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).

Appendix A Selected Papers for Data Extraction

ID	Citation
[S1]	Marback, A., Do, H., He, K., Kondamarri, S. & Xu, D. 2013. A Threat Model Based Approach to Security Testing. Software: Practice And Experience, 43, 241-258.
[S2]	Jones, R. L. & Rastogi, A. 2004. Secure Coding: Building Security Into The Software Development Life Cycle. Information Systems Security, 13, 29-39.
[S3]	Hein, D. & Saiedian, H. 2009. Secure Software Engineering: Learning from The Past to Address Future Challenges. Information Security Journal: A Global Perspective, 18, 8-25.
[S4]	Allen, J. 2007. Why Is Security a Software Issue? EDPACS, 36, 1-13.
[S5]	Mouratidis, H., Giorgini, P. & Manson, G. 2003. Integrating Security and Systems Engineering: Towards The Modelling of Secure Information Systems. In: Eder, J. & Missikoff, M. (Eds.) Advanced Information Systems Engineering. Springer Berlin Heidelberg.
[S6]	Abramov, J., Anson, O., Dahan, M., Shoval, P. & Sturm, A. 2012. A Methodology for Integrating Access Control Policies Within Database Development. Computers & Security, 31, 299-314.
[S7]	Alkussayer, A. & Allen, W. 2009. The ISDF Framework: Integrating Security Patterns and Best Practices. In: Park, J., Zhan, J., Lee, C., Wang, G., Kim, T.-H. & Yeo, S.-S. (Eds.) Advances in Information Security and Its Application. Springer Berlin Heidelberg.
[S8]	Salini, P. & Kanmani, S. 2013. Model Oriented Security Requirements Engineering (MOSRE) Framework for Web Applications. 2nd International Conference On Advances in Computing and Information Technology, ACITY 2012. Chennai.
[S9]	Al-Amin, S., Ajmeri, N., Du, H., Berglund, E.Z. and Singh, M.P., 2018. Toward effective adoption of secure software development practices. Simulation Modelling Practice and Theory, 85, pp.33-46.
[S10]	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
[S11]	Xie, J., Lipford, H. R. & Chu, B. Evaluating Interactive Support for Secure Programming. 30th ACM Conference On Human Factors in Computing Systems, CHI 2012, 2012 Austin, Tx. 2707-2716.
[S12]	Guan, H., Chen, W., Liu, L. & Yang, H. 2011. Environment-Driven Threats Elicitation for Web Applications. In: O'shea, J., Nguyen, N., Crockett, K., Howlett, R. & Jain, L. (Eds.) Agent and Multi-Agent Systems: Technologies and Applications. Springer Berlin Heidelberg.
[S13]	Bartsch, S. Practitioners' Perspectives On Security In Agile Development. Availability, Reliability and Security (ARES), 2011 Sixth International Conference On, 22-26 Aug. 2011 2011. 479-484.
[S14]	Zuccato, A., Daniels, N. & Jampathom, C. Service Security Requirement Profiles for Telecom: How Software Engineers May Tackle Security. Availability, Reliability and Security (ARES), 2011 Sixth International Conference On, 22-26 Aug. 2011 2011. 521-526.
[S15]	Baca, D., Petersen, K., Carlsson, B. & Lundberg, L. Static Code Analysis to Detect Software Security Vulnerabilities - Does Experience Matter? Availability, Reliability and Security, 2009. ARES '09. International Conference On, 16-19 March 2009 2009. 804-810.
[S16]	Geer, D. 2010. Are Companies Actually Using Secure Development Life Cycles? Computer, 43, 12-16.
[S17]	Okubo, T., Kaiya, H. & Yoshioka, N. Mutual Refinement of Security Requirements and Architecture Using Twin Peaks Model. Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual, 16-20 July 2012 2012. 367-372.
[S18]	Kleidermacher, D. & Wolf, M. Using Static Analysis to Improve Communications Infrastructure. Digital Avionics Systems Conference, 2008. Dasc 2008. IEEE/AIAA 27th, 26-30 Oct. 2008 2008. 1.D.5-1-1.D.5-6.
[S19]	Xie, J., Chu, B. & Richter Lipford, H. 2011. IDEA: Interactive Support for Secure Software Development. In: Erlingsson, Ú., Wieringa, R. & Zannone, N. (Eds.) Engineering Secure Software and Systems. Springer Berlin Heidelberg.

ID	Citation
[S20]	Mockel, C. & Abdallah, A. E. Threat Modeling Approaches and Tools for Securing Architectural Designs of an E-Banking Application. Information Assurance and Security (IAS), 2010 Sixth International Conference On, 23-25 Aug. 2010 2010. 149-154.
[S21]	Díaz, G. & Bermejo, J. R. 2013. Static Analysis of Source Code Security: Assessment of Tools Against Samate Tests. Information And Software Technology, 55, 1462-1476.
[S22]	Mohammad, A. and Abushariah, M., 2017, May. Secure software engineering: Evaluation of emerging trends. In 2017 8th International Conference on Information Technology (ICIT) (pp. 814-818). IEEE.
[S23]	Chand, P. 2005. Building India as The Destination for Secure Software Development – Next Wave of Opportunities for The ICT Industry. In: Jajodia, S. & Mazumdar, C. (Eds.) Information Systems Security. Springer Berlin Heidelberg.
[S24]	Teodoro, N. & Serrão, C. Web Application Security: Improving Critical Web-Based Applications Quality Through In-Depth Security Analysis. International Conference On Information Society, I-Society 2011, 2011 London. 457-462.
[S25]	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, 443-448.
[S26]	Haron, G.R. and Siong, N.K., 2011, December. Extrapolating security requirements to an established software process: Version 1.0. In 2011 International Conference for Internet Technology and Secured Transactions (pp. 752-757). IEEE.
[S27]	Colley, J. 2010. Why Secure Coding Is Not Enough: Professionals' Perspective. In: Pohlmann, N., Reimer, H. & Schneider, W. (Eds.) ISSE 2009 Securing Electronic Business Processes. Vieweg+Teubner.
[S28]	Payne, J. 2010. Integrating Application Security into Software Development. It Professional, 12, 6-9.
[S29]	Riaz, M., Slankas, J., King, J. & Williams, L. 2014. Using Templates to Elicit Implied Security Requirements from Functional Requirements - A Controlled Experiment. Proceedings of The 8th ACM/IEEE International Symposium On Empirical Software Engineering and Measurement. Torino, Italy: ACM.
[S30]	Okubo, T. & Tanaka, H. 2008. Web Security Patterns for Analysis and Design. Proceedings of The 15th Conference On Pattern Languages of Programs. Nashville, Tennessee, USA: ACM.
[S31]	Zhu, J., Chu, B., Lipford, H. & Thomas, T. 2015. Mitigating Access Control Vulnerabilities Through Interactive Static Analysis. Proceedings of The 20th ACM Symposium On Access Control Models and Technologies. Vienna, Austria: ACM.
[S32]	Wurster, G. & Oorschot, P. C. V. 2008. The Developer Is the Enemy. Proceedings of The 2008 Workshop On New Security Paradigms. Lake Tahoe, California, Usa: ACM.
[S33]	Witschey, J., Xiao, S. & Murphy-Hill, E. 2014. Technical and Personal Factors Influencing Developers' Adoption of Security Tools. Proceedings of The 2014 ACM Workshop On Security Information Workers. Scottsdale, Arizona, USA: ACM.
[S34]	Schneider, K., Knauss, E., Houmb, S., Islam, S. & Jürjens, J. 2012. Enhancing Security Requirements Engineering by Organizational Learning. Requirements Engineering, 17, 35-56.
[S35]	Karpati, P., Sindre, G. & Opdahl, A. 2010. Visualizing Cyber Attacks with Misuse Case Maps. In: Wieringa, R. & Persson, A. (Eds.) Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg.
[S36]	Knauss, E., Houmb, S., Schneider, K., Islam, S. & Jürjens, J. 2011. Supporting Requirements Engineers in Recognising Security Issues. In: Berry, D. & Franch, X. (Eds.) Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg.
[S37]	Bonver, E. & Cohen, M. 2008. Developing and Retaining a Security Testing Mindset. Security & Privacy, IEEE, 6, 82-85.
[S38]	Chess, B. & Arkin, B. 2011. Software Security in Practice. Security & Privacy, IEEE, 9, 89-92.
[S39]	Davis, N., Humphrey, W., Redwine, J. S. T., Zibulski, G. & Mcgraw, G. 2004. Processes for Producing Secure Software. Security & Privacy, IEEE, 2, 18-25.
[S40]	Diamant, J. 2011. Resilient Security Architecture: A Complementary Approach to Reducing Vulnerabilities. Security & Privacy, IEEE, 9, 80-84.

ID	Citation
[S41]	Jain, S. & Ingle, M. Techno-Management View of Secured Software Development. Software Engineering (CONSEG), 2012 CSI Sixth International Conference On, 5-7 Sept. 2012 2012. 1-6.
[S42]	Byers, D. & Shahmehri, N. Design of A Process for Software Security. Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference On, 10-13 April 2007 2007. 301-309.
[S43]	Glisson, W. B. & Welland, R. Web Development Evolution: The Assimilation of Web Engineering Security. Web Congress, 2005. La-Web 2005. Third Latin American, 31 Oct.-2 Nov. 2005 2005. 5 Pp.
[S44]	Jing, X., Lipford, H. R. & Bill, C. Why Do Programmers Make Security Errors? Visual Languages and Human-Centric Computing (Vl/Hcc), 2011 IEEE Symposium On, 18-22 Sept. 2011 2011. 161-164

Appendix B Field Note Template and Sample

B.1 Field Note Template

This field note template is designed for the reflection purposes during the interview session. The templates consist of:

A. Invitation Letter

B. Consent Form

C. Contact Summary Form

D. Interview Questions

A. Invitation Letter

Date:

Dear **____,**

This letter is an invitation to consider participating in a study I am conducting as part of my PhD degree in the Advanced Informatics School at the Universiti Teknologi Malaysia (UTM) under the supervision of Dr. Mohd Naz'ri Mahrain. I would like to provide you with more information about this study and what your involvement would entail if you decide to take part.

In the recent year, more services are going online and similarly the Malaysian Public Sector had targeted zero face-to-face service delivery with 90% of all government services available online by 2015. Security becomes a challenge and increases the importance of safeguarding the web application from internal and external threats. To reduce vulnerabilities in the application, secure software development have been introduced by the security experts. ***Secure software development is defined as "the set of activities performed to develop, maintain, and deliver a secure software solution".***

This study will focus on identifying factors and challenges faced by Malaysian Public Sector in implementing secure software development processes and the current secure software development practice at each organization. Therefore, I would like to include your organization as one of several organizations to be involved in my study.

Participation in this study is voluntary. It will involve an interview of approximately ***30 minutes*** *in length to take place in a mutually agreed upon location. You may decline to answer any of the interview questions if you wish. Further, you may decide to withdraw from this study at any time without any negative consequences by advising the researcher. With your permission, the interview will be audio recorded to facilitate collection of information, and later transcribed for analysis. All information you provide is considered completely confidential. Your name will not appear in any thesis or report resulting from this study, however, with your permission anonymous quotations may be used. Data collected during this study will be kept confidential. There are no known or anticipated risks to you as a participant in this study.*

If you have any questions regarding this study, or would like additional information to assist you in reaching a decision about participation, please contact me at ***016-2029444*** *or by email at* ***kanniah.srilakshmi@gmail.com****.*

I hope that the results of my study will be of benefit to those organizations directly involved in the study, as well as to the broader research community.

I very much look forward to speaking with you and thank you in advance for your assistance in this study.

Yours sincerely,

SRI LAKSHMI KANNIAH

B. CONSENT FORM

I have read the information presented in the information letter about a study being conducted by Sri Lakshmi Kanniah of the Advanced Informatics School at the Universiti Teknologi Malaysia (UTM). I have had the opportunity to ask any questions related to this study, to receive satisfactory answers to my questions, and any additional details I wanted.

I am aware that I have the option of allowing my interview to be audio recorded to ensure an accurate recording of my responses.

I am also aware that excerpts from the interview may be included in the thesis and/or publications to come from this research, with the understanding that the quotations will be anonymous.

I was informed that I may withdraw my consent at any time without penalty by advising the researcher.

With full knowledge of all foregoing, I agree, of my own free will, to participate in this study.

□YES □NO

I agree to have my interview audio recorded.

□YES □NO

I agree to the use of anonymous quotations in any thesis or publication that comes of this research.

□YES □NO

Participant Name: ______________________________

Participant Signature: ______________________________

Date: ______________________________

C. CONTACT SUMMARY

Contact: (visit/phone/email)	Contact Date:
Venue:	
Detail of the contact person :	
Interviewee ID :	Phone Number:
Name :	
Position:	

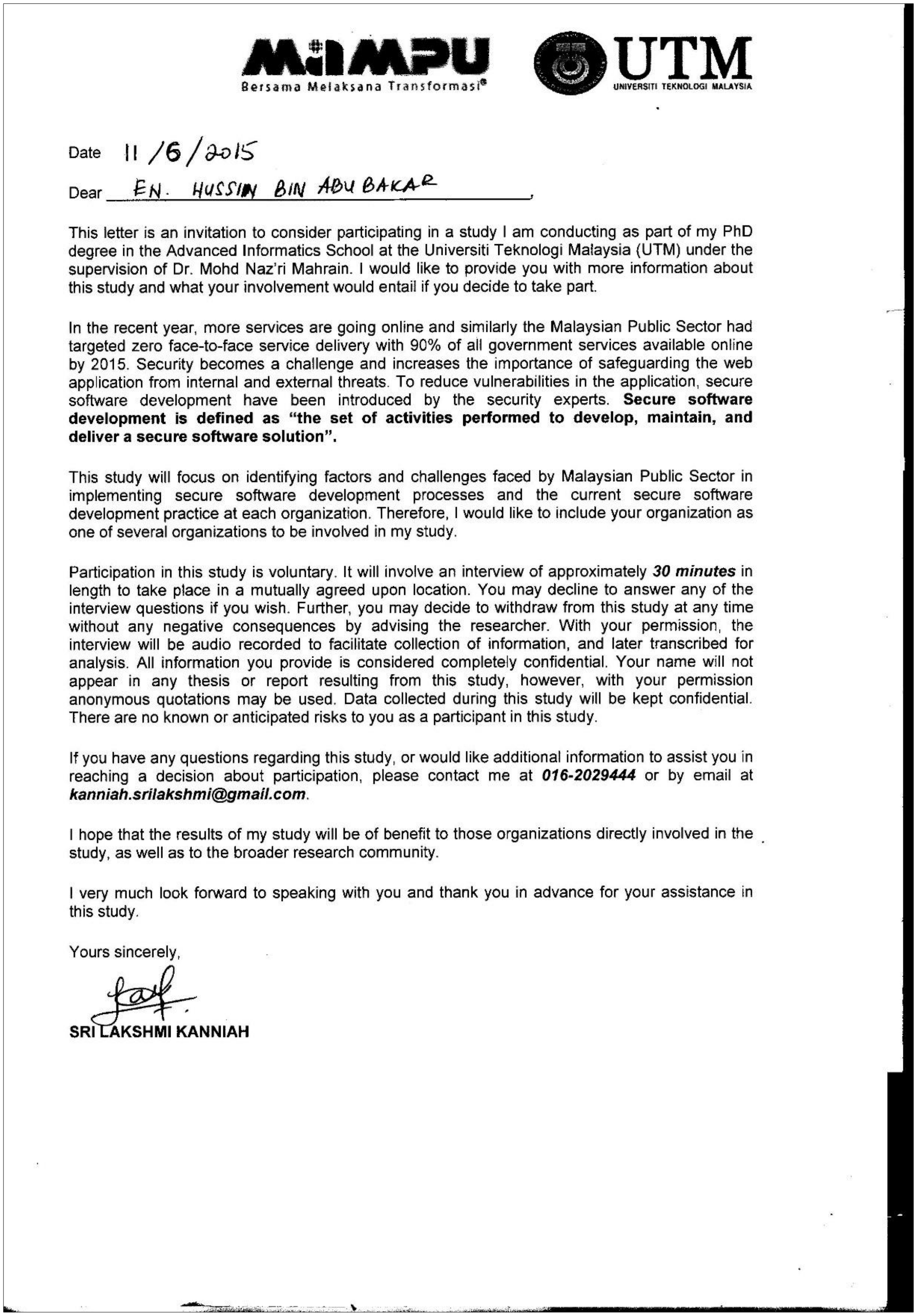
D. INTERVIEW QUESTIONS

The objective of this interview is to understand the need for secure software development process at Malaysia Public Sector. Secure software development process can be defined as the set of activities performed to develop, maintain, and deliver a secure software solution. The interview session will take approximately 30 minutes.

No.	Category of Information	Questions
1.	Information Category 1: Demographic questions	1. How long have you been involved in software development? 2. What is your role in software development? 3. Who do you report to? 4. What software development methodology is preferred or used in government? 5. Is your organization certified by ISO/IEC27001? a. If yes, how many services? Does it include all controls under System acquisition, Development and Maintenance? b. If no, why?
2.	Information Category 2: Questions relating to factors identification	Security has become a major issue in software development. Vulnerabilities in software enable hackers to compromise and steal information. Nowadays hackers' targets are focused on software as the networks are well guarded through the implementations of firewalls and intrusion detection systems. One way to protect our software is by implementing secure software development practices throughout the development lifecycle. 6. What policies exist to facilitate the implementation of secure software development practices in government? 7. Does all the software produced by your organization follow guidelines provided by standards? a. If yes, which standard? b. If no, why? 8. What are the problems/issues faced by your organization in implementing secure software development practices at your organization? 9. How can your organization improve the security of the software produced at your organization? 10. Who do you think play an important role in developing secure software? 11. What are the current secure software development practices at your organization? 12. Is your organization ready to adopt existing secure software development standards/models?

B.2 Sample of Field Note

B.2.1 Invitation Letter



MAMPU
Bersama Melaksana Transformasi®

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Date 11/6/2015

Dear EN. HUSSIN BIN ABU BAKAR,

This letter is an invitation to consider participating in a study I am conducting as part of my PhD degree in the Advanced Informatics School at the Universiti Teknologi Malaysia (UTM) under the supervision of Dr. Mohd Naz'ri Mahrain. I would like to provide you with more information about this study and what your involvement would entail if you decide to take part.

In the recent year, more services are going online and similarly the Malaysian Public Sector had targeted zero face-to-face service delivery with 90% of all government services available online by 2015. Security becomes a challenge and increases the importance of safeguarding the web application from internal and external threats. To reduce vulnerabilities in the application, secure software development have been introduced by the security experts. **Secure software development is defined as "the set of activities performed to develop, maintain, and deliver a secure software solution".**

This study will focus on identifying factors and challenges faced by Malaysian Public Sector in implementing secure software development processes and the current secure software development practice at each organization. Therefore, I would like to include your organization as one of several organizations to be involved in my study.

Participation in this study is voluntary. It will involve an interview of approximately ***30 minutes*** in length to take place in a mutually agreed upon location. You may decline to answer any of the interview questions if you wish. Further, you may decide to withdraw from this study at any time without any negative consequences by advising the researcher. With your permission, the interview will be audio recorded to facilitate collection of information, and later transcribed for analysis. All information you provide is considered completely confidential. Your name will not appear in any thesis or report resulting from this study, however, with your permission anonymous quotations may be used. Data collected during this study will be kept confidential. There are no known or anticipated risks to you as a participant in this study.

If you have any questions regarding this study, or would like additional information to assist you in reaching a decision about participation, please contact me at ***016-2029444*** or by email at ***kanniah.srilakshmi@gmail.com***.

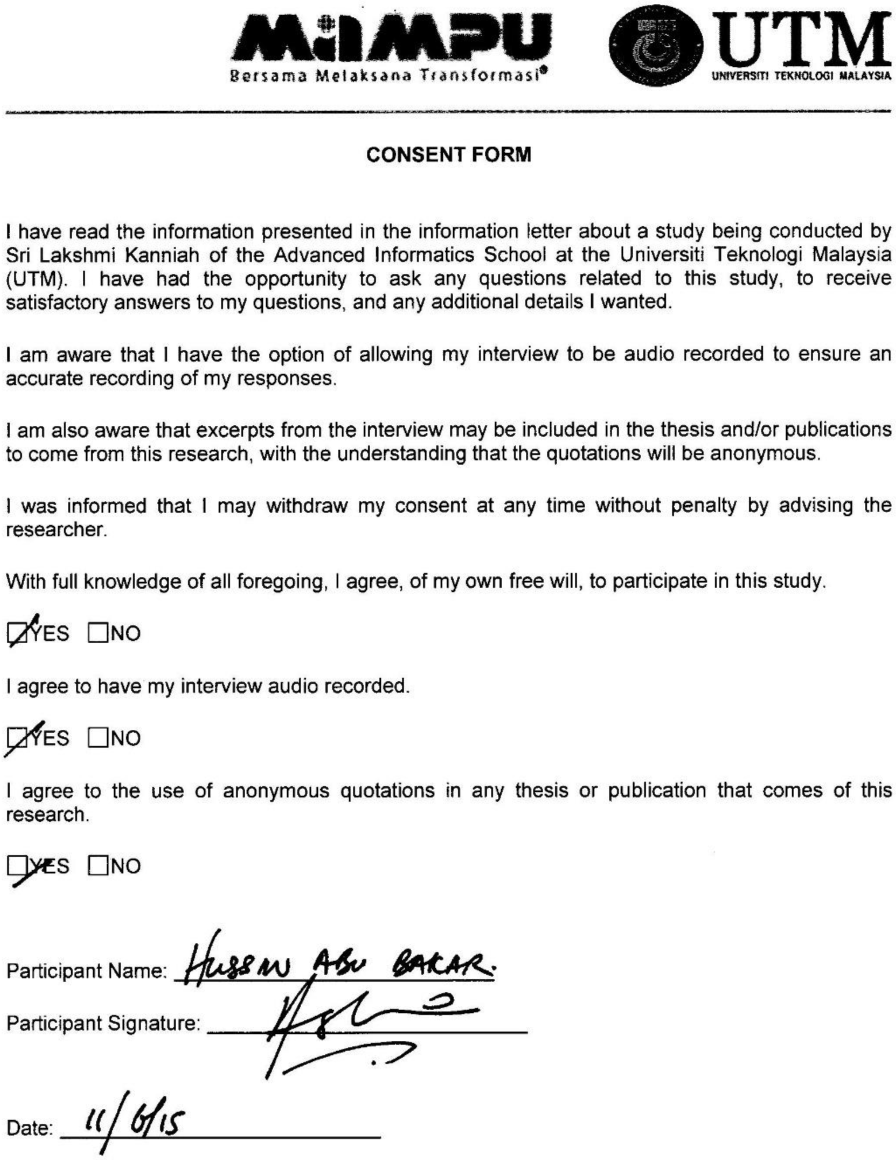
I hope that the results of my study will be of benefit to those organizations directly involved in the study, as well as to the broader research community.

I very much look forward to speaking with you and thank you in advance for your assistance in this study.

Yours sincerely,

SRI LAKSHMI KANNIAH

B.2.2 Consent Form



MAMPU
Bersama Melaksana Transformasi®

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

CONSENT FORM

I have read the information presented in the information letter about a study being conducted by Sri Lakshmi Kanniah of the Advanced Informatics School at the Universiti Teknologi Malaysia (UTM). I have had the opportunity to ask any questions related to this study, to receive satisfactory answers to my questions, and any additional details I wanted.

I am aware that I have the option of allowing my interview to be audio recorded to ensure an accurate recording of my responses.

I am also aware that excerpts from the interview may be included in the thesis and/or publications to come from this research, with the understanding that the quotations will be anonymous.

I was informed that I may withdraw my consent at any time without penalty by advising the researcher.

With full knowledge of all foregoing, I agree, of my own free will, to participate in this study.

☑YES □NO

I agree to have my interview audio recorded.

☑YES □NO

I agree to the use of anonymous quotations in any thesis or publication that comes of this research.

☑YES □NO

Participant Name: HUSSIN ABU BAKAR.

Participant Signature: ____________

Date: 11/6/15

B.2.3 Contact Summary

Contact: Visit	Contact Date: 11 Jun 2026
Venue: MAMPU, Cyberjaya	
Detail of the contact person :	
Interviewee ID : R3	Phone Number: -
Name : En. Hussin bin Abu Bakar	
Position: ICT Consultant	

Appendix C   Example of Interview Transcript

Respondent 15

1. How long have you been involved in software development?
   34 years.

2. What is your role in software development?
   Head of system development for consulting services to public sector. Currently my role is to provide advisory and consultancy for all agencies in public sector with regards to anything to do with software development.

3. Who do you report to?
   Government Chief Information Office (GCIO)

4. What software development methodology is preferred or used in government?
   Own methodology. Customized methodology. Adaptation of waterfall and agile methodology. Look at the situation before deciding which methodology to use.

5. Is your organization certified by ISO/IEC27001?
   MAMPU is certified but not for system development. Only security services are certified.

a. If yes, how many services? Does it include all controls under System acquisition, Development and Maintenance?

b. If no, why?
   For us to go for certification we must be ready first. We are not ready for certification. Each agency they have their own way of working. We are providing consulting services. The way I look at it consulting services are the one we need to certify rather than the actual development process. These are two different things. Each agency they have their own development outfit then they can look at their own way to certify their product but we are not into that business. We are providing consulting and educational services. For that we can go for certification and we are not ready yet. But of course in longer term the plan is to get that certified. In MAMPU we have got a whole division called application development division.

6. What policies exist to facilitate the implementation of secure software development practices in government?
   The way I look at it, from what I know there are no policies. There are just some guidelines that when you do development the major thing that you must focus is that security is one key component. When we do development we always talk about the 2 different aspects of development. One is the functional requirement and the other is the non-functional aspect. And if you look at the non-functional aspect, one of the key areas that we focus upon is security. So when you get the requirements for security, we have to make sure that the requirements are very well defined and then the design takes into account all your requirements. And at the end of the day when you do acceptance you make sure that all these things are tested and fulfil the security requirement. Basically that is what we advise. Previously the focus has always been on functional aspect. They did not know what non-functional requirements are. But now that awareness is already there. Of course there are other aspects in non-functional requirement but the key aspect that we always stress upon is the security requirement is derived and all build the system where these requirements are addressed. That is a thing that we do.

   Where do the security requirements come from? Do you have some kind of policy or guideline?
   As far as I'm concern we don't have security policy but probably the security division might have security guidelines.

   We have DKICT. Do you think DKICT can be used as a security guideline?

Whatever which is relevant in DKICT will become part of the security requirements. When you talk about security requirements for an application, first of all you must look at the person relevant who are a party to the particular application and of course indirectly the DKICT may become an input. I do not know whether you want to call that as a policy because that is "dokumen keselamatan" and every organization has it but to me it is very general. So for each application it has its own unique and peculiar requirement. So all those things should be taken into account. The overriding things is each application you have define clearly what your security requirements are and for that you have to get the right people to come on board to provide you with all the inputs and together with that all other relevant documents. The other thing we have advising agencies is when it comes to applications which are deemed to be highly sensitive or critical or has to be secured what we tell them is go for third party testing. When they go for third party testing, one of the component that we ask the third party to test is the security. That is done post development. When you talk about security testing, the third party can actually come in from the very beginning of the project where they look at the documents to see whether all requirements are covered, designed and tested. In my point of view, when you at it holistically, security testing is definitely not post development. It starts right from the day that you start requirements. That is how I advise people along that line. To add credibility to public service application, we get third party to test application because they have the tools and expertise to test. Those tools are very expensive. We are not in the position to get those tools so we get third party to come in.

7. Does all the software produced by your organization follow guidelines provided by standards?
   No.

b. If no, why?
One of the reason that this org don't go for this is personal awareness. They are not aware that these standards exist and they can make use of these standards. When we prepare non-functional requirement in fact we make references to ISO documents to see what relevant requirements come under the umbrella of software security and all its components. But when you talk about organization, probably I think they are not aware or the other reason could be they are not ready or they don't have the expertise. In most cases you find that those big projects are actually outsourced. There are very few organization actually do in-house development. If they do in-house development most cases they only focus on small application which does not require that much of security. So when you outsource this they have to actually ensure that the security aspects are covered very well. from what I know, when they talk about security requirements, they don't talk about these standards or fulfilling what ISO requirements is. Or if they were to give a proposal for whatever tender the company itself we don't see them actually addressing this kind of issue. But if you talk about capability maturity model and all this kind of things I do not know. I've not seen any company that proudly says that we are CMMI level 3, 4 or 5. I've not seen any. Probably the companies out there do not see this as requirement for the government. There is no incentive for them to be CMMI certified. If they don't have it there, then of course when they come in they cannot propose that. If the government wants to insist on that then they might not get any companies to participate. It's like a chicken an egg kind of thing. The government wants the companies to be CMM level 4 and above. Can we get that? It's difficult. When you talk to most organizations they don't even understand CMMI also and what does it involve. It'll take years for you to come to that level. Maybe there are one or two companies. But not sufficient to fulfill government procedure (government must have more than 3 companies to participate in tender). This issues needs to be addressed holistically. It's not only government 's decision. Even if government ask for it, are there enough companies?
The constraints other than not having awareness the other thing is cost and also timeline which is given to us. Sometimes the stakeholders they want projects to be done in 3 months. Sometime you find that you have to let go of some things. So the easiest way to address this is what they know is only the functional aspect of the software. They provide functionality requirements because that is what the end user wants to see. Security and all other non-functional requirements are at the back. The user doesn't see. So this is what actually happens. They get around this and say later we will fix that but that never happens and damage will be done. When damage is done they have no choice but to address the issue later. The other constraint is cost. When you want address security issues, there is an element of cost. Let's say if they not conversant in doing whatever required to meet security requirement then you have to get enough budget. In most cases when agencies do their budgeting, the way the come out with budget there is no proper analysis or evaluation done before coming out with actual cause needed. Security and non-

functional requirements nobody bothers about that. This project is more or less like this and just put a figure. Usually you find that what they do is the easiest way that this people go about doing. When they look at mandays they see how many that they can develop. How many functionalities I have to provide. They never take into account on how much security needed, what is response time, robustness of the system. So when you put the system on board then you find the system fails. System fails because not meeting non-functional aspects. For them non-functional aspects are always involves cost. In most cases, you have to buy something else. You have upgrade software, get external expertise, security software embedded. These are the contributory factors leading to the government not being able to provide those kind of so called secured software. As I said earlier, for those projects deemed as critical and high impact what we do know is through our JTICT which MAMPU chairs we have made that as one of the requirements. If we evaluate and find that the project falls under that category that is one of the "syarat kelululusan" which is to get third party tester to come in. when the third part tester comes in there are various components that the test and one of it that they have to test is security. So at least we are confident that before it is rolled out to the client we are reasonably confident that the application meets all requirement including security requirements. Once this thing takes a route in all the agencies then they will be able to see what third party testing is all about. And on their own they can initiate that security on aspect that we have to take account. When it comes to higher level management, they have to be told about the importance, criticality and need for having security testing as one key component in both the costing and timeline and the value it brings. Management would not know until you tell them. When you talk about the highest level of management it always involve non-IT people (jawatankuasa Pemandu). It's the job of the IT personnel to inform the management on what the implications are if they don't address non-functional requirement. If don't allocate cost and time this is the risk that you take. You tell them upfront. But that never happens now. Risk assessment is not there at all. Probably all they want is to get their paper approved and get on to their project with little consideration on non-functional requirements.

8. What are the problems/issues faced by your organization in implementing secure software development practices at your organization?
   First the security guidelines must be clearly understood what the whole is about. The human nature is that if they don't understand something, they won' do it. If the guidelines require specialist knowledge and they don't have it, then it becomes a key deterrent in implementing it. Secondly the issue is by following the guidelines they will weigh. If I implement these guidelines then I have to spend another three months on this project. If it will cost me another one million. These are factors that can actually affect the implementation of these. Unless these are costed in the project plan, these things will never take off. People must educated with the importance of security. These must be mandated.

9. How can your organization improve the security of the software produced at your organization?
   First, we must have a security policy or document that will be a guide for all agencies. Once the guideline is there, there must be training and awareness and the agencies must have someone who is well versed in the guidelines to look over the implementation of the guidelines. There is no check and balance. Each agency suppose to have ICTSO. But I do not know what their role is. These task should be included into their role and make sure the security governance is in place. there must be someone monitoring this. Otherwise they will never get this done.

10. Who do you think play an important role in developing secure software?
    I would think the developers and security personnel. When I say developers I'm talking about requirement people, designers and coders. They must have knowledge in secure coding. But the requirements must come from people who knows about security. Before they do acceptance these security personnel must test and verify that the requirements are fulfilled. The security personnel must be made responsible.

11. What are the current secure software development practices at your organization?

12. Is your organization ready to adopt existing secure software development standards/models?
    We are willing to accept but whether we are ready or not we must fulfil the factors mentioned earlier.

Appendix D Questionnaire for Identification of Secure Software Development Practices for Malaysian Public Sector

IDENTIFICATION OF SECURE SOFTWARE DEVELOPMENT ACTIVITIES FOR MALAYSIAN PUBLIC SECTOR

The objective of this questionnaire is to seek information in identifying secure software development activities for Malaysian Public Sector.

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

The questionnaire form encompasses three parts which are Section A, Section B and Section C of 11 pages in total. **Section A** is about the demography of the respondents while **Section B** is related to potential secure software development activities for Malaysian Public Sector. **Section C** is about software security threats at your organization.

The respondents are chosen from each ministry in Malaysian Public Sector. They are selected among IT managers and software developers who are responsible in software development in each ministry.

It is a pleasure if you could spend 15 minutes to respond to this questionnaire.

Thank you for your cooperation.

PhD Candidate:

Sri Lakshmi Kanniah
Advanced Informatics School (AIS)
Universiti Teknologi Malaysia (UTM),
Kuala Lumpur.
Email : *kanniah.srilakshmi@gmail.com*
Tel. No. : *016-2029444*

Supervisors:

Dr. Mohd Naz’ri Mahrin
Advanced Informatics School (AIS),
Universiti Teknologi Malaysia (UTM),
Kuala Lumpur.
Email: mdnazrim@utm.my

SECTION A: DEMOGRAPHY OF THE RESPONDENTS

Instruction: Please tick □ in appropriate box. Respondents must tick ONLY ONE options.

1. Ministry / *Kementerian*:	
□ Ministry of Finance *Kementerian Kewangan*	□ Ministry of Federal Territories *Kementerian Wilayah Persekutuan*
□ Ministry of Education *Kementerian Pendidikan*	□ Ministry of Defence *Kementerian Pertahanan*
□ Ministry of Transport *Kementerian Pengangkutan*	□Ministry of Agriculture and Agro-based Industry *Kementerian Pertanian dan Industri Asas Tani*
□ Ministry of Plantation Industries and Commodities *Kementerian Perusahaan Perladangan dan Komoditi*	□ Ministry of Works *Kementerian Kerja Raya*
□ Ministry of Home Affairs *Kementerian Dalam Negeri*	□ Ministry of Health *Kementerian Kesihatan*
□ Ministry of Communication and Multimedia *Kementerian Komunikasi Dan Multimedia*	□ Ministry of Youth and Sports *Kementerian Belia dan Sukan*
□ Ministry of Energy, Green Technology and Water *Kementerian Tenaga, Teknologi Hijau dan Air*	□ Ministry of Human Resources *Kementerian Sumber Manusia*
□ Ministry of Rural and Regional Development *Kementerian Kemajuan Luar Bandar dan Wilayah*	□ Ministry of Urban Wellbeing, Housing and Local Government *Kementerian Kesejahteraan Bandar, Perumahan Dan Kerajaan Tempatan*
□ Ministry of International Trade and Industry *Kementerian Perdagangan Antarabangsa dan Industri*	□ Ministry of Foreign Affairs *Kementerian Luar Negeri*
□ Ministry of Science, Technology and Innovation *Kementerian Sains, Teknologi dan Inovasi*	□ Ministry of Tourism and Culture *Kementerian Pelancongan dan Kebudayaan*
□ Ministry of Women, Family and Community Development *Kementerian Pembangunan Wanita, Keluarga dan Masyarakat*	□ Ministry of Domestic Trade, Cooperative and Consumerism *Kementerian Perdagangan Dalam Negeri dan Hal Ehwal pengguna*
□ Ministry of Natural Resources and Environment *Kementerian Sumber Asli dan Alam Sekitar*	□ Malaysian Administrative Modernisation and Management Planning Unit (MAMPU) *Unit Pemodenan Tadbiran Dan Perancangan Pengurusan Malaysia (MAMPU)*
□ Others: ______________________________	□ Public Service Department *Jabatan Perkhidmatan Awam*
2 Gender : □ Male □ Female ***Jantina*** *Lelaki* *Perempuan*	
3. Age : _______ in years *(dalam tahun)* ***Umur***	
4. Service Scheme : □ Top Management ***Kumpulan Perkhidmatan*** *Pengurusan Tertinggi* □ Management & Professional (Gred 41–54)	

Pengurusan & Profesional (Gred 41–54)

□ Support Group (Gred 29-38)
Kumpulan Sokongan (Gred 29–38)

5. Your experience in software product development :
Pengalaman dalam pembangunan system

□ Less than one year
Kurang 1 tahun

□ 1 – 5 years
1 – 5 tahun

□ 6 – 10 years
6 – 10 tahun

□ More than 10 years
Lebih daripada 10 tahun

6. Do you have awareness and knowledge on security practices in software development?
Adakah anda mempunyai kesedaran dan pengetahuan mengenai amalan keselamatan dalam pembangunan sistem?

□ Yes / Ya □ No / Tidak

7. Your role in software product development:
Peranan anda dalam pembangunan sistem

□ Project Manager
□ Requirement Specifier
□ Designer
□ Architect
□ Tester
□ Security Auditor

SECTION B: SECURITY PRACTICES IN SOFTWARE DEVELOPMENT

The following statements are the secure development practices that can reduce security vulnerabilities in the software. Please tick to indicate the level of agreement for each practice.

A	**Institute Security Awareness Program** *Purpose : i)To ensure project members consider security to be an important project goal through training and accountability.* *ii)Ensure project members have enough exposure to security to deal with it effectively.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
1	Provide security training to all team members					
2	Distribute and present security requirements to all team members before development					
3	Project managers must assess to see whether the developers are following the security guidelines given from time to time.					
4	Appoint a project security officer for each individual project					
5	Reward developers for following security guidelines consistently over a period of time					
B	**Monitoring Security Metrics** *Purpose : i) Gauge the likely security posture of the ongoing development effort.* *ii) Enforce accountability for inadequate security.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
6	Identify all security metrics that can be used to determine security posture of the software at the beginning of the project					
7	Monitor the usage of the metric to evaluate the effectiveness of the metric.					
8	Strategize data collection and produce output report in appropriate format for the team					
9	Periodically collect and evaluate metrics					
C	**Specify operational environment** *Purpose :Document assumptions and requirements about the operating environment, so that the impact on security can be assessed.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
10	Identify requirements and assumptions related to operating system and its components that could have security impact on the software					
11	Identify requirements and assumptions related to network architecture and resources such as databases and bandwidth that could have security impact on the software					

D	**Identify global security policy** *Purpose: i) Provide default baseline product security business requirements.* *ii)Provide a way to compare the security posture of different products across an organization.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
12	Build a global project security policy, if necessary					
13	Determine suitability of global requirements to project					
E	**Identify resources and trust boundaries** *Purpose: Provide a structured foundation for understanding the security requirements of a system.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
14	Describe the architecture of the system from the perspective of the network					
15	Identify data resources such as databases and Access Control List(ACL)					
F	**Identify user roles and resource capabilities** *Purpose: Define system roles and the capabilities/resources that the role can access.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
16	Identify capabilities (read, write, execute, create, and delete) for files and databases used in the project.					
17	Map system roles (e.g. administrator, users and guest) to capabilities					
18	Identify the attacker profile (insiders, “Script Kiddies”, Competitors, Government, Activist) what they want to gain					
G	**Document security-relevant requirements** *Purpose: Document business-level and functional requirements for security.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
19	Document clear business requirements					
20	Develop functional security requirements showing how the basic security services are addressed for each resource in the project					
21	Specify all third party components required in the project					
22	Specify mechanisms to address potential security risk for each resource					
23	Resolve deficiencies and conflicts between business, functional and global requirements					
H	**Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.** *Purpose: i) Communicate potential risks to stakeholder.* *ii) Communicate rationale for security-relevant decisions to stakeholder.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
24	Identify misuse cases for each actor present in the system					
25	Describe and document misuse cases					
26	Identify defense mechanisms for misuse cases					

27	Review and discuss the misuse case with stakeholders, so that they have a clear understanding of the misuse case					
I	**Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.** *Purpose: Specify all entry points to a program in a structured way to facilitate analysis.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
28	Define the specific mechanisms through which anyone could interact with the application regardless of their role in the system					
29	Identify all roles that could possibly access the defined entry point.					
30	For each entry point, document the resources that should be accessible from that entry point					
J	**Apply security principles to design** *Purpose*: *i) Harden application design by applying security design principles.* *ii)Identify security risks in third-party components.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
31	Refine existing application security in the system design					
32	Identify solutions for meeting security requirements at each identified point in the design					
33	Build hardened protocol specifications such as SSL/TLS					
34	Design hardened API interfaces					
K	**Research and assess security posture of technology solutions** *Purpose: i)Assess security risks in third-party components.* *ii)Determine how effective a technology is likely to be at reducing risks.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
35	Get structured technology assessment from vendor before integrating into the system					
36	Perform security risk assessment on vendor products					
37	Receive permission to perform security testing of vendor products					
38	Perform security testing on vendor products					
L	**Annotate class designs with security properties** *Purpose: Elaborate security policies for individual data fields.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
39	Map each data element back to the requirements to determine the requirements on that data					
40	For each data field define the owning role or roles and which role or roles have access to which basic capabilities throughout the lifetime of the data					
41	Annotate methods to identify which operations they perform on data					
M	**Specify database security configuration** *Purpose: i) Define a secure default configuration for database resources that are deployed as part of an implementation.* *ii) Identify a recommended configuration for database resources for databases that are deployed by a third party.*					

		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
42	Identify candidate database configuration					
43	Validate that the baseline configuration properly addresses the security requirements for that database.					
N	**Perform security analysis of system requirements and design (threat modeling)** *Purpose: i) Assess likely system risks in a timely and cost-effective manner by analyzing the requirements and design.* *ii) Identify high-level system threats that are documented neither in requirements nor in supplemental documentation.* *iii) Identify inadequate or improper security requirements.* *iv) Assess the security impact of non-security requirements.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
44	Before performing a security analysis, review all existing high-level system documentation such as user manuals and architectural documentation					
45	Review non-security requirements					
46	Assess completeness of security requirements					
47	Identify threats on assets/capabilities					
48	Determine level of risk					
49	For each identified risk, identify any feasible approaches for mitigating the risk and evaluate their cost and effectiveness					
50	Evaluate findings, determine whether the assessments are actually correct to the business and make risk-based decisions based on this information.					
O	**Integrate security analysis into source code management process** *Purpose: Automate implementation-level security analysis and metrics collection.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
51	Select dynamic or static analysis tools to be integrated into development process					
52	Determine analysis integration point (check-in process, as part of the build process, or independently)					
53	Integrate analysis technology					
P	**Implement interface contracts** *Purpose: i) Provide unit-level semantic input validation.* *ii) Identify reliability errors in a structured way at the earliest point in time.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
54	Implement validation and error handling on each function or method inputs					
55	Implement validation on each function or method outputs					
Q	**Implement and elaborate resource policies and security technologies** *Purpose: Implement security functionality to specification*					

		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
56	The developer should identify any remaining ambiguities in the specification of security properties or technologies					
57	The implementor should ensure that all coding guidelines are met — especially security guidelines					
R	**Address reported security issues** *Purpose: Ensure that identified security risks in an implementation are properly considered.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
58	When a security risk/issue is identified in a system, further investigation should be assigned to the appropriate designer / architect					
59	Assess likely exposure and impact of the issue/risk					
60	Determine and execute short term or long-term remediation strategies					
61	Perform testing to ensure that the issue/risk was properly addressed					
S	**Perform source-level security review** *Purpose: Find security vulnerabilities introduced into implementation*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
62	Scope out the areas that merit the most attention before performing source-level security review					
63	Run automated analysis tools					
64	Evaluate each potential risk identified by the tool					
65	Identify additional risks by reviewing both those risks identified in the architectural analysis and a database of common risks.					
T	**Identify, implement and perform security tests** *Purpose: i) Find security problems not found by implementation review.* *ii) Find security risks introduced by the operational environment.* *iii) Act as a defense-in-depth mechanism, catching failures in design, specification, or implementation.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
66	Identify security tests for individual requirements					
67	Identify tests that will determine which system roles can gain access to each resource					
68	Using a common testing checklist, determine what other security tests are appropriate to the system					
69	Implement test plan					
70	Perform the identified security tests as specified in the test plan					

U	**Verify security attributes of resources** *Purpose: Confirm that software abides by previously defined security policies.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
71	Check whether permissions granted by the system's default install exactly match those put forth by the resource specifier in the security requirements					
72	Specify in the requirements, a security profile or operational security guide what resources the system should be able to access					
V	**Perform code signing** *Purpose: Provide the stakeholder with a way to validate the origin and integrity of the software.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
73	Obtain code signing credentials(e.g. PKI, CA)					
74	Identify signing targets such as a single archive file (JAR, WAR, or CAB)					
75	Sign identified targets					
W	**Build operational security guide** *Purpose: i)Provide stakeholder with documentation on operational security measures that can better secure the product.* *ii) Provide documentation for the use of security functionality within the product.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
76	Document pre-install configuration requirements					
77	Document application activity including network ports, files on the file system, registry resources, database resources					
78	Document the security architecture including authentication mechanisms, default policies for authentication and other functions, and any security protocols that are mandatory or optional					
79	Document security configuration mechanisms					
80	Document significant risks and known compensating controls					
X	**Manage security issue disclosure process** *Purpose: i) Communicate effectively with outside security researchers when security issues are identified in released software, facilitating more effective prevention technologies.* *ii) Communicate effectively with customers when security issues are identified in released software.*					
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
81	A central security response alias should be established, such as security@ or secalert@ and published on the web site if possible					
82	On receipt of the vulnerability disclosure, respond with acknowledgement of receipt, as well as a reasonable timetable for addressing the vulnerability.					

83	The reported vulnerability should be entered into the process for dealing with reported security issues					
84	Communicate relevant information to the researcher					
85	Provide a security advisory and customer access to remediation					

SECTION C: SOFTWARE SECURITY THREATS

Please indicate the level of agreement with the following statement on software security threats at your organization.

		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
86	Software produced by government organization is prone to security threats.					
87	Software owned by my organization is highly secured.					
88	My organization would face significant business disruption if the software is compromised					
89	My organization spends a lot on software security and resolutions					
90	My organization implements secure software development throughout the development lifecycle.					
91	I'm well trained in implementing secure software development practices					

Appendix E Delphi Questionnaire

A SECURE SOFTWARE DEVELOPMENT PRACTICE SELECTION MODEL FOR MALAYSIAN PUBLIC SECTOR

RESEARCH INFORMATION

Thank you for your interest in this study. Please read the information in this section carefully before you begin to answer the questionnaires at the following section.

Description: The whole model development on this study will be based on Delphi technique. The Delphi technique consist of three (3) phases with each phase carrying specific objectives as stated below:

PHASE 1:	Participants are required to state their level of agreement on factors that influence implementation of SSD practices in public sector;
PHASE 2	Participants are required to state their level of agreement on indicators for assessing the factors achieved by the organization; and
PHASE 3	Participants are required to map the factors to SSD practice being influenced by the factor.

Each phase will be conducted in minimum two (2) rounds of survey or until consensus among the respondents is achieved.

As part of my doctoral dissertation at Advanced Informatics School (AIS), Universiti Teknologi Malaysia (UTM) you are kindly invited to participate in all three (3) phases of Delphi study which aims at developing Secure Software Development (SSD) practice Selection model based on the factors. Therefore, I am requesting for your kind cooperation in giving your time, experience and thoughts by answering the questionnaire form provided. Your cooperation is most essential as it could be beneficial to both industry and academia.

Importance Definition: Within the context of this research, Secure software development is defined as "the set of activities performed to develop, maintain, and deliver a secure software solution". SSD practices listed in this research were adopted from the Comprehensive Lightweight Application Security Process (CLASP). CLASP provides well organized and structured approach for locating security concerns into the phases of software development lifecycle.

Privacy Protection: Please be assured that all responses in this questionnaire would be kept strictly confidential and will only be used for academic purposes only. If you have any additional query about this research, please contact me at kanniah.srilakshmi@gmail.com or my supervisor, Dr. Mohd. Naz'ri bin Mahrin at mdnazrim@utm.my .

Yours Sincerely,

Sri Lakshmi Kanniah
Advanced Informatics School (AIS)
Universiti Teknologi Malaysia (UTM), KL
Email : kanniah.srilakshmi@gmail.com
Tel. No. : 016-2029444

Supervisor:
Dr. Mohd Naz'ri Mahrin
Advanced Informatics School (AIS),
Universiti Teknologi Malaysia(UTM) KL
Email: mdnazrim@utm.my .

DELPHI SURVEY – PHASE 1

A Secure Software Development Practice Selection Model for Malaysian Public Sector

The objective of ***Phase 1 survey*** is to seek information from experts to determine factors that influence implementation of Secure Software Development practices in public sector and to suggest new factors, if any. (*estimated length of survey is 10 mins*)

Profile

Name:

Years of Experience in Software Development:

Job Title:

Please name the software development projects you were involved in:

List Professional Certificates:

Findings from literature and case studies conducted indicate that Secure Software Development practice implementation in the public sector organisation may be influenced by various factors. These factors are presented in the following table. Which of these factors do you think is influences implementation of Secure Software Development practices in Malaysian Public Service Organization?

Factors that influences the implementation Secure Software Development Practices in Malaysian Public Service Organization	*How strongly do you agree with the factor on the left? (□ tick only one)*				
	Strongly Agree	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Institutional Context					
Change Management *(Involves strategies and techniques required to encourage acceptance and support for implementation of new practices)*					
Policy Enforcement *(Enforcement of policies by organization which leaves the developers with no choice but to follow SSD practices)*					
Security Training and Awareness *(Effective security training and awareness plan to provide skill and knowledge on SSD practices)*					
Reward and Incentives *(Providing incentives and rewards to team members who responsibly secure the system)*					
Organization's objectives and culture *(The organization‘s objective and culture aligned with SSD implementation)*					
People and Action					
Developer *(The attitude, motivation and skills of a developer to implement SSD practices)*					
Top Management *(Willingly providing support from Management to the other team)*					
Security Experts *(A group experts who can provide consultation advice on software security)*					
Project Manager *(A competent individual heading the project and ensure SSD practices are implemented throughout the software development lifecycle)*					
Please state your comments here:					

Factors that influences the implementation Secure Software Development Practices in Malaysian Public Service Organization	*How strongly do you agree with the factor on the left?* *(□ tick only one)*				
	Strongly Agree	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Project Content					
Automated tool support *(Automated security tools that can facilitate secure software development)*					
Cost *(Adequate budget is allocated to implement SSD practices)*					
Project Team *(A dedicated team with members from various functions and expertise working to develop a common software)*					
Security Audit Team *(A team of security experts who are able to verify and validate the security aspects of the system before production)*					
Segregation of role *(Each team member to be given specific roles)*					
Team size *(Team size is relevant to the size of the project)*					
Team Collaboration *(Working together and the basis for bringing together the knowledge, experience and skills of team members)*					
Development Time *(Adequate development time to allow implementation of secure development practices)*					
Security Documentation *(Documentation of all security practices implemented for each project)*					
Software development methodology *(A standard method that provides an element of control over the sequence of development activities)*					
Internal Metrics and KPI *(Establishment of internal metrics and key performance indicators that can be used to determine the progress and success of the organization's security evolution)*					
Please state your comments here: **I recommend secure software development practice to be adopted in public sector to produce a high quality product/application that exceeds customer expectation/needs**					

Factors that influences the implementation Secure Software Development Practices in Malaysian Public Service Organization	*How strongly do you agree with the factor on the left? (□ tick only one)*				
	Strongly Agree	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Please suggest any factor (if any) that is important to the successful implementation of SSD in public sector and rate it accordingly					

DELPHI SURVEY - PHASE 2

A Secure Software Development Practice Selection Model for Malaysian Public Sector

The objective of ***second phase survey*** is to seek information from experts to determine assessment criteria for each factor that influence implementation of Secure Software Development practices in public sector and to suggest new criteria, if any.

Information

Name :

Day/Time :

Place :

PhD Candidate:

Sri Lakshmi Kanniah
Advanced Informatics School (AIS)
Universiti Technologi Malaysia (UTM),
(UTM),
Kuala Lumpur.
Email : *kanniah.srilakshmi@gmail.com*
Tel. No. : *016-2029444*

Supervisors:

Dr. Mohd Naz'ri Mahrin
Advanced Informatics School (AIS),
Universiti Teknologi Malaysia

Kuala Lumpur
Email: mdnazrim@utm.my .

Based on the factors selected in Section 1, please state your level of agreement on the assessment criteria for each of the factor. Please provide suggestion for improving the indicator or suggest other indicators in the suggested column, if any.

Factor	**Assessment Criteria**	**How strongly do you agree with the factor on the left? (□ tick only one)**					**Other criteria / Suggestion for improvement**
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	
Change Management	Existence of Change Management Team						
	Change management strategies are well communicated with stakeholders.						
Policy Enforcement	SSD practices and procedures are continually monitored to ensure compliance with security policy						
	SSD practices and procedures are externally audited						
	SSD violations are reported to the proper authority						
Security Training and Awareness	Adequate SSD security training given to all developers						
	SSD policy is communicated well						
	Developers are educated or trained about new security policies						
	Developers aware of my information security roles and responsibilities						
	Top management and developers are aware of the risk of not following the SSD policy						

Factor	Assessment Criteria	How strongly do you agree with the factor on the left? (□ tick only one)					Other criteria / Suggestion for improvement
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	
	Developers are familiar with the SSD policy						
	Developers aware of the procedures for reporting security policy violation						
Reward and Incentives	Existence of reward policy						
	Developers are aware of the reward policy						
Organization's objectives and culture	Existence of a learning and development culture						
	Existence of a participative decision making culture						
	Existence of a support and collaboration culture						
	Existence of a power sharing culture						
	Existence of tolerance for conflicts and risk culture						
Developer	Existence of communication skills						
	Existence of IT management skills						
	Existence of planning skills						
	Existence of technical skills						
	Existence of SSD experience						

Factor	**Assessment Criteria**	**How strongly do you agree with the factor on the left? (□ tick only one)**					**Other criteria / Suggestion for improvement**
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	
	Existence of controlling skills						
Top Management	The degree to which functional managers willingly assign resources to the SSD implementation as they are needed						
	The degree to which the need for long-term SSD support resources is recognized by management						
	The degree to which executive management is enthusiastic about the possibilities of SSD						
	The degree to which all levels of management support the overall goals of the SSD						
Security Experts	Existence of sufficient security experts						
	Existence of communication skills						
	Existence of IT management skills						
	Existence of planning skills						
	Existence of technical skills						
	Existence of SSD experience						

Factor	Assessment Criteria	How strongly do you agree with the factor on the left? (□ tick only one)					Other criteria / Suggestion for improvement
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	
	Existence of controlling skills						
Project Manager	Existence of communication skills						
	Existence of IT management skills						
	Existence of planning skills						
	Existence of technical skills						
	Existence of SSD experience						
	Existence of controlling skills						
Automated tool support	Existence of tools to support secure software development (e.g static analyzer, penetration testing tools)						
	Existence of policy on using automated secure software development tools						
	Developers are trained to use the tool						
	Existence of complete technical documentation for the tools						
	Tools are compatible with development environment						
	Tools are easy to use						

Factor	Assessment Criteria	How strongly do you agree with the factor on the left? (□ tick only one)					Other criteria / Suggestion for improvement
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	
Cost	Sufficient budget is allocated for SSD implementation						
Project Team	Existence of the both business and technical knowledge into the project team						
	Existence of a balanced, cooperative, cross functional and full time project team						
	The degree to which project team performance is fairly compensated						
	Existence of the empowered project team members						
	The degree to which project team have prior experience in large IT projects.						
Security Audit Team	Existence of security audit team						
	Existence of well-defined audit procedures and has gained management's approval						
	Audit policies and procedure are clearly understood by audit team						

Factor	Assessment Criteria	How strongly do you agree with the factor on the left? (□ tick only one)					Other criteria / Suggestion for improvement
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	
Segregation of role	Team member's role and responsibilities are clearly defined and documented (eg. Requirement specifier, designer, tester)						
	Team members understand their role and responsibilities in a particular project						
Team size	Sufficient number of team members has been allocated for the project						
Team Collaboration	Existence of development team and security team in the organization						
	Existence of a channel where development team and security team communicates with each other (via meetings, forums or other communication channels)						
Development Time	Adequate development time is allocated for SSD implementation						
Software development methodology	Existence of a standard development methodology (e.g agile, Rapid prototyping, waterfall)						

Factor	**Assessment Criteria**	**How strongly do you agree with the factor on the left? (□ tick only one)**					**Other criteria / Suggestion for improvement**
		Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	
	All team members are aware of the activities involved in the methodology						
Security Reference Guide	Existence of a reference guide related to SSD implementation						
	Reference guides are clear and easy to be understood						
	Reference guides are easily accessible to developers						
Internal Security Metrics and KPI	Internal security metrics and KPI are documented						
	Internal security metrics and KPI are communicated well among the project team members						
	Implementation of internal security metrics and KPI are frequently monitored by project manager						

DELPHI SURVEY - PHASE 3

A Secure Software Development Practice Selection Model for Malaysian Public Sector

The objective of ***phase 3 survey*** is to seek information from experts to determine factor that influences implementation of each Secure Software Development practices in public sector.

Information

Name :

Day/Time :

Place :

PhD Candidate:	Supervisors:
Sri Lakshmi Kanniah	Dr. Mohd Naz'ri Mahrin
Advanced Informatics School (AIS)	Advanced Informatics School (AIS),
Universiti Technologi Malaysia (UTM),	Universiti Teknologi Malaysia (UTM),
Kuala Lumpur.	Kuala Lumpur
Email : *kanniah.srilakshmi@gmail.com*	Email: mdnazrim@utm.my .
Tel. No. : *016-2029444*	

Instruction: Findings from literature and case studies conducted indicate that SSD implementation in the public sector organisation may be influenced by various factors. These factors are presented in the following table. Which of these factors do you think is important to ensure a successful implementation of each SSD practice in Malaysian Public Service Organization? *(✓ you may tick more than one factor for each practice).*

SSD Practices (adopted from CLASP model)	**Institutional Context**					**People and Action Context**				**Project Content Context**								**System Development Processes Context**		
	Change Management	**Policy Enforcement**	**Security Training and Awareness**	**Reward and Incentives**	**Organization's objectives and culture**	**Developer**	**Top Management**	**Security Experts**	**Project Manager**	**Automated tool support**	**Cost**	**Project Team**	**Security Audit Team**	**Segregation of role**	**Team size**	**Team Collaboration**	**Development Time**	**Security Documentation**	**Software development methodology**	**Internal Security Metrics and KPI**
Institute Security Awareness Program																				
Monitoring Security Metrics																				
Specify operational environment																				
Identify global security policy																				
Identify resources and trust boundaries																				
Identify user roles and resource capabilities																				

SSD Practices (adopted from CLASP model)	Institutional Context					People and Action Context				Project Content Context								System Development Processes Context		
	Change Management	Policy Enforcement	Security Training and Awareness	Reward and Incentives	Organization's objectives and culture	Developer	Top Management	Security Experts	Project Manager	Automated tool support	Cost	Project Team	Security Audit Team	Segregation of role	Team size	Team Collaboration	Development Time	Security Documentation	Software development methodology	Internal Security Metrics and KPI
Document security-relevant requirements																				
Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.																				
Identify attack surface The system attack surface is the collection of possible entry points for an attacker.																				
Research and assess security posture of technology solutions																				
Annotate class designs with security properties																				

SSD Practices (adopted from CLASP model)	Institutional Context					People and Action Context				Project Content Context								System Development Processes Context		
	Change Management	Policy Enforcement	Security Training and Awareness	Reward and Incentives	Organization's objectives and culture	Developer	Top Management	Security Experts	Project Manager	Automated tool support	Cost	Project Team	Security Audit Team	Segregation of role	Team size	Team Collaboration	Development Time	Security Documentation	Software development methodology	Internal Security Metrics and KPI
Specify database security configuration																				
Perform security analysis of system requirements and design (threat modeling)																				
Integrate security analysis into source code management process																				
Implement interface contracts																				
Implement and elaborate resource policies and security technologies																				
Address reported security issues																				
Perform testing to ensure that the issue/risk was properly addressed																				

SSD Practices (adopted from CLASP model)	**Institutional Context**					**People and Action Context**				**Project Content Context**								**System Development Processes Context**		
	Change Management	**Policy Enforcement**	**Security Training and Awareness**	**Reward and Incentives**	**Organization's objectives and culture**	**Developer**	**Top Management**	**Security Experts**	**Project Manager**	**Automated tool support**	**Cost**	**Project Team**	**Security Audit Team**	**Segregation of role**	**Team size**	**Team Collaboration**	**Development Time**	**Security Documentation**	**Software development methodology**	**Internal Security Metrics and KPI**
Perform source-level security review																				
Identify, implement and perform security tests																				
Verify security attributes of resources																				
Perform code signing																				
Build operational security guide																				
Manage security issue disclosure process																				
Total																				

Appendix F Model Adoption Guidelines

Secure Software Development Practice Selection Model Adoption Guidelines

PURPOSE

This document outlines the Secure Software Development Practice Selection Model's adoption guidelines for the use software project managers in organizations that involved in selection of secure software development practices for individual software development projects. These guidelines consist of the action needed to be taken by the project managers during the adoption of the model to facilitate selection of suitable secure software development practices during software development phases. Secure Software Development Practice Selection Model consist of three elements: Factors, Assessment Criteria and Secure software development practices. All these elements have been agreed and validated by the software development experts from Malaysian Public Service Organization. The main purpose of this model is to assist Project Managers to select secure software development practices by assessing the influential factors.

RESEARCH DEFINITION

"Secure Software Development" defined as the set of activities performed to develop, maintain, and deliver a secure software solution.

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

"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.

"Assessment Criteria" refer to questions or statement used to identify the existence of the factor in the project.

"Project Manager" refers to an individual who leads a software development project and responsible of selecting secure software development practices for the project.

MODEL EXPLANATION

The description of influential factors and assessment criteria are explained here.

No.	Factor	Description	Assessment Criteria
1.	Change Management (CM)	Involves strategies and techniques required to encourage acceptance and support for implementation of new practices.	Existence of a formal management team in the organization.
			Change management strategies are well communicated with stakeholders.
2.	Policy Enforcement	Enforcement of policies by organization which leaves the developers with no choice but to follow SSD practices.	SSD practices and procedures are continually monitored to ensure compliance with security policy
			SSD practices and procedures are externally audited
			SSD violations are reported to the proper authority
			Actions against violations are always taken
3.	Security Training and Awareness	Effective security training and awareness plan to provide skill and knowledge on SSD practices.	SSD practices and procedures are continually monitored to ensure compliance with security policy
			SSD practices and procedures are externally audited
			SSD violations are reported to the proper authority
			Actions against violations are always taken
			SSD practices and procedures are continually monitored to ensure compliance with security policy
			SSD practices and procedures are externally audited
			SSD violations are reported to the proper authority
4.	Reward and Incentives	Providing incentives and rewards to team members who responsibly secure the system	Existence of reward policy
			Developers are aware of the reward policy
5.	Organization's objectives and culture	The organization‘s objective and culture aligned with SSD implementation.	Existence of a learning and development culture
			Existence of a participative decision making culture
			Existence of a support and collaboration culture
			Existence of a power sharing culture
			Existence of tolerance for conflicts and risk culture
6.	Developer	The attitude, motivation and skills of a developer to implement SSD practices.	Existence of communication skills
			Existence of IT management skills
			Existence of planning skills
			Existence of technical skills
			Existence of SSD experience
			Existence of controlling skills
7.	Top Management	Willingly providing support from Management to the other team	Top management considers information security an important organizational priority
			Senior management gives strong and consistent support to the security program
			Senior management is always involved in key information security activities.
			Management ensures that appropriate individuals are made responsible for specific aspects of information security
			Management ensures that everyone who takes information security actions, and makes information security decisions and are held accountable for their decisions and actions

No.	Factor	Description	Assessment Criteria
8.	Security Experts	A group experts who can provide consultation advice on software security	Existence of sufficient security experts
			Existence of communication skills
			Existence of IT management skills
			Existence of planning skills
			Existence of technical skills
			Existence of SSD experience
			Existence of controlling skills
9.	Project Manager	A competent individual heading the project and ensure SSD practices are implemented throughout the software development lifecycle	Existence of communication skills
			Existence of IT management skills
			Existence of planning skills
			Existence of technical skills
			Existence of SSD experience
			Existence of controlling skills
			Existence of communication skills
10.	Automated tool support	Automated security tools that can facilitate secure software development.	Existence of tools to support secure software development (e.g static analyzer, penetration testing tools)
			Existence of policy on using automated secure software development tools
			Developers are trained to use the tool
			Existence of complete technical documentation for the tools
			Tools are compatible with development environment
11.	Cost	Adequate budget is allocated to implement SSD practices	Sufficient budget is allocated for SSD implementation
12.	Project Team	A dedicated team with members from various functions and expertise working to develop a common software	Existence of the both business and technical knowledge into the project team
			Existence of a balanced, cooperative, cross functional and full time project team
			The degree to which project team performance is fairly compensated
			Existence of the empowered project team members
			The degree to which project team have prior experience in large IT projects.
13.	Independent Security Audit Team	A team of security experts who are able to verify and validate the security aspects of the system before production.	Existence of security audit team
			Existence of well-defined audit procedures and has gained management's approval
			Audit policies and procedure are clearly understood by audit team
14.	Segregation of role	Each team member to be given specific roles	Team member's role and responsibilities are clearly defined and documented (eg. Requirement specifier, designer, tester)
			Team members understand their role and responsibilities in a particular project
15.	Team size	Team size is relevant to the size of the project	Sufficient number of team members has been allocated for the project
16.	Team Collaboration	Working together and the basis for bringing together the knowledge, experience and skills of team members.	Existence of development team and security team in the organization
			Existence of a channel where development team and security team communicates with each other (via meetings, forums or other communication channels)
17.	Development Time	Adequate development time to allow implementation of secure development practices	Adequate development time is allocated for SSD implementation

No.	Factor	Description	Assessment Criteria
18.	Software development methodology	A standard method that provides an element of control over the sequence of development activities	Existence of a standard development methodology (e.g agile, Rapid prototyping, waterfall)
			All team members are aware of the activities involved in the methodology
19.	Security Reference Guide	Secure software development reference guides to facilitate developers	Existence of a reference guide related to SSD implementation
			Reference guides are clear and easy to be understood
			Reference guides are easily accessible to developers
20.	Internal Metrics and KPI	Establishment of internal metrics and key performance indicators that can be used to determine the progress and success of the organization's security evolution	Internal security metrics and KPI are documented
			Internal security metrics and KPI are communicated well among the project team members
			Implementation of internal security metrics and KPI are frequently monitored by project manager

HOW TO USE THIS MODEL

STEP 1

The criteria below will assist the user to identify the influential factors that are fulfilled by the organization. Please rate your level of agreement with the following statements pertaining to your project and organization. *(Strongly Disagree (1), Disagree (2), Neutral (3), Agree (4), Strongly Agree (5))*

Factors	Assessment Criteria	How strongly do you agree that this criteria is fulfilled by your organization? *(□ tick only one)*
Change Management	Existence of Change Management Team	□ 1 □ 2 □ 3 □ 4 □ 5
	Change management strategies are well communicated with stakeholders.	□ 1 □ 2 □ 3 □ 4 □ 5
Policy Enforcement	SSD practices and procedures are continually monitored to ensure compliance with security policy	□ 1 □ 2 □ 3 □ 4 □ 5
	SSD practices and procedures are externally audited	□ 1 □ 2 □ 3 □ 4 □ 5
	SSD violations are reported to the proper authority	□ 1 □ 2 □ 3 □ 4 □ 5
Security Training and Awareness	Adequate SSD security training given to all developers	□ 1 □ 2 □ 3 □ 4 □ 5
	SSD policy is communicated well	□ 1 □ 2 □ 3 □ 4 □ 5
	Developers are educated or trained about new security policies	□ 1 □ 2 □ 3 □ 4 □ 5
	Developers aware of my information security roles and responsibilities	□ 1 □ 2 □ 3 □ 4 □ 5
	Top management and developers are aware of the risk of not following the SSD policy	□ 1 □ 2 □ 3 □ 4 □ 5
	Developers are familiar with the SSD policy	□ 1 □ 2 □ 3 □ 4 □ 5
	Developers aware of the procedures for reporting security policy violation	□ 1 □ 2 □ 3 □ 4 □ 5
Reward and Incentives	Existence of reward policy	□ 1 □ 2 □ 3 □ 4 □ 5
	Developers are aware of the reward policy	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of a learning and development culture	□ 1 □ 2 □ 3 □ 4 □ 5

Factors	Assessment Criteria	How strongly do you agree that this criteria is fulfilled by your organization? *(□ tick only one)*
Organization's objectives and culture	Existence of a participative decision making culture	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of a support and collaboration culture	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of a power sharing culture	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of tolerance for conflicts and risk culture	□ 1 □ 2 □ 3 □ 4 □ 5
Developer	Existence of communication skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of IT management skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of planning skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of technical skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of SSD experience	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of controlling skills	□ 1 □ 2 □ 3 □ 4 □ 5
Top Management	The degree to which functional managers willingly assign resources to the SSD implementation as they are needed	□ 1 □ 2 □ 3 □ 4 □ 5
	The degree to which the need for long-term SSD support resources is recognized by management	□ 1 □ 2 □ 3 □ 4 □ 5
	The degree to which executive management is enthusiastic about the possibilities of SSD	□ 1 □ 2 □ 3 □ 4 □ 5
	The degree to which all levels of management support the overall goals of the SSD	□ 1 □ 2 □ 3 □ 4 □ 5
Security Experts	Existence of sufficient security experts	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of communication skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of IT management skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of planning skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of technical skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of SSD experience	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of controlling skills	□ 1 □ 2 □ 3 □ 4 □ 5
Project Manager	Existence of communication skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of IT management skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of planning skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of technical skills	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of SSD experience	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of controlling skills	□ 1 □ 2 □ 3 □ 4 □ 5
Automated tool support	Existence of tools to support secure software development (e.g static analyzer, penetration testing tools)	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of policy on using automated secure software development tools	□ 1 □ 2 □ 3 □ 4 □ 5
	Developers are trained to use the tool	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of complete technical documentation for the tools	□ 1 □ 2 □ 3 □ 4 □ 5
	Tools are compatible with development environment	□ 1 □ 2 □ 3 □ 4 □ 5
	Tools are easy to use	□ 1 □ 2 □ 3 □ 4 □ 5
Cost	Sufficient budget is allocated for SSD implementation	□ 1 □ 2 □ 3 □ 4 □ 5
Project Team	Existence of the both business and technical knowledge into the project team	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of a balanced, cooperative, cross functional and full time project team	□ 1 □ 2 □ 3 □ 4 □ 5
	The degree to which project team performance is fairly compensated	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of the empowered project team members	□ 1 □ 2 □ 3 □ 4 □ 5
	The degree to which project team have prior experience in large IT projects.	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of security audit team	□ 1 □ 2 □ 3 □ 4 □ 5

Factors	Assessment Criteria	How strongly do you agree that this criteria is fulfilled by your organization? *(□ tick only one)*
Security Audit Team	Existence of well-defined audit procedures and has gained management's approval	□ 1 □ 2 □ 3 □ 4 □ 5
	Audit policies and procedure are clearly understood by audit team	□ 1 □ 2 □ 3 □ 4 □ 5
Segregation of role	Team member's role and responsibilities are clearly defined and documented (eg. Requirement specifier, designer, tester)	□ 1 □ 2 □ 3 □ 4 □ 5
	Team members understand their role and responsibilities in a particular project	□ 1 □ 2 □ 3 □ 4 □ 5
Team size	Sufficient number of team members has been allocated for the project	□ 1 □ 2 □ 3 □ 4 □ 5
Team Collaboration	Existence of development team and security team in the organization	□ 1 □ 2 □ 3 □ 4 □ 5
	Existence of a channel where development team and security team communicates with each other (via meetings, forums or other communication channels)	□ 1 □ 2 □ 3 □ 4 □ 5
Development Time	Adequate development time is allocated for SSD implementation	□ 1 □ 2 □ 3 □ 4 □ 5
Software development methodology	Existence of a standard development methodology (e.g agile, Rapid prototyping, waterfall)	□ 1 □ 2 □ 3 □ 4 □ 5
	All team members are aware of the activities involved in the methodology	□ 1 □ 2 □ 3 □ 4 □ 5
Security Reference Guide	Existence of a reference guide related to SSD implementation	□ 1 □ 2 □ 3 □ 4 □ 5
	Reference guides are clear and easy to be understood	□ 1 □ 2 □ 3 □ 4 □ 5
	Reference guides are easily accessible to developers	□ 1 □ 2 □ 3 □ 4 □ 5
Internal Security Metrics and KPI	Internal security metrics and KPI are documented	□ 1 □ 2 □ 3 □ 4 □ 5
	Internal security metrics and KPI are communicated well among the project team members	□ 1 □ 2 □ 3 □ 4 □ 5
	Implementation of internal security metrics and KPI are frequently monitored by project manager	□ 1 □ 2 □ 3 □ 4 □ 5

STEP 2

The value for fulfilment of factor is calculated using simple average method. For example, CM1 and CM2 are criteria to assess the fulfilment of "Change Management" factor. For example, the participant states "Agree (3)" for CM1 and "Disagree (1)" for CM2. The value for fulfilment of Change Management factor will be 2 ((3 + 1)/2). For the purpose of this study, only factors with values 3 and above are included to further select potential secure software development practice from the **Table A: List of secure software development practices influenced by each factor** as shown below.

Table A: List of secure software development practices influenced by each factor

Factors	Practice ID	Secure software development practices
Institutional Context		
Change Management	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
Policy Enforcement	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P3	Specify operational environment
	P7	Document security-relevant requirements
	P16	Implement interface contracts
	P21	Verify security attributes of resources
	P22	Perform code signing
	P23	Build operational security guide
	P24	Manage security issue disclosure process
Security Training and Awareness	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P7	Document security-relevant requirements
	P11	Research and assess security posture of technology solutions
	P12	Annotate class designs with security properties
	P23	Build operational security guide
Rewards and Incentives	P1	Institute Security Awareness Program
	P6	Identify user roles and resource capabilities
Organization's objectives and culture	P1	Institute Security Awareness Program
	P4	Identify global security policy
	P24	Manage security issue disclosure process
People and Action		
Developer	P1	Institute Security Awareness Program
	P6	Identify user roles and resource capabilities
	P7	Document security-relevant requirements
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P10	Apply security principles to design
	P11	Research and assess security posture of technology solutions
	P12	Annotate class designs with security properties
	P13	Specify database security configuration
	P14	Perform security analysis of system requirements and design (threat modeling)
	P15	Integrate security analysis into source code management process
	P16	Implement interface contracts
	P17	Implement and elaborate resource policies and security technologies
	P18	Address reported security issues
	P19	Perform source-level security review
	P20	Identify, implement and perform security tests
	P21	Verify security attributes of resources
	P22	Perform code signing
Top Management	P1	Institute Security Awareness Program

Factors	Practice ID	Secure software development practices
	P4	Identify global security policy
	P7	Document security-relevant requirements
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P16	Implement interface contracts
	P17	Implement and elaborate resource policies and security technologies
	P22	Perform code signing
	P23	Build operational security guide
	P24	Manage security issue disclosure process
Security Experts	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P3	Specify operational environment
	P4	Identify global security policy
	P5	Identify resources and trust boundaries
	P6	Identify user roles and resource capabilities
	P7	Document security-relevant requirements
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P10	Apply security principles to design
	P11	Research and assess security posture of technology solutions
	P12	Annotate class designs with security properties
	P13	Specify database security configuration
	P14	Perform security analysis of system requirements and design (threat modeling)
	P15	Integrate security analysis into source code management process
	P16	Implement interface contracts
	P17	Implement and elaborate resource policies and security technologies
	P18	Address reported security issues
	P19	Perform source-level security review
	P20	Identify, implement and perform security tests
	P21	Verify security attributes of resources
	P22	Perform code signing
	P23	Build operational security guide
	P24	Manage security issue disclosure process
Project Manager	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P3	Specify operational environment
	P4	Identify global security policy
	P5	Identify resources and trust boundaries
	P6	Identify user roles and resource capabilities
	P7	Document security-relevant requirements

Factors	Practice ID	Secure software development practices
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P10	Apply security principles to design
	P13	Specify database security configuration
	P16	Implement interface contracts
	P17	Implement and elaborate resource policies and security technologies
	P20	Identify, implement and perform security tests
	P21	Verify security attributes of resources
	P22	Perform code signing
	P23	Build operational security guide
	P24	Manage security issue disclosure process
Project Content		
Automated tool support	P2	Monitoring Security Metrics
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P10	Apply security principles to design
	P11	Research and assess security posture of technology solutions
	P14	Perform security analysis of system requirements and design (threat modeling)
	P18	Address reported security issues
	P19	Perform source-level security review
Cost	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P11	Research and assess security posture of technology solutions
	P14	Perform security analysis of system requirements and design (threat modeling)
	P18	Address reported security issues
Project Team	P1	Institute Security Awareness Program
	P4	Identify global security policy
	P5	Identify resources and trust boundaries
	P6	Identify user roles and resource capabilities
	P7	Document security-relevant requirements
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P11	Research and assess security posture of technology solutions
	P12	Annotate class designs with security properties
	P13	Specify database security configuration
	P15	Integrate security analysis into source code management process
	P16	Implement interface contracts

Factors	**Practice ID**	**Secure software development practices**
	P20	Identify, implement and perform security tests
	P22	Perform code signing
	P24	Manage security issue disclosure process
Security Audit Team	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P4	Identify global security policy
	P5	Identify resources and trust boundaries
	P6	Identify user roles and resource capabilities
	P7	Document security-relevant requirements
	P11	Research and assess security posture of technology solutions
	P17	Implement and elaborate resource policies and security technologies
	P18	Address reported security issues
	P19	Perform source-level security review
	P20	Identify, implement and perform security tests
	P21	Verify security attributes of resources
	P24	Manage security issue disclosure process
Segregation of role	P6	Identify user roles and resource capabilities
Team size	P10	Apply security principles to design
	P11	Research and assess security posture of technology solutions
	P12	Annotate class designs with security properties
	P13	Specify database security configuration
	P19	Perform source-level security review
	P20	Identify, implement and perform security tests
	P24	Manage security issue disclosure process
Team Collaboration	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P3	Specify operational environment
	P4	Identify global security policy
	P5	Identify resources and trust boundaries
	P6	Identify user roles and resource capabilities
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P11	Research and assess security posture of technology solutions
	P13	Specify database security configuration
	P17	Implement and elaborate resource policies and security technologies
	P18	Address reported security issues
	P20	Identify, implement and perform security tests
	P22	Perform code signing
	P23	Build operational security guide
Development Time	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P11	Research and assess security posture of technology solutions

Factors	Practice ID	Secure software development practices
	P12	Annotate class designs with security properties
	P13	Specify database security configuration
	P14	Perform security analysis of system requirements and design (threat modeling)
	P18	Address reported security issues
	P19	Perform source-level security review
	P20	Identify, implement and perform security tests
System Development Processes		
Security Documentation	P1	Institute Security Awareness Program
	P2	Monitoring Security Metrics
	P3	Specify operational environment
	P6	Identify user roles and resource capabilities
	P7	Document security-relevant requirements
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.
	P10	Apply security principles to design
	P11	Research and assess security posture of technology solutions
	P12	Annotate class designs with security properties
	P13	Specify database security configuration
	P14	Perform security analysis of system requirements and design (threat modeling)
	P15	Integrate security analysis into source code management process
	P16	Implement interface contracts
	P17	Implement and elaborate resource policies and security technologies
	P19	Perform source-level security review
	P21	Verify security attributes of resources
	P22	Perform code signing
	P23	Build operational security guide
	P24	Manage security issue disclosure process
Software development methodology	P6	Identify user roles and resource capabilities
	P21	Verify security attributes of resources
Internal Metrics and KPI	P2	Monitoring Security Metrics
	P4	Identify global security policy
	P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.
	P21	Verify security attributes of resources
	P23	Build operational security guide

Appendix G Model Evaluation Questionnaire

SECURE SOFTWARE DEVELOPMENT PRACTICE SELECTION MODEL FOR MALAYSIAN PUBLIC SERVICE ORGANIZATION

EVALUATION QUESTIONNAIRE

Description

Secure Software Development Practice Selection Model for Malaysian Public Service Organization consist of three elements: Influential factors, Assessment Criteria and Secure Software Development Practices. All these elements have been agreed and validated by the software development experts from Malaysian Public Service Organization. The main purpose of SSDPAM is to assist Project Managers to identify secure software development practices based on the factors achieved. Secure software development is defined as "the set of activities performed to develop, maintain, and deliver a secure software solution".

Instruction

Please answer all the questions in this survey. The objective of this survey is to identify influential factors that have been achieved by the software project through a set of assessment indicators. The results from this survey will be used in to determine the Secure Software Development Practices that could be adopted for the project.

Prepared by :

Sri Lakshmi Kanniah
Advanced Informatics School (AIS)
Universiti Teknologi Malaysia (UTM), KL
Email : kanniah.srilakshmi@gmail.com
Tel. No. : 016-2029444

Supervisor :

Dr. Mohd Naz'ri Mahrin
Advanced Informatics School (AIS)
Universiti Teknologi Malaysia (UTM), KL
Email: mdnazrim@utm.my

RESPONDENT'S PROFILE

Name ________________________________

Designation: ________________________________

Role: - - - - - -

Project Title: ________________________________

Name of Organization: ________________________________

Project Description:

Instructions: Which of these Secure Software Development Practices are currently being implemented in the software development project. Please choose Yes/No for each practice

Item	Practice	Activities	Yes/No
P1	Institute Security Awareness Program	• Provide security training to all team members. • Distribute and present security requirements to all team members before development. • Project managers must assess to see whether the developers are following the security guidelines given from time to time. • Appoint a project security officer for each individual project. • Reward developers for following security guidelines consistently over a period of time.	
P2	Monitoring Security Metrics	• Identify all security metrics that can be used to determine security posture of the software at the beginning of the project. • Monitor the usage of the metric to evaluate the effectiveness of the metric. • Strategize data collection and produce output report in appropriate format for the team. • Periodically collect and evaluate metrics.	
P3	Specify operational environment	• Identify requirements and assumptions related to operating system and its components that could have security impact on the software. • Identify requirements and assumptions related to network architecture and resources such as databases and bandwidth that could have security impact on the software	
P4	Identify global security policy	• Build a global project security policy, if necessary • Determine suitability of global requirements to project	
P5	Identify resources and trust boundaries	• Describe the architecture of the system from the perspective of the network • Identify data resources such as databases and Access Control List(ACL)	
P6	Identify user roles and resource capabilities	• Identify capabilities (read, write, execute, create, and delete) for files and databases used in the project. • Map system roles (e.g. administrator, users and guest) to capabilities • Identify the attacker profile (insiders, “Script Kiddies”, Competitors, Government, Activist) what they want to gain	
P7	Document security-relevant requirements	• Document clear business requirements • Develop functional security requirements showing how the basic security services are addressed for each resource in the project • Specify all third party components required in the project • Specify mechanisms to address potential security risk for each resource • Resolve deficiencies and conflicts between business, functional and global requirements	

P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.	• Identify misuse cases for each actor present in the system • Describe and document misuse cases • Identify defense mechanisms for misuse cases • Review and discuss the misuse case with stakeholders, so that they have a clear • understanding of the misuse case	
P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.	• Define the specific mechanisms through which anyone could interact with the application regardless of their role in the system • Identify all roles that could possibly access the defined entry point. • For each entry point, document the resources that should be accessible from that entry point	
P10	Apply security principles to design	• Refine existing application security in the system design • Identify solutions for meeting security requirements at each identified point in • the design • Build hardened protocol specifications such as SSL/TLS • Design hardened API interfaces	
P11	Research and assess security posture of technology solutions	• Get structured technology assessment from vendor before integrating into the system • Perform security risk assessment on vendor products • Receive permission to perform security testing of vendor products • Perform security testing on vendor products	
P12	Annotate class designs with security properties	• Map each data element back to the requirements to determine the • requirements on that data • For each data field define the owning role or roles and which role or roles have access to which basic capabilities throughout the lifetime of the data • Annotate methods to identify which operations they perform on data	
P13	Specify database security configuration	• Identify candidate database configuration • Validate that the baseline configuration properly addresses the security requirements for that database.	
P14	Perform security analysis of system requirements and design (threat modelling)	• Before performing a security analysis, review all existing high-level system documentation such as user manuals and architectural documentation • Review non-security requirements • Assess completeness of security requirements • Identify threats on assets/capabilities • Determine level of risk • For each identified risk, identify any feasible approaches for mitigating the risk and evaluate their cost and effectiveness • Evaluate findings, determine whether the assessments are actually correct to the business and make risk-based decisions based on this information.	
P15	Integrate security analysis into source code management process	• Select dynamic or static analysis tools to be integrated into development process • Determine analysis integration point (check-in process, as part of the build process, or independently) • Integrate analysis technology	

P16	Implement interface contracts	• Implement validation and error handling on each function or method inputs • Implement validation on each function or method outputs	
P17	Implement and elaborate resource policies and security technologies	• The developer should identify any remaining ambiguities in the specification of security properties or technologies • The implementer should ensure that all coding guidelines are met — especially security guidelines	
P18	Address reported security issues	• When a security risk/issue is identified in a system, further investigation should be assigned to the appropriate designer / architect • Assess likely exposure and impact of the issue/risk • Determine and execute short term or long-term remediation strategies • Perform testing to ensure that the issue/risk was properly addressed	
P19	Perform source-level security review	• Scope out the areas that merit the most attention before performing source-level security review • Run automated analysis tools • Evaluate each potential risk identified by the tool • Identify additional risks by reviewing both those risks identified in the architectural analysis and a database of common risks.	
P20	Identify, implement and perform security tests	• Identify security tests for individual requirements • Identify tests that will determine which system roles can gain access to each resource • Using a common testing checklist, determine what other security tests are appropriate to the system • Implement test plan • Perform the identified security tests as specified in the test plan	
P21	Verify security attributes of resources	• Check whether permissions granted by the system's default install exactly • match those put forth by the resource specifier in the security requirements • Specify in the requirements, a security profile or operational security guide what resources the system should be able to access	
P22	Perform code signing	• Obtain code signing credentials(e.g. PKI, CA) • Identify signing targets such as a single archive file (JAR, WAR, or CAB) • Sign identified targets	
P23	Build operational security guide	• Document pre-install configuration requirements • Document application activity including network ports, files on the file system, registry resources, database resources • Document the security architecture including authentication mechanisms, • default policies for authentication and other functions, and any security • protocols that are mandatory or optional • Document security configuration mechanisms • Document significant risks and known compensating controls	

P24	Manage security issue disclosure process	• A central security response alias should be established, such as security@ or • secalert@ and published on the web site if possible • On receipt of the vulnerability disclosure, respond with acknowledgement of • receipt, as well as a reasonable timetable for addressing the vulnerability. • The reported vulnerability should be entered into the process for dealing with • reported security issues • Communicate relevant information to the researcher • Provide a security advisory and customer access to remediation	

Please rate your level of agreement with the following statements pertaining to your project and organization.		
(Strongly Disagree(0), Disagree (1), Neutral (2), Agree (3), Strongly Agree(4))		
INSTITUTIONAL CONTEXT		
No.	**Question**	**Answer**
CM1	A formal management team exist in my organization.	
CM2	Change management strategies are well communicated with stakeholders.	
PE1.	SSD practices and procedures are continually monitored to ensure compliance with security policy	
PE2.	SSD practices and procedures are externally audited	
PE3.	SSD violations are reported to the proper authority	
TA1.	Adequate SSD security training given to all developers	
TA2.	SSD policy is communicated well	
TA3.	Developers are educated or trained about new security policies	
TA4.	Developers aware of my information security roles and responsibilities	
TA5.	Top management and developers are aware of the risk of not following the SSD policy	
TA6.	Developers are familiar with the SSD policy	
TA7.	Developers aware of the procedures for reporting security policy violation	
RI1.	Existence of reward policy	
RI2.	Developers are aware of the reward policy	
OC1.	Existence of a learning and development culture	
OC2.	Existence of a participative decision making culture	
OC3.	Existence of a support and collaboration culture	
OC4.	Existence of a power sharing culture	
0C5.	Existence of tolerance for conflicts and risk culture	

PEOPLE AND ACTION		
No.	**Question**	Answer
D1.	Developer has fair level of communication skills.	
D2.	Developer has fair level IT management skills	
D3.	Developer has good planning skills.	

D4.	Developer has sound of technical skills	
D5.	Developer has experience in secure software development	
D6.	Developer has controlling skills.	
TM1.	Functional managers are willing to assign resources to the SSD implementation as they are needed	
TM2.	The need for long-term SSD support resources is recognized by management.	
TM3.	Executive management is enthusiastic about the possibilities of SSD	
TM4.	All levels of management support the overall goals of the SSD	
SE1.	We have sufficient number of security experts.	
SE2.	The security expert has good communication skills	
SE3.	The security expert has IT management skills	
SE4.	The security expert has good planning skills	
SE5.	The security expert has good technical skills	
SE6.	The security expert has experience in secure software development	
SE7.	The security expert has controlling skills	
PM1.	Project Manager has good communication skills	
PM2.	Project Manager has IT management skills	
PM3.	Project Manager has good planning skills	
PM4.	Project Manager has good technical skills	
PM5.	Project Manager has experience in secure software development	
PM6.	Project Manager has controlling skills	

PROJECT CONTENT		
No.	**Question**	Answer
AT1.	We use tools to support secure software development (e.g static analyzer, penetration testing tools)	
AT2	A formal policy exist on using automated secure software development tools	
AT3.	Developers are trained to use the tool	
AT4.	A complete technical documentation for the tools exist.	
AT5.	Tools are compatible with development environment	
AT6.	Tools are easy to use	
C1.	Sufficient budget is allocated for SSD implementation	
PT1.	Both business and technical knowledge exist in the project team	
PT2.	A balanced, cooperative, cross functional and full time project team exist within the project	
PT3.	The degree to which project team performance is fairly compensated	
PT4.	Existence of the empowered project team members	
PT5.	The degree to which project team have prior experience in large IT projects.	
SAT1.	Existence of security audit team	
SAT2.	Existence of well-defined audit procedures and has gained management's approval	

SAT3.	Audit policies and procedure are clearly understood by audit team	
SR1.	Team member’s role and responsibilities are clearly defined and documented (eg. Requirement specifier, designer, tester)	
SR2.	Team members understand their role and responsibilities in a particular project	
TS1.	Sufficient number of team members has been allocated for the project	
TC1.	Existence of development team and security team in the organization	
TC2.	Existence of a channel where development team and security team communicates with each other (via meetings, forums or other communication channels)	
DT1.	Adequate development time is allocated for SSD implementation	

SYSTEM DEVELOPMENT PROCESS		
No.	**Question**	Answer
SDM1.	We use a standard development methodology (e.g agile, Rapid prototyping, waterfall)	
SDM2.	All team members are aware of the activities involved in the methodology	
DOC1.	Reference guides related to SSD implementation exist	
DOC2.	Reference guides are clear and easy to be understood	
DOC3.	Reference guides are easily accessible to developers	
MK1.	Internal security metrics and KPI are documented	
MK2.	Internal security metrics and KPI are communicated well among the project team members	
MK3.	Implementation of internal security metrics and KPI are frequently monitored by project manager	

Appendix H List of factors Based on Secure Software Development Practices

Id	Secure Software Development Practice	Factors Influencing The Practice
P1	Institute Security Awareness Program	Change Management Policy Enforcement Security Training and Awareness Rewards and Incentives Organization's objectives and culture Developer Top Management Security Experts Project Manager Cost Project Team Security Audit Team Team Collaboration Development Time Security Documentation
P2	Monitoring Security Metrics	Change Management Policy Enforcement Security Training and Awareness Security Experts Project Manager Automated tool support Cost Security Audit Team Team Collaboration Development Time Security Documentation Internal Metrics and KPI
P3	Specify operational environment	Policy Enforcement Security Experts Project Manager Team Collaboration Security Documentation
P4	Identify global security policy	Organization's objectives and culture Top Management Security Experts Project Manager Project Team Security Audit Team Team Collaboration Internal Metrics and KPI

Id	Secure Software Development Practice	Factors Influencing The Practice
P5	Identify resources and trust boundaries	Security Experts Project Manager Project Team Security Audit Team Team Collaboration
P6	Identify user roles and resource capabilities	Rewards and Incentives Developer Security Experts Project Manager Project Team Security Audit Team Segregation of role Team Collaboration Security Documentation Software development methodology
P7	Document security-relevant requirements	Policy Enforcement Security Training and Awareness Developer Top Management Security Experts Project Manager Project Team Security Audit Team Security Documentation
P8	Detail misuse cases. Misuse cases are identical to use cases, except that they are meant to detail common attempted abuses of the system.	Top Management Security Experts Project Manager Automated tool support Cost Development Time Security Documentation Internal Metrics and KPI
P9	Identify attack surface. The system attack surface is the collection of possible entry points for an attacker.	Developer Security Experts Project Manager Automated tool support Project Team Team Collaboration Security Documentation
P10	Apply security principles to design	Developer Security Experts Project Manager Automated tool support Team size Security Documentation

Id	Secure Software Development Practice	Factors Influencing The Practice
P11	Research and assess security posture of technology solutions	Security Training and Awareness Developer Security Experts Automated tool support Cost Project Team Security Audit Team Team size Team Collaboration Development Time Security Documentation
P12	Annotate class designs with security properties	Security Training and Awareness Developer Security Experts Project Team Team size Development Time Security Documentation
P13	Specify database security configuration	Developer Security Experts Project Manager Project Team Team size Team Collaboration Development Time Security Documentation
P14	Perform security analysis of system requirements and design (threat modeling)	Developer Security Experts Automated tool support Cost Development Time Security Documentation
P15	Integrate security analysis into source code management process	Developer Security Experts Project Team Security Documentation
P16	Implement interface contracts	Policy Enforcement Developer Top Management Security Experts Project Manager Project Team Security Documentation
P17	Implement and elaborate resource policies and security technologies	Developer Top Management Security Experts Project Manager Security Audit Team Team Collaboration Security Documentation

Id	Secure Software Development Practice	Factors Influencing The Practice
P18	Address reported security issues	Developer Security Experts Automated tool support Cost Security Audit Team Team Collaboration Development Time
P19	Perform source-level security review	Developer Security Experts Automated tool support Security Audit Team Team size Development Time Security Documentation
P20	Identify, implement and perform security tests	Developer Security Experts Project Manager Project Team Security Audit Team Team size Team Collaboration Development Time
P21	Verify security attributes of resources	Policy Enforcement Developer Security Experts Project Manager Security Audit Team Security Documentation Software development methodology Internal Metrics and KPI
P22	Perform code signing	Policy Enforcement Developer Top Management Security Experts Project Manager Project Team Team Collaboration Security Documentation
P23	Build operational security guide	Policy Enforcement Security Training and Awareness Top Management Security Experts Project Manager Team Collaboration Security Documentation Internal Metrics and KPI

Id	Secure Software Development Practice	Factors Influencing The Practice
P24	Manage security issue disclosure process	Policy Enforcement Organization's objectives and culture Top Management Security Experts Project Manager Project Team Security Audit Team Team size Security Documentation

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.