Secure Software Development Practice Adoption Model: A Delphi Study

Sri Lakshmi Kanniah and Mohd Naz'ri bin Mahrin
Advanced Informatics School, Universiti Teknologi Malaysia,
Kuala Lumpur, MALAYSIA.
lksri2@live.utm.my

Abstract— **Developing secure software is a major concern in public service organizations as highly-sensitive and confidential data are transacted through online applications. A great number of departments around the public sectors depend on online services to ensure effective services delivery. The insecure software can lead to loss of revenue and damage to business reputation. Implementation of secure development practices throughout the software development lifecycle is influenced by many various factors such as organizational and people factor. Although numerous methods, models and standards in regards to secure software development has been established, implementation of the whole model is quite challenging as it involves cost, skill and time. On that account, this paper presents the results of the Delphi study conducted at the Malaysian Public Service Organization (MPS) with the aim to identify the factors which affect the implementation of secure software development practices. Identified factors are mapped to the security practices in order to establish a relationship between the factors and security practices. In the efforts to achieve this objective, 10 experts who were involved in software development from Malaysian Public Service Organization participated in the study.**

Index Terms—**Delphi; Secure Software Development; Software Development; Software Security.**

I. Introduction

The rapid growths of internet and e-commerce have instilled revolutionary changes in peoples' lifestyle and living standards. An apparent example of this phenomenon can be witnessed through the fact that almost all business organizations convert the way they run their daily operations and marketing activities from manual to the use of websites. In the same way, the government has been trying to deliver their services effectively and efficiently to meet the needs of citizens, employees, and businesses through electronic means. In accordance with that, E-Government (EG) was initiated to provide online government services delivery to the public users and prompted many government organizations to execute its implementation. Similarly, the Malaysian Public Sector targeted zero face-to-face services delivery with 90% of government services made available online by 2015 [1]. As more services go online, security became the biggest challenge thus increasing the importance of safeguarding the web application from internal and external threats.

This paper reports on the results of an exploratory study conducted at the Malaysian Public Service Organization (MPS) to identify the factors influencing the implementation of secure software development practices. Before addressing the security vulnerability issues that are present in their software, it is important to understand the factors that can influence the organization to implement secure development practices. Several researches have been conducted on secure software development issues [2-4]; however, none of them focused primarily on public service organizations.

II. Related Work

Research in security covers a varied range of approaches and processes that deal with security during software development. Several actions have been suggested in order to incorporate security in the software development life cycle (SDLC) by using different software models. Several modifications have been made to traditional lifecycle by inserting security activities into traditional lifecycle for the purpose of creating security enhanced methodologies and processes [5].

Researchers at University College London have developed 'Appropriate and Effective Guidance in Information Security' (AEGIS), a research model that has integrated security and usability using a spiral model, based on UML. This model defines a UML meta-model of the definition and the rational over the system's assets [6]. AEGIS guides developers to deal with security and usability requirements in system design. The UML meta-model defined by authors identifies assets, the context of operation and supporting the modeling of security requirements. All security decisions in AEGIS are derived from knowledge of assets of the system. Core security activities for system design sessions in AEGIS are: Identification of assets and security requirements, analysis of risk and secure design, and identification of the risks, vulnerabilities, and threats to the system. The output from these activities is documented in a design document which consists of the system architecture with all specified countermeasures. In AEGIS, security expertise is absent in the development process. Moreover, decision making in the selection of security countermeasures is done by stakeholders. The author's rationale behind this is that decision-makers are "better suited to deal with the enforcement of the social requirements of security" while developers are "necessary for the technical implementation of security.

Secure Software Development Model (SSDM) which was developed at the Nigerian University of Agriculture [7] integrates security activities into engineering process, which are: Security training, threat modeling, security specification, review of security specification, and penetration testing. Furthermore, SSDM has separated security specification from functional specification.

The security process 'Comprehensive, Lightweight Application Security Process' — CLASP [8] introduces a lightweight process for SSD. CLASP provides structured practices for deriving security requirements of software systems [9]. CLASP outlines seven key best practices, such as Security awareness, application evaluations, derivation of security requirements, implementation of secure development practices, developing vulnerability remediation measures, defining and monitoring metrics, and publishing operational guidelines. CLASP also specifies a set of activities that should be incorporated in the development lifecycle. CLASP provides roles and security to structure and supports the activities in the resources methodology.

'The Microsoft's Security Development Lifecycle (SDL) has incorporated security activities into each development phase of SDLC [10]. Its purpose is to reduce the number of vulnerabilities in software [10]. SDL consists of a set of activities that overcome security issues. The activities in SDL are grouped in phases, which can be mapped to general software development phases.

Seven 'touchpoints' exhibit how software developers can implement them in the development stages. The aim of 'touchpoints' is to increase effectiveness through: code review, architectural risk analysis, penetration testing, risk-based security tests, abuse cases, security requirements, and security operations [9].

Conclusively, the aforementioned models focus on what is needed to build secure software. However, there is a lack of research on identifying the factors required for successful implementation of the SSD process.

III. METHODOLOGY

A Delphi survey technique was conducted with eight (8) experts to determine the SSD implementation factor, assessment indicators for each and the relationship between factors and practices. This information facilitates the development team in identifying SSD practices for each software development project based on the achievement of factors.

A. *Delphi Method*

The Delphi method was found to be advantageous: (1) to explore or expose underlying assumptions or information leading to different judgments and to seek out information which may generate a consensus on the part of the respondent group [5]. [6] identified two major areas for application of the Delphi technique are the traditional forecasting and more recently concept/framework development where studies typically involve a two-step process being: (1) identifying and elaborating a set of concepts and (2) classification/taxonomy development. A more comprehensive view of experts in software development was required to identify the factors and assessment criteria. Expert input is also needed in identifying the dependence of secure software development practices on the factors. This can be achieved by mapping each practice with factors influencing the practice. Views from experts could vary according to their level of knowledge and experiences. Thus, the Delphi method was found to be appropriate in exploring these similarities and differences in opinions from experts. The Delphi method is also suitable to facilitate extensive and effective communication and collaboration by multiple experts in determining the factors, indicators, and practices which are dependent on the factors.

Besides this, the Delphi method encourages sincere opinions from experts without imposing any pressure or conflict that commonly occurs during face-to-face meetings. This improves the validity of the results obtained from this study [7]. Furthermore, the Delphi method is also capable of providing reliable consensus on views among experts, without possible biases during the process [8].

The Delphi method was executed in three phases. Phase 1 and Phase 2 was completed in a single round. However, Phase 3 was completed in two rounds to achieve satisfactory consensus among experts. The objectives of each phase are shown in Table 1.

B. *Experts Selection*

A fundamental aspect of the Delphi Technique is the selection of the expert panel. [9] indicates that this selection will potentially determine the success of a Delphi study. The initial targeted sample size of experts was 10. To obtain the target sample size of experts in this method, purposive sampling was used with a combination of expert sampling and snowball sampling. Expert sampling and snowball sampling are non-probability sampling techniques, whereby with expert sampling, experts were chosen based on a set of predefined criteria in the area of knowledge and expertise aligned with the objectives of the Delphi method, as well as their ability and willingness to contribute to the study [10]. Since the population of experts with experience in software development and/or software security in the Malaysian Public Sector is unknown, and it was difficult to locate the required experts in the population, snowballing sampling was used to penetrate the unknown population. Therefore, the selection of experts was made on a referral basis. A total of 10 experts participated in this Delphi study consisting of two consultants and eight senior ICT practitioners in the public sector who are involved in software development.

Table 1
Three phases of the Delphi study

Phase	Objective
Phase 1 (1 Round)	To determine factors that influence implementation of Secure Software Development practices in the public sector and to suggest new factors, if any
Phase 2 (1 Round)	To determine assessment indicators for each factor that influence implementation of Secure Software Development practices in the public sector and to suggest new indicators, if any
Phase 3 (2 rounds)	To determine Secure Software Development practices which are dependent on each factor

C. *Questionnaire Development for Delphi Study*

The questionnaires for the Delphi phases were designed with the appropriate assessment items to achieve the method's objectives. The questionnaires for all three phases were piloted prior to actual the Delphi study to detect and correct inflexibility in terms of questionnaire design, measurement, and analysis. This also increases the validity and reliability of the results from the Delphi study.

IV. RESULTS

A. Phase 1 Delphi Study: To determine factors that influence implementation of Secure Software Development practices in the public sector and to suggest new factors

Phase 1 Delphi study aimed to determine factors that influence the implementation of Secure Software Development practices in the public sector. Experts were asked to state their level of agreement for each of the factor. For this purpose, a structured questionnaire was completed by each of the selected experts. Factors listed in the questionnaire was derived through Systematic Literature Review [11] and interviews with practitioners from public service organizations. An influential factor was determined using a five-point Likert scale: Strongly Agree (5), Agree (4), Neutral (3), Disagree (2) and Strongly Disagree (1). Suggestions by experts on potential factors were analyzed qualitatively using the Grounded Theory. Using the Grounded Theory technique, suggested factors were analyzed to ensure the factors were new factors and different from those listed as constructs in the questionnaire. Meanwhile, the Quantitative analysis was conducted using SPSS Statistics 20 to determine the mode for the probability of occurrences identified by experts. The modes were used as a measure of consensus among the experts. The consensus is achieved if at least 75% of the total experts agreed on a mode, or the majority of the experts have stated the given mode. If multiple modes were calculated, this shows consensus is not achieved among the experts with regards to an agreement on the factors. The modes for the influential factors after Phase 1 Delphi Study is shown in Table 2. The results clearly indicate that all the experts agree that the stated factors influence secure software development practice implementation. Since consensus was achieved in the First Round, there was no need for a second round.

Table 2
Mode for Level of Agreement on Influential Factors for Secure Software Development Practice Adoption

	Factors	Mode of Agreement
A	Institutional Context	
1.	Change Management	5
2	Policy Enforcement	5
3	Security Training and Awareness	5
4	Reward and Incentives	4
5	Organization's objectives and culture	5
B	People and Action	
6	Developer	5
7	Top Management	4
8	Security Experts	5
9	Project Manager	4
C	Project Content	
10	Automated tool support	5
11	Cost	4
12	Project Team	5
13	Security Audit Team	5
14	Segregation of role	4
15	Team size	5
16	Team Collaboration	5
17	Development Time	5
D	System Development Processes	
18	Security Documentation	4
19	Software development methodology	5
20	Internal Metrics and KPI	5

Table 3
Mean for Level of Agreement on Assessment Indicators

Factor	Assessment Indicators	Mean
	ORGANIZATIONAL CONTEXT	
Change Management (CM)	CM1. Existence of Change Management Team	4.4
	CM2. Change management strategies are well communicated with stakeholders.	4.7
Policy Enforcement (PE)	PE1. SSD practices and procedures are continually monitored to ensure compliance with security policy	4.7
	PE2.SSD practices and procedures are externally audited	4.2
	PE3.SSD violations are reported to the proper authority	4.5
Security Training and Awareness (TA)	TA1.Adequate SSD security training is given to all developers	4.6
	TA2.SSD policy is communicated well	4.1
	TA3.Developers are educated or trained about new security policies	4.6
	TA4.Developers aware of my information security roles and responsibilities	4.5
	TA5.Top management and developers are aware of the risk of not following the SSD policy	4.3
	TA6.Developers are familiar with the SSD policy	4.2
	TA7.Developers aware of the procedures for reporting security policy violation	4.1
Reward and Incentives (RI)	RI1.Existence of reward policy	4.0
	RI2.Developers are aware of the reward policy	4.1
Organization's objectives and culture (OC)	OC1.Existence of a learning and development culture	4.4
	OC2.Existence of a participative decision making culture	4.3
	OC3.Existence of a support and collaboration culture	4.4
	OC4.Existence of a power sharing culture	4.3
	OC5.Existence of tolerance for conflicts and risk culture	4.6
	PEOPLE AND ACTION	
Developer (D)	D1.Existence of communication skills	4.4
	D2.Existence of IT management skills	4.3
	D3.Existence of planning skills	4.4
	D4.Existence of technical skills	4.6
	D5.Existence of SSD experience	4.4
	D6.Existence of controlling skills	4.2
Top Management (TM)	TM1.The degree to which functional managers willingly assign resources to the SSD implementation as they are needed	4.2
	TM2.The degree to which the need for long-term SSD support resources is recognized by management	4.1
	TM3.The degree to which executive management is enthusiastic about the possibilities of SSD	4.1
	TM4.The degree to which all levels of management support the overall goals of the SSD	4.5
Security Experts (SE)	SE1.Existence of sufficient security experts	4.9
	SE2.Existence of communication skills	4.7
	SE3.Existence of IT management skills	4.4
	SE4.Existence of planning skills	4.6
	SE5.Existence of technical skills	5.0
	SE6.Existence of SSD experience	4.8
	SE7.Existence of controlling skills	4.6
Project Manager (PM)	PM1.Existence of communication skills	4.6
	PM2.Existence of IT management skills	4.6
	PM3.Existence of planning skills	4.7
	PM4.Existence of technical skills	4.1
	PM5.Existence of SSD experience	4.2
	PM6.Existence of controlling skills	4.7

B. Phase 2 Delphi Study: To determine assessment indicators for each factor that influence implementation of Secure Software Development practices in public sector and to suggest new indicators, if any

Phase 2 Delphi study aimed to determine assessment indicators for each factor that influence the implementation of Secure Software Development practices in the public sector. Experts were asked to state their level of agreement for each of the assessment indicator. Similar to Phase 1, a structured questionnaire was completed by each of the selected experts. Assessment indicators included in the survey questionnaire was derived from literature [12-14] and interviews with practitioners from public service organizations. A five-point Likert scale: Strongly Agree (5), Agree (4), Neutral (3), Disagree (2) and Strongly Disagree (1) was used to determine the agreement on each of the assessment indicator. The response from the experts was analyzed using quantitative analysis method. Mean was used as a measure of consensus among the experts. Indicators with a mean of 4.0 and above are accepted and used in the next phase. There was no need for an additional round since consensus was achieved in the First Round. The result of the Phase 2 study (for Organizational Context and People and Action) is shown in Table 3.

Table 4
List of Secure Software Development Practices Adopted from CLASP model

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 the 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 the 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

C. Phase 3 Delphi Study: To determine Secure Software Development practices which are dependent on each factor

The third and final phase of Delphi is aimed to map the Secure Software Development practice with factors that influence the implementation of the practice. The practices were adopted from the CLASP model as shown in Table 4.

The data from the phase 3 study was analysed, and the factors were ranked according to the total number of practices that were being influenced by each factor. Figure 1 shows the top 10 factors that influence the implementation of secure software development practices.

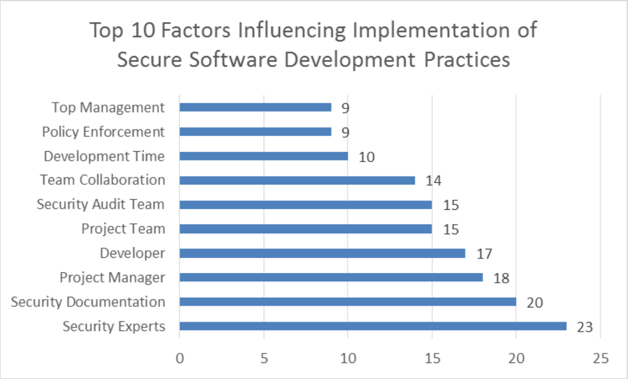

Figure 1: Ranking of top 10 factors that influence the implementation of Secure Software Development Practices

The Delphi study was conducted to facilitate the development of Secure Software Development Practice Adoption Model based on the factors achieved by public service organizations. In Phase 3, the practices were mapped to the factors that affect the implementation of each practice. Table 5 depicts the list of practices affected by each factor.

Table 5
List of Secure Software Development Practices by Influential Factors

	Factors	List of Affected Practices
A	Institutional Context	
1.	Change Management	P1, P2
2	Policy Enforcement	P1, P2, P3, P7, P16, P21 – P24
3	Security Training and Awareness	P1, P2, P7, P11, P12, P23
4	Reward and Incentives	P1
5	Organization's objectives and culture	P1, P4, P24
B	People and Action	
6	Developer	P1, P6, P7, P9 – P22
7	Top Management	P1, P4, P7, P8, P16, P17, P22 – P24
8	Security Experts	P1 – P21, P23 – P24
9	Project Manager	P1 – P10, P13, P16, P17, P20 – P24
C	Project Content	
10	Automated tool support	P2, P8 – P11, P14, P18 – P19
11	Cost	P1 – P2, P8, P11, P14, P18
12	Project Team	P1, P4-P7, P9, P11-P13,P15 –P16, P20 – P22, P24
13	Security Audit Team	P1, P2, P4-P7, P11, P17-P21,P24
14	Segregation of role	P6
15	Team size	P10 –P13, P19 –P20, P24
16	Team Collaboration	P1-P6, P9,P11,P13,P17, P18,P20,P22-P23
17	Development Time	P1-P2, P8, P11-P14, P18-P20
D	System Development Processes	
18	Security Documentation	P1-P3, P6-P17, P19,P21-P24
19	Software development methodology	P6,P21,
20	Internal Metrics and KPI	P2, P4, P8, P21, P23

V. Discussion

Based on the results shown in Figure 1 and Table 5, Security Experts influences the implementation of secure software development practices the most as it is top of the list. As secure software development involves technical knowledge, the need for having a security expert becomes an important factor followed by Security Documentation, the Project Manager, and Developer. Meanwhile, the less influential factors are Change Management, Rewards, and Incentives, Segregation of Role and System Development Methodology. From the perspective of the secure software development practices, implementation of P1(15 factors) and P2 (12 factors) are influenced by the most number of factors meanwhile P3(5 factors), P8(5 factors) and P15(4 factors) are the least influenced practices. This study determines the factors and practices affected by the factors. The results obtained can be used to assess the achievement of factors and identify the practices that are able to be implemented at any organization. Since CLASP model enables tailoring of its practices, this model can be used to select the most applicable practices by the organization, hence produce an acceptable secured software. Furthermore, the organization will also be able to improve on their weaknesses by taking measures in achieving all the factors required to implement secure software development practices.

VI. Conclusion

Security has become a very important quality attribute for all kinds of software and must be considered from the initial stages of the software development process. The insecure software can lead to loss of revenue and damage the business reputation. Public service organizations are facing challenges in implementing secure development practices due to the high cost, lack of skills and development time. This study has taken a novel approach by identifying factors that affecting each secure development practice. These factors are assessed by assessment indicators to identify the achievement level of the factors for any software development project. Based on the achieved factors, the list of practices that can be applied to the software project can be identified. This approach enables software to be developed with some security practices based on the organization's environment and also improve the factors not achieved by the organization. Future work involves evaluating the model by conducting multiple case study in Malaysian Public Service Organization.

Acknowledgment

The authors would like to thank the Universiti Teknologi Malaysia for their support and cooperation including students and other individuals who are either directly or indirectly involved in this project.

References

[1] MAMPU, *The Malaysian Public Sector ICT Strategic Plan*. 2011.

[2] Xiao, S., J. Witschey, and E. Murphy-Hill, *Social influences on secure development tool adoption: why security tools spread*, in *Proceedings of the 17th ACM conference on Computer supported cooperative work &#38; social computing*. 2014, ACM: Baltimore, Maryland, USA. p. 1095-1106.

[3] Thuraisingham, B. and K.W. Hamlen. *Challenges and Future Directions of Software Technology: Secure Software Development*. in *Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual*. 2010. IEEE.

[4] Woon, I.M.Y. and A. Kankanhalli, *Investigation of IS professionals' intention to practise secure development of applications.* International Journal of Human Computer Studies, 2007. 65(1): p. 29-41.

[5] Turoff, M., *The design of a policy Delphi.* Technological forecasting and social change, 1970. 2(2): p. 149-171.

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

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

[8] Dalkey, N. and O. Helmer, *An experimental application of the Delphi method to the use of experts.* Management science, 1963. 9(3): p. 458-467.

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

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

[11] Kanniah, S.L. and M.N.r. Mahrin, *A Review on Factors Influencing Implementation of Secure Software Development Practices.* World Academy of Science, Engineering and Technology, International Journal of Social, Behavioural, Educational, Economic, Business and Industrial Engineering, 2016. 10(8): p. 2860-2867.

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

[13] Alnatheer, M., T. Chan, and K. Nelson. *Understanding And Measuring Information Security Culture*. in *PACIS*. 2012.

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