphi method: an Overview. In: ICAR-Central Institute of Fisheries Technology.

Salini, P., & Kanmani, S. (2012). Survey and analysis on Security Requirements Engineering. *Computers & Electrical Engineering*.

Salini, P., & Kanmani, S. (2013) Model Oriented Security Requirements Engineering (MOSRE) framework for web applications. In*: Vol. 177 AISC. 2nd International Conference on Advances in Computing and Information Technology, ACITY 2012* (pp. 341-353). Chennai.

Sandhya Menon, S. N., and Qishin Tariq. (2018, 10 Jun 2018). Details of 4.9 million students may have been hacked. *The Star Online*. Retrieved from https://www.thestar.com.my/news/nation/2018/06/10/details-of-49-million-students-may-have-been-hacked/

Shirey, R. (2007). *Internet security glossary, version 2* (2070-1721). Retrieved from

Shuaibu, B. M., Norwawi, N. M., Selamat, M. H., & Al-Alwani, A. (2013). Systematic review of web application security development model. *Artificial Intelligence Review*, 1-18.

Siddiqui, S. T. (2017). Significance of security metrics in secure software development. *Significance, 12*(6).

Skulmoski, G. J., Hartman, F. T., & Krahn, J. (2007). The Delphi method for graduate research. *Journal of Information Technology Education: Research, 6*, 1-21.

Society, I. C., Bourque, P., & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*: IEEE Computer Society Press.

Sodiya, A. S., Onashoga, S. A., & Ajayi, O. B. (2006). Towards building secure software systems. *Issues in Informing Science and Information Technology, 3*.

Sonia, A. S. (2014). Selection of security activities for integration with Agile methods after combining their agility and effectiveness. *Int. J. Web Appl., 6*(2), 57-67.

Steward Jr, C., Wahsheh, L. A., Ahmad, A., Graham, J. M., Hinds, C. V., Williams, A. T., & DeLoatch, S. J. (2012). *Software security: The dangerous afterthought.* Paper presented at the 2012 Ninth International Conference on Information Technology-New Generations.

Story, V., Hurdley, L., Smith, G., & Saker, J. (2000). Methodological and practical implications of the Delphi technique in marketing decision-making: a re-assessment. *The Marketing Review, 1*(4), 487-504.

Strauss, A., & Corbin, J. (1994). Grounded theory methodology. *Handbook of qualitative research*, 273-285.

Teodoro, N., & Serrao, C. (2011, 27-29 June 2011). *Web application security: Improving critical web-based applications quality through in-depth security analysis.* Paper presented at the *International Conference on Information Society (i-Society),* 2011 (pp. 457-462). IEEE.

Terpstra, E., Daneva, M., & Wang, C. (2017). *Agile practitioners' understanding of security requirements: insights from a grounded theory analysis.* Paper presented at the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW).

Thangaratinam, V., & Selvambigai, S. (2011). *Health technology assessment in maternal and perinatal medicine: delphi survey of practice, systematic reviews of evidence and meta analyses.* University of Birmingham.

Tharenou, P., Donohue, R., & Cooper, B. (2007). *Management research methods*: Cambridge University Press Melbourne.

Thurmond, V. A. (2001). The point of triangulation. *Journal of nursing scholarship, 33*(3), 253-258.

Tøndel, I. A., Jaatun, M. G., Cruzes, D. S., & Moe, N. B. (2017). Risk centric activities in secure software development in public organisations. *International Journal of Secure Software Engineering (IJSSE), 8*(4), 1-30.

Uma, S., & Roger, B. (2003). *Research methods for business: A skill building approach.* John Wiley & Sons.

Upadhyaya, P., Shakya, S., & Pokharel, M. (2012, 23-25 Nov. 2012). *E-government security readiness assessment for developing countries: Case study: Nepal Govt. organizations.* Paper presented at Third Asian Himalayas International Conference on Internet (AH-ICI), 2012.

Viega, J., & McGraw, G. (2001). *Building secure software: how to avoid security problems the right way*: Pearson Education.

Viera, A. J., & Garrett, J. M. (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med, 37*(5), 360-363.

Williams, P. L., & Webb, C. (1994). The Delphi technique: a methodological discussion. *Journal of advanced nursing, 19*(1), 180-186.

Witschey, J., Xiao, S., & Murphy-Hill, E. (2014a). Technical and Personal Factors Influencing Developers' Adoption of Security Tools. 23-26. doi:10.1145/2663887.2663898.

Witschey, J., Xiao, S., & Murphy-Hill, E. (2014b). *Technical and Personal Factors Influencing Developers' Adoption of Security Tools*. Paper presented at the Proceedings of the 2014 ACM Workshop on Security Information Workers, Scottsdale, Arizona, USA.

Witschey, J., Zielinska, O., Welk, A., Murphy-Hill, E., Mayhorn, C., & Zimmermann, T. (2015). *Quantifying developers' adoption of security tools.* Paper presented at the Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering.

Xie, J., Chu, B., & Richter Lipford, H. (2011). Idea: Interactive Support for Secure Software Development. In Ú. Erlingsson, R. Wieringa, & N. Zannone (Eds.), *Engineering Secure Software and Systems* (Vol. 6542, pp. 248-255): Springer Berlin Heidelberg.

Xu, D. (2013). Software Security Testing of an Online Banking System. *SIGCSE'13*.

Yahya, S., Kamalrudin, M., Sidek, S., Jaimun, M., Yusof, J., Hua, A. K., & Gani, P. (2019). *A Review Paper: Security Requirement Patterns for a Secure Software Development.* Paper presented at the 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS).

Ying, T. P. (2018, January 24, 2018 @ 12:06am). Personal data of 220,000 organ donors leaked online. *New Straits Time* Retrieved from https://www.nst.com.my/news/nation/2018/01/328140/personal-data-220000-organ-donors-leaked-online.

Zhu, J., Chu, B., Lipford, H., & Thomas, T. (2015). *Mitigating Access Control Vulnerabilities through Interactive Static Analysis*. Paper presented at the Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria.

Zia, T. A., & Rizvi, A. (2011). Source code embedded (SCEM) security framework.

Zuccato, A., Daniels, N., & Jampathom, C. (2011, 22-26 Aug. 2011). *Service Security Requirement Profiles for Telecom: How Software Engineers May Tackle Security.* Paper presented at the 2011 Sixth International Conference on Availability, Reliability and Security (ARES).

LIST OF PUBLICATIONS

1. Kanniah, S.L. and Mahrin, M.N.R., 2016. A review on factors influencing implementation of secure software development practices. *International Journal of Computer and Systems Engineering*, *10*(8), pp.3032-3039.

2. Kanniah, S.L. and Mahrin, M.N.R., 2018. Secure Software Development Practice Selection Model: A Delphi Study. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(2-8), pp.71-75.

3. Kanniah, S.L. and Mahrin, M.N.R.B., 2017. Influential Factors Affecting Secure Software Development Implementation at Public Service Organization: An Exploratory Study. *Advanced Science Letters*, *23*(9), pp.9157-9162.