INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Challenges and limitations in secure software development adoption - A qualitative analysis in Malaysian software industry prospect

Zulfikar Ahmed Maher[1], Asadullah Shah[2]*, Shahmurad Chandio[3], Hazwani Mohd Mohadis[4], Noor Hayani Binti Abd Rahim[4]

1 Kulliyyah of Information and Communication Technology, International Islamic University
2 Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Tel.: +6016-2977026
3 Institute of Mathematics and Computer Science, University of Sindh Jamshoro
4 Kulliyyah of Information and Communication Technology, International Islamic University Malaysia

OPEN ACCESS

Received: 07-06-2020
Accepted: 03-07-2020
Published: 27-07-2020

Editor: Dr. Natarajan Gajendran

Citation: Maher ZA, Shah A, Chandio S, Mohadis HM, Rahim NHBA (2020) Challenges and limitations in secure software development adoption - A qualitative analysis in Malaysian software industry prospect. Indian Journal of Science and Technology 13(26): 2601-2608. https://doi.org/10.17485/IJST/v13i26.848

***Corresponding author**.
Asadullah Shah

Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Tel.: +6016-2977026
asadullah@iium.edu.my

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

Abstract

Background/Objectives Inclusion of security in software development from the initial design phase has not been consistently addressed by the software developers. As a result there is an abundance of software systems with weak security. The objective of this study is to find out factors influencing developer's intention to adopt secure software development practices. **Methodology**: This study is based on qualitative research methodology. Interviews were conducted from the professionals working at senior positions at Malaysian software development organization. All the interviews were first transcribed, as they were digitally recorded. Then transcribed data was analyzed in a way that all frequent words or repetitive concepts were highlighted, after which many similar or relevant concepts were grouped together and categorized as themes and sub themes. **Findings**: The data was analyzed using the thematic analysis method. The results revealed five main themes, whereas each main theme has subthemes. These subthemes are parameters to justify the main theme. Main themes were identified in the light of the interviewee's response. The main results include interviewee's demographic characteristics, and then the main themes identified include, Adoption of SSD practices, Influencing authorities, Motivating Factors, Attitude towards SSD, Hindrances / Issues towards SSD Adoption. Sub themes included: Security Culture, Change Management, Applications of SSD, Managers, Security Expert, Training, Incentives, Security Awareness, Performance Expectancy, Facilitating Conditions, Demographic Characteristics, Need to use SSD, No clear guidelines, Strict Project Timeline, Lack of Security knowledge. The overall interview results show that secure software development practices adoption level in most part of the software industry is not up to the satisfactory level. **Novelty/Applications:** This research explores the factors impeding the

implementation of the best security practices, and barriers to secure software development practices adoption. This study can be used as guideline to be followed for the implementation of secure software development practices in software industry.

Keywords: Secure software development adoption; organizational factors; software developer intention; security development; software security

1 Introduction

Software applications are often produced in the fastest and cheapest way, with no or little focus on security. Software security is a relatively new field and has been pointed as an afterthought. Firstly, the software is released, and then the security problems that are found during its usage are fixed. Realizing security at later stages of software development (SD) results in increased risks of occurring security flaws. Fixing system risks and vulnerabilities after software development cost high for developers and users. This fact can be observed when reading the release notes of a software product, which usually indicate some patches to fix vulnerabilities. The problem with this reactive approach is that there could be potential consequences with the exploitation of the discovered breaches such as brand reputation damage and money losses.

Software security is essential for protecting assets, resources and the information of an organization and the individuals. Data is the most valuable asset and to protect the data of an organization is very important. There is a need of consideration for software security during software development process. Usually, security is often addressed after the software implementation phase and its being ignored at initial phases of the software development. Historically, security has been considered as an afterthought in software development, where the focus was mainly on functionality. However, increasing threats led to acknowledging the importance of addressing security in the development lifecycle [1]. From recent past, big software firms are taking initiatives for security integration within their development life cycle, Such as, Google has appointed an independent Security Team which is responsible for reviewing security during the design phase and implementation phases of their software development, this team also provides consultation and related remedies on security risks. Microsoft has employed a security-oriented software development process called Microsoft Security Development Lifecycle (SDL) since 2004 [2]. This process considers security concerns from the early stages of software development life cycle (SDLC). Many proposals have been presented for incorporation of security in SDLC [3] by integrating security from the early stages of the SDLC when vulnerabilities are less expensive to mitigate. Considering security at early stages has showed much better outcomes as compared to when security was viewed as an additional task.

Despite these efforts, software vulnerabilities persist. With increasing connectivity and progress towards the Internet of Things (IoT), threats have changed [4] and software security is often critical. Also, the security threats are not limited to the large enterprises; even Small and Medium Enterprises (SMEs) organizations are frequently been targeted by the cyber-attacks.

Developing secure software is not a straight forward task; expertise from a number of people are needed to accomplish this task. Initially requirement engineers and software designer are required to collect security requirements along with the functional requirements of the software to be developed to accommodate the software developers. From software functional and non-functional requirements, its architectural diagrams are defined in a way that it can facilitate software developers to develop secure software system. There are a number of considerations in this process of developing secure

software which includes; technical limitations of the software developers, a set of constraints, and functional goals of the system to be developed. Taking into consideration that most of the developers are expert at coding functionality of the software system, but they lack expertise in security implementation. Developing a secure system needs expertise in secure software development practices. Proper knowledge of security implementation and how to develop security mechanisms in a software system is a challenging job for a developer.

From the recent past, software industry has focused on the need of the support for software developers to adequately address with security and privacy [5–7]. Developers, although considered experts in their own domain, are typically not security experts [6]. They sometimes make mistakes that affect the privacy and security of their whole system [5, 6]. It is difficult for the non-security expert software developers to understand security constraints as there is lack of common methods related to security modeling. Most of the software developers lack security expertise [8] due to which they face difficulty in deploying security constraints [9, 10]. The complexity of security mechanisms makes it difficult for an ordinary software developer to understand the security mechanisms and fulfill the security requirements to achieve the secure implementation goal. Identifying potential security threats and security vulnerabilities during software development process is not easy for software developers as they are not usually security experts [10]. For this challenge, there is no clear solution has been provided [11]. Software developers find it difficult to select appropriate security mechanisms because there are a number of security mechanism are present in the literature as well as from the industry without concrete guidelines for their use, secondly location of the security code within the system along with its abstraction is also deemed difficult by the developers [12]. Software developers need concrete guidelines for developing secure applications [13]. It is also important to guide developers about different security attacks and their mitigation within the developing system is also very important [14]. In [15] authors discussed the availability of security modeling and analysis tools. A number of methods have been presented in literature for inclusion of security requirements at initial phase of software development, but there is a lack of connection between these security requirements with the design of secure architectures. Despite the fact that there are a number of methodologies present for the development of secure software system, majority of the developers are reluctant to use them because most of the software developers lack the skills and experience needed to use these methodologies. However, software developers working in the industry might not be willing to use these methodologies because of the additional time, costs, and effort needed for secure development.

Some of the studies found in the literature focuses on the exploring the factors affecting secure software development. The authors in [16] proposed a model based on unified theory of acceptance and use of technology 2 (UTUAT2) to investigate the Influencing factors that affect the implementation of secure software development practices among software developers. The authors focused on using the basic constructs of UTUAT2 theory with one additional construct Switching Cost (SC). However, that research paper was limited to the proposal and no results have been presented in that study. In another study [17], authors analyzed 44 studies by performing a detailed literature review to identify factors affecting secure software practices adoption. The authors listed 24 factors which they identified and characterized them in four categories as (i) Institutional Context (ii) People and action (iii) and (iv) System Development Process. Authors in [18] performed a quantitative survey to investigate factors which affect information systems professional's intention to use secure development practices. Total 184 information security professionals took part in the survey. The survey was developed based on two major theories of information system namely as; theory of planned behavior and theory of reasoned action. Attitude and subjective norm were found consistent factors based on both theories on information systems professional are intention to use secure development practice. Some of the other studies found on literature focuses on other aspects such as [19] finding factors relevant to implementation of Privacy by Design (PbD) by exploring the influencing factors related to an individual and from the organizational contexts. In [20], authors presented the results of a survey of the existing literature focusing on fog computing and pointed out the security gaps in its application. The authors in [21] discussed the evolving issues in Internet of Things and scrutinizing its privacy implications. Secondly, they classified the privacy threats and highlighted the challenges related to privacy of Internet of Things.

Comparing with the previous studies [16–18] in similar prospect, this study provides more detailed results as one to one interview were conducted with the experts from the industry. While most of the previous studies were only proposals such as [16] and mostly other studies were based on the qualitative survey methods, where respondents only have to choose an answer from the given options. In interview method there is a provision of getting the answers in more detail and can dig further deep in the problem space. In this way the opinions from the expert built a strong argument related to the topics discussed. The details of the results obtained are discussed in the section 3.

2 Methodology

Qualitative research approach was adopted for this study using face to face interview method for data collection. The significance of utilizing the qualitative research is clearly recommended to build up a picture from the participant's words. For instance, in a face to face interviews, the participant's expressions can be watched and speeches can be analyzed with more profound

comprehend of the perspectives communicated.

The expert sampling techniques was used in this study. This is a non-probability sampling technique used for the collection of data when the consideration of information is more important than sample size. For this purpose, experts in the software industry were identified, and after their consent, they were interviewed. A letter for permission was sent to higher authorities, and the same was forwarded to target interviewee for the appointment to conduct interviews. The target interviewees responded by phone and fixed the time for the interview at their offices. Although five out of six responded, in the end four interviews were successfully conducted.

The qualitative research required in-depth data to fully understand the phenomenon. Some ethical issues may arise, thus the process for data collection requires a sufficient level of trust and the highest level of moral and efficient communications. Hence, to cover field issues, participants were interviewed at their preferred location and time. A reminder was sent to them a day before the interview. The equipment for the interview checked prior to interview location. To maintain ethics in this research, standard ethical principles were adopted, like reciprocity, assessment of risk, confidentiality, informed consent, data access, and ownership (Creswell, 2012).

The participants in this research were assigned a number or aliases in analyses to protect their identity. Moreover, a simple gift of honor was given to participants as a token of gratitude for their insights, while spending time sharing information, experience and ideas.

3 Results and Discussion

3.1 Demographic details of participants

Demographic details [Table 1] revealed that all four respondents for qualitative interview were male; three candidates were with strong academic backgrounds of qualification of Masters and one candidate having Diploma. One of them has the designation of Chief Technology Officer (CTO) and the other with the designation of Security expert. One candidate was working senior developer and the fourth one was working as Project Manager. Moreover, the interviewee one had 20 years of software industry experience, while the second interviewee had 16 years of development experience while the third interviewee had 12 years of programming experience and the fourth one was 20 years experienced. The age of one respondent was between 45, while the second one was 44 years old, the third one was 38 and the fourth one was 42 years of age. Although all of them specialize in programming and software development, but they belong to different organizations.

Table 1. Demographic Details of the participants

Number	Gender	Age	Work Experience	Job Tittle	Specialization/ Qualification
1	Male	45	20 years	Chief Technology Officer (CTO)	Masters
2	Male	44	16 years	Security Expert	Masters
3	Male	38	12 years	Senior Developer	Diploma
4	Male	42	20 years	Project Manager	Masters

3.2 Themes and subthemes

A careful analysis of the qualitative data resulted in five main themes, whereas each main theme has subthemes [Figure 1]. Subthemes are parameters to justify the main theme. Main themes were identified in the light of interviewee's response to the question presented. The main results start, with interviewee's demographic characteristics, then the main themes identified include, Adoption of SSD practices, Influencing authorities, Motivating Factors, Attitude towards SSD, Hindrances / Issues towards SSD Adoption. [Figure 1] demonstrates the main themes and subthemes. Sub themes included: Security Culture, Change Management, Applications of SSD, Managers, Security Expert, Training, Incentives, Security Awareness, Performance Expectancy, Facilitating Conditions, Demographic Characteristics, Need to use SSD, No clear guidelines, Strict Project Timeline, Lack of Security knowledge.

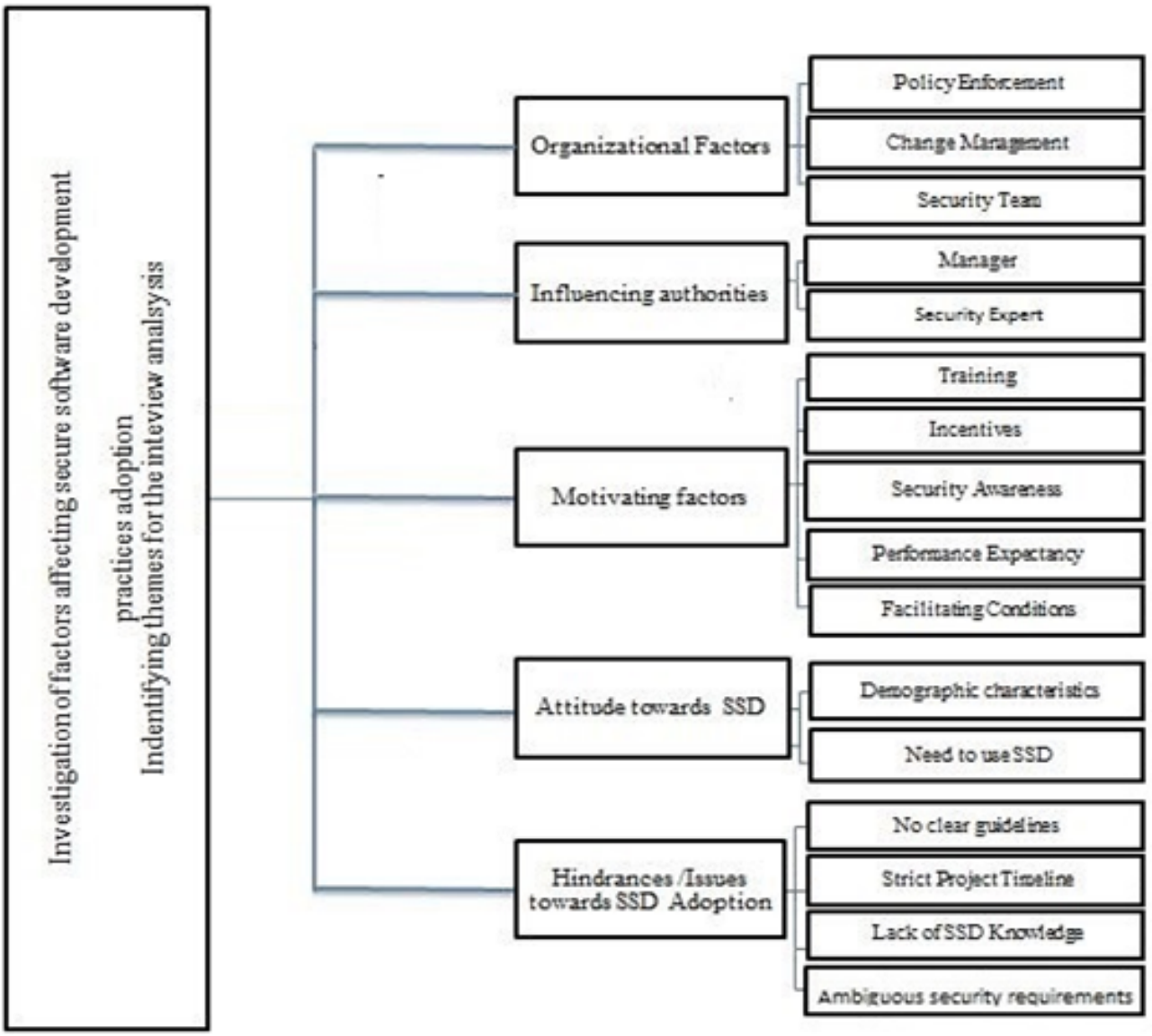

Fig 1. Themes and subthemes

3.2.1 Theme 1: Organizational factors

Mostly the organizations around the world are based on vision and adopted policies to reach their stated visions. Likewise, interviewers discussed vision and policies of their organizations towards the use of secure software development.

The results showed that despite the importance of developing secure application, the interviews indicated its lack in industry. The analysis indicated a lack of policy enforcement by the top management to use SSD practices in industry. It was even reported that there is less concern from organization to management related to security and there is no clear policy has been introduced.

Therefore, it was evident that there was no proper management or proper policy guidelines are presented for security incorporation in software development. Moreover, it was explained that most of the developers are not comfortable when they are asked to adopt a new technique. Because it requires extra effort for them to learn and master the technique and also they need extra time for this. The other respondent said that there is resistance against adopting secure development practices because their organization does not offer any reward or incentive for it. The last respondent said that the culture for using secure software

development has not been developed because the much needed change has not been embraced and managed effectively. Overall, it can be assumed that there is a lack of vision and effort by the management of the organization towards the secure software development practices and methodologies use; it needs to be seriously highlighted and focused.

3.2.2 Theme 2: Influencing authorities

The second theme derived from the interview analysis was influencing authorities that affect the secure software development methodologies use. Management and security expert were identified as sub themes. Both factors play the leadership role to monitor and develop the secure application. It was noted that the main monitoring and controlling authority is Managers working in software firms, which not only provides guidelines but also initiates and introduces new technologies and methodologies within the organization.

The top management incorporates policies, sets objectives, prepares strategies and makes available necessary resources, and performs monitoring role to implement policies. Top management's commitment and participation strongly influence organizational culture which in turn impacts employee's attitudes towards perceived behavioral control over use of secure software practices. The strong influence of managers is evident in the interviews that there is a strong influence of the top management for the use of secure development. Based on the results, it is assumed that top management is not very clear for secure software development application in organization nor are they very serious about making it necessary for all the developers to use secure software development practices. The other problem was highlighted is lack of the presence of security experts in the development teams or within the organization which adversely affects the use of secure software development practices.

3.2.3 Theme 3: Motivating factors

Motivation is defined as the process of inspiring someone towards something. The results demonstrated a few motivating factors that can affect the intention of developers to use secure software development practices. A number of studies had been conducted in various fields to know the motivation of people for specific situations. This research was also conducted to know the motivation of developers towards the use secure software development practices. The data analysis produced the various factors related to motivation, which provide the base for a developer to focus on secure software development practices. All related factors were given as subthemes, which includes Training, Incentives, Security Awareness, Performance Expectancy and Facilitating Conditions.

The most important motivating factor extracted from the results was training that was being understood as the disseminating factor for the proper use of secure software development practices. It was also evident that training plays a vital role in creating intention in developers towards use of secure software development practices. Therefore, training should be conducted on a regular basis for the improvement of the developer's skills and to learn latest software development methodologies. The second important factor pointed out was incentives. The interviewers were of the opinion that regular incentives in the form of allowance, gifts, certificates, and promotions could enhance the intention of developers to use secure software development practices. The security awareness was also noted as one of the motivating factors in developers. Hence, all interviewees described developer's security awareness level is very important on his decision to use any secure development practice. Developers who are less aware with the security usually tend to ignore the security implementation during their development.

In addition, Performance Expectancy was also found to be an important motivating factor for developers. It was explained in interviews that most of the developers think that if they use any secure software development methodology, it will take extra time to finish. So they think their performance will be slowed down. Additionally, it was mentioned that to reach the standard performance, they should be facilitated with provision of tool which facilitate them to produce secure application without hindering their performance.

Furthermore, facilitating conditions were also considered important. All the interviewees focused on proper infrastructure and facilitations to be provided to developers. They were still not satisfied with available facilities though they expressed that if they have facilitation conditions like security teams, management support to build secure application and availability of security and testing tools will result in better adoption level of secure development.

3.2.4 Theme 4: Attitude towards secure software development

Data analysis revealed that different factors could influence developers to use secure software development practices. Thus, the main theme attitude towards use secure software development practices was formed by combining the related factors as subthemes, categorized as demographic, and need to use SSD. It was revealed during data analysis that demographic factors like gender do not have much difference in intention, but age somehow is influential, especially the senior level developers who studied programming long ago and were less aware of the secure development methodologies show reluctance to use secure

development practices. It was revealed that the younger developers at early stages of their career are much enthusiastic to use new methodologies and they are ready to use secure development practices.

Moreover, it was suggested by the interviewees in the results that developers who are more familiar with the latest software attacks and vulnerabilities will feel more need to use secure software development practices as compared to others.

3.2.5 Theme 5: Hindrances /Issues towards secure software development Adoption

The last theme that emerged from the interviewee's data was the challenges towards use secure software development practices. The theme highlights factors that bring challenges towards the use of secure software development, like no clear guidelines, strict project time line, lack of secure development knowledge and ambiguous security requirements. It was reported that there is a lack of vision, a lack of clear-cut guidelines from the top management of their firm. No clear guidelines on policy matters regarding security to incorporate in the developing systems. In addition, Lack of knowledge about security and Secure software development methodologies is also reported as a big barrier to use of secure development practices. Security related trainings are suggested to increase the secure software development knowledge among developers. It was also reported in interview part that most of the time software was developed with strict time lines to meet the commitment with the customers and also to start working on new project. Due these strict deadlines ignorance towards security is common. Moreover, Software security requirements are reported as difficult to understand for developers. Sometimes a developer doesn't know how to fulfill the certain security requirement which is also a found strong reason be not including security in their routine development. It was also suggested that the mechanism from security requirements specifications to the inclusion is an ambiguous process and it should be performed by separate security expert teams.

4 Conclusion

This study revealed that there is a lack of vision, a lack of clear-cut guidelines from the top management of their firm and sometimes there are no clear guidelines on policy matters regarding security to incorporate in the developing systems. When the customer specifically asks about security incorporation then it is being incorporated, projects are initiated but due to lack of compulsion not implemented. So organizations must work upon the work force as well we must work out the standards Operational Procedures (SOPs) which developers can follow. Another problem reported that most of the times software was developed with strict time lines to meet the commitment with the customers and also to start working on new project. Due these strict deadlines ignorance towards security is common. Most of the time developers use basic agile development methodology in this case. It was further inferred that that most of the software developers are not comfortable when they are asked to adopt a new software development technique. Finally, it is expected that this research will provide guidelines to software industry to understand the expected challenges and issue related to secure software development adoption by their developers. The result of this study fills the gap of the literature related to the behavior of software developers towards adoption of secure software development practices and provides important guidelines for the software development practices adoption and recommend improvements for the usage behavior of developers. By following the recommendation of this study, the software industry can plan a smooth process of secure development adoption.

References

1. Geer D. Are companies actually using secure development life cycles?. *Computer.* 2010;43(6).
2. Chess B, McGraw G. Static analysis for security. *IEEE Security and Privacy Magazine.* 2004;2:76-79.
3. Fonseca J, Vieira M. A survey on secure software development lifecycles. in *In Software Development Techniques for Constructive Information Systems Design* (others . , ed.):57-73IGI Global 2013.
4. Howard M, Lipner S. *The security development lifecycle*;8. Redmond: Microsoft Press. Scholar Digital Librar. 2006.
5. Acar Y, Fahl S, Mazurek ML. You are not your developer, either: A research agenda for usable security and privacy research beyond end users. . in *In2016 IEEE Cybersecurity Development (SecDev)* (others . , ed.):3-8IEEE 2016.
6. Green M, Smith M. Developers are Not the Enemy!: The Need for Usable Security APIs. *IEEE Security & Privacy.* 2016;14(5):40-46.
7. Pieczul O, Foley S, Zurko ME. Developer-centered Security and the Symmetry of Ignorance. *Proceedings of the 2017 New Security Paradigms Workshop.* 2017:46-56.
8. Bouaziz R, Scristudio KS. A security pattern integration tool. in *In2016 International Conference on Information Technology for Organizations Development (IT4OD)* (others . , ed.):1-6IEEE 2016.
9. Vieira M, Antunes N. *Introduction to Software Security Concepts. In Innovative Technologies for Dependable OTS-Based Critical Systems.* Milano: Springer 2013.
10. Maher ZA, Sani NFM, Din J, Jabar M. Use of Security Patterns for Development of Secure Healthcare Information System. *Journal of Medical Imaging and Health Informatics.* 2016;6(6):1541-1547.
11. Fernandez EB. *Security patterns and a methodology to apply them. InSecurity and dependability for ambient intelligence.* Boston, MA: Springer 2009.

12. Bouaziz R, Hamid B, Desnos N. Towards a better integration of patterns in secure component-based systems design. *In International Conference on Computational Science and Its Applications.* 2011:607-621.
13. Lodderstedt T, Basin D, Doser J, Secureuml . SecureUML: A UML-based modeling language for model-driven security. in *InInternational Conference on the Unified Modeling Language*(Berlin, Heidelberg):426-441Springer 2002.
14. Lincke SJ. Designing software security with UML extensions: post-conference workshop. *Journal of Computing Sciences in Colleges.* 2012;28(1):149-52.
15. Vysoký M. Diagram of Security. Information Sciences and Technologies. *Information Sciences and Technologies.* 2012;4(1):39-39.
16. Maher ZA, Shaikh H, Khan MS, Arbaaeen A, Shah A. Factors Affecting Secure Software Development Practices Among Developers-An Investigation. in *In2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (others . , ed.):1-6IEEE 2018.
17. Kanniah SL, Mahrin MN. A review on factors influencing implementation of secure software development practices. *International Journal of Computer and Systems Engineering.* 2016;2(8):3032-3039.
18. Woon IM, Kankanhalli A. Investigation of IS professionals' intention to practise secure development of applications. *International Journal of Human-Computer Studies.* 2007;65(1):29-41.
19. Bu F, Wang N, Jiang B, Liang H. "Privacy by Design" implementation: Information system engineers' perspective. *International Journal of Information Management.* 2020;53:102124-102124.
20. Khan S, Parkinson S, Qin Y. Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing.* 2017;6(1).
21. Ziegeldorf JH, Morchon OG, Wehrle K. Privacy in the Internet of Things: threats andchallenges. *Security and Communication Networks.* 2014;7:2728-2742.