INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN BUSINESS & SOCIAL SCIENCES

Vol 14, Issue 1, (2024) E-ISSN: 2222-6990

Securing Software Development: A Holistic Exploration of Security Awareness in Software Development Teams

Aftab Janisar[1], Khairul Shafee[2], Aliza Sarlan[3], Umar Maiwada[4] and Anas A. Salameh[5]

[1,2,3,4]Department of Computer and Information Science Universiti Teknologi Petronas, 32610 Seri Iskandar Perak Malaysia, [5]Department of Management Information Systems, College of Business Administration, Prince Sattam bin Abdulaziz University, 165 Al-Kharj 11942, Saudi Arabia

Corresponding Author Email: aftab_22001362@utp.edu.my

To Link this Article: http://dx.doi.org/10.6007/IJARBSS/v14-i1/20545 DOI:10.6007/IJARBSS/v14-i1/20545

Published Date: 10 January 2024

Abstract

Security awareness is crucial at every stage of the software development life cycle. Studies emphasize the importance of addressing security requirements (SR) early in the requirement engineering phase to effectively mitigate security issues. However, the software development team (SDT) currently lacks sufficient awareness regarding the security requirements assurance (SRA) for mitigating security issues in secure software development. The objective of this study is to assess the (SDT) security knowledge in early software development. A survey was distributed, questions were based on (SR) within the context of security requirement engineering (SRE). A total of 58 responded to the survey. The results indicate that the (SDT) demonstrates a satisfactory level of knowledge regarding security (KOS), security requirements elicitation and analysis (SREA), and approaches within the domain of SRE. However, the results pertaining to security requirement assurance (SRA) were found unsatisfactory. Descriptive statistics were employed to analyse the mean scores of KOS=3.79, SRE=3.61, SREA=3.67, and SRA=2.71. SRE presented the strong Pearson correlation with SREA=.596**. Also, regression coefficient produces positive outcome with (SRA) and (SREA). Though, software development teams need to collaborate with the researcher to enhance the awareness about security requirement assurance during the secure development process.

Keywords: Software Development Life Cycle (SDLC), Requirement Engineering, Security Requirement Engineering, Survey, Software Development Organization (SDO), Security Knowledge, Security Requirement Elicitation and Analysis, Security Requirements Assurance

Introduction

Software development enterprises (SDE) are important to the growth of a country's economy (Ragkhitwetsagul et al., 2022). Software development enterprises (SDEs) which include software startups, are essential for a nation's competitiveness and innovation (Laporte & O'Connor, 2016). With the increasing number of software development enterprises, growing demand for software has rapidly increased with the increasing reliance of users on the software. In today's competitive economic environment, the demand for reliable and secure software is growing by the day. It is feasible to create secure software by considering functional and nonfunctional requirements in software development process (Li et al., 2017).

Though, it is crucial from the start to ensure the security of software, to avoid security issues later in software development (Niazi et al., 2020). Reason behind security issues and software failure is overlooking the security requirements in requirement stage (Li et al., 2017). Security requirements include system confidentiality, availability, integrity, accessibility, and accountability etc. are the security goals. Security requirements are sometimes added after system design (Canedo et al., 2022). This overfitting of security requirements in the current architecture of the system results in security vulnerabilities (Zareen et al., 2020). Early identification of security requirements secure the software systems, According to the 2016 California Breach Report, “nearly 50 million records of Californians have been hacked and the majority of these breaches originated from security failures” (Harris & General, 2016; Sadiq et al., 2021). Security is commonly categorized as a non-functional requirement, often leading to its validation during the software development's final phase. Nonetheless, emphasizing software security during the initial stages is crucial (Khan & Khan, 2018b).

In recent years, security requirements engineering (SRE) has grown, and security frameworks have been proposed in the academic community and other scientific publications (Rehman et al., 2018). There are now various approaches to security requirements engineering, including “Secure Quality Requirements Engineering” SQUARE, CLASP, secure i*, UMLsec , SREP, secure Tropos and MORSE etc. . Each framework has its advantages and limitations and is best suited for a certain function. Nevertheless, these techniques continue to lack a complete security requirements process and depend on the knowledge of security specialists (Li et al., 2017). Therefore, software engineering teams are faced with a difficult decision when selecting an acceptable security requirements engineering approach based on their demands and expectations. Thus, the objective of this study is to assess the knowledge of software development professionals such as requirement engineers, system analyst, software engineers and others on security and SRE approaches. SRE methods, this work will help organizations in the development of software to better understand existing security initiatives used in the development of secure software. It can also provide researchers with a basis for designing and developing new methods of software security and identifying new axis of research.(Khan et al., 2020).

The National Institute of Standards and Technology (NIST) defines security assurance as "the degree of certainty that the security features, practices, procedures, and architecture of an information system accurately communicate and enforce the security policy." (Jahan, Pasco, et al., 2019). The use of security requirement assurance is increasing in the development of secure critical systems, especially in industries such as transportation systems, medical devices, financial systems, military systems, healthcare, and automotive (Mohamad et al., 2021). Therefore, one of the primary reasons for the success of the threats and attacks is lack of attention to the elicitation and analysis of security requirements (Anderson, 2020), and cannot be neglected any longer (Rehman et al., 2018). The increasing

amount of software security threats, along with increased security awareness, implies that software security assurance is no longer an option, but rather a requirement (Khan & Khan, 2018a).

The rest of this paper is organized as follows: Section 2 discusses relevant work in the domain of security requirements engineering. Section 3 describes the methodology. Section 4 covers the Results and discussion. Section 5 is concerned with the conclusion and future research.

Literature Review

The rising number of security problems in recent years, forced the research community to put security concerns at the top of their list of priorities (Anwar Mohammad et al., 2019). Security should be a part of the whole SDLC process, and it should be a part of it right from the start, in the requirements phase (Mahmood et al., 2022; Sonmez & Kilic, 2021). Requirement Engineering is the fundamental and crucial phase in SDLC (Batta & Srivastava, 2021), because this stage is persuaded over to the remaining phases of SDLC (Fernández et al., 2019) (Melegati et al., 2019; Ambreen et al., 2016; Mall, 2018; Khan et al., 2021).

Security concerns are not implemented by software engineers as a continuous process in early software development; they are valued at the end of software development (Humayun et al., 2023; Nazir & Nazir, 2018). According to the research "Veracode, 2018," software developers aren't paying enough attention to security issues; therefore, all applications are vulnerable to threats (Weir et al., 2022). The majority of software engineers initially do not care about security concerns, Yet, the software engineers are gradually realizing that security for requirements engineering is essential for software development (Weir et al., 2021; Weir et al., 2022). According to recent study, many software development procedures do not clearly contain methods for integrating software security from the early stages of software development (Khan & Khan, 2018b).

Security is considered as a non-functional requirement. Security requirements refer to the set of specifications, standards, and measures that must be put in the place to ensure the confidentiality, integrity, and availability of information and system in an organization (Yeng et al., 2020). Practitioners and researchers have suggested that security related aspects must be incorporated in the requirement phase to avoid the rework and stop spreading the problems in later stages (Qadir & Ahmad, 2022). Unfortunately, addressing the security requirements in requirement engineering phase is still not concrete (Villamizar et al., 2018).

Several studies describe methods for systematically eliciting security requirements (Steinmann & Ochoa, 2022). This Almadani (2022) study explain method for eliciting security requirements based on problem and abuse frames, abuse frames and problem frames were also employed to capture threats and vulnerabilities (Mufti et al., 2018; Ansari et al., 2022). Furthermore, (Zareen et al., 2020) presented a methodology for extracting and modelling security requirements from a business process model. Model-oriented security requirements engineering (MORSE) has been used in web-based E-Health applications (Li et al., 2017). Misuse instances are another technique for eliciting security requirements (Anwar Mohammad et al., 2019), Misuse cases were presented to capture security threats and requirements (Zareen et al., 2020). Security requirement frameworks should be contained in any the SDLC process to enhance the development methodology, such as SQUARE Anwar Mohammad et al (2019); Qadir & Ahmad (2022) Mufti et al (2018), CLASP Qadir & Ahmad, (2022) Mufti et al (2018), secure i* , UMLsec Mohammad et al (2019), SREP (Mufti et al., 2018), secure Tropos Mohammad et al (2019), MORSE Prabhakaran & Selvadurai (2018) and

many more. According to the findings of two recent studies (Fujdiak et al., 2019; Lopez et al., 2019) in order to increase developer teams security, it is necessary to "improve software engineers security awareness;".

Table 1
SRE Approaches and their Security Criteria with Requirement Elicitation and Analysis

	Criteria											Requirement Elicitation					Requirement Analysis			
SRE Methods	Flexibility	scalability	Threats,	Validation of Requirements	Risk analysis	vulnerability	Consistency	Integration of security requirements	Assets identification	Misuse Modeling	Domain Knowledge	Elicit security Requirements	Identify stakeholders	Support Requirement elicitation process	Identify security goals	Other non-functional requirements	Complete SR can be produced.	identify business objectives.	Missed SR can be included	SR conflict can be resolved
SQUARE	✓	✓	✓	✓	✓	•	✓	•	•	✓	•	✓	•	✓	✓	✓	✓	•	✓	✓
Secure Tropos	✓	✓	✓	✓	•	✓	•	✓	•	-	-	✓	✓	✓	✓	✓	✓	✓	•	✓
Abuse Frames	-	✓	✓	✓	•	✓		•	✓	-	-	✓	•	•	✓	•	✓	✓	✓	•
SREP	✓	✓	✓	✓	✓	✓	✓	•	✓	•	•	✓	•	•	✓	•	✓	✓	✓	✓
Misuse-cases	✓	•	✓	•	•	✓	•	✓	•	✓	•	✓	✓	✓	✓	✓	✓	•	✓	✓
UMLsec	-	-	✓	•	✓	✓	-	-	•	✓	-	✓	•	•	✓	•	✓	✓	✓	•
MORSE	✓	•	✓	•	•	✓	•	✓	•	•	•	✓	✓	✓	✓	•	✓	✓	✓	✓
SecureUML	-	-	•	•	•	•	-	-	•	-	-	✓	•	•	✓	•	✓	✓	✓	✓

Early in the software development process, proper assurance measures with security assurance must be implemented to prevent future problems in critical systems domains (Kabir, 2021; Khan & Khan, 2018b; Mohamad et al., 2021). Research studies conducted on security assurance are (Marshall, et al., 2019; Mohamad et al., 2021; Maksimov et al., 2019; Jahan et al., 2020; Lin et al., 2020; Calinescu et al., 2017; Bloomfield et al., 2017). However, the benefit of adding the security assurance for security requirements will raise confidence and enhance the integrity of software. Knowing the impact will help stakeholders maximize the positive effects to meet security goals/measures (Katt & Prasher).

Research Methodology

This study adopts a quantitative approach to determine the knowledge level of security and awareness of software development teams in perspective of security requirement, security requirement assurance and security requirement engineering. The participants of the study are software engineers, system analysts, security analysts, business analysts, and product managers from software development organizations (SDO's). Participants were contacted using their emails and shared online google form with them to gather data.

Survey

The survey consists of four major sections. The first section is about general information, which includes several listed questions in a systematic order. This section is knowledge about security (KOS), which provides a variety of options, evaluated by the participants. Second section about security requirement engineering, which provides the necessary information and knowledge about SRE domain. Third section of this study focuses on the comprehensive analysis of security requirement elicitation and analysis (SREA). To achieve this, a carefully

structured series of security questions is presented to the participants in a logical sequence. Fourth section is about security requirement assurance (SRA), which provides generic information and knowledge about security requirement assurance approaches and its understanding. Participants of the survey were selected at random to inquire about the security concern in the software development process. The survey was designed in a way to ensure that the questions in this survey that effectively covers various concepts, theories and practices in security requirements assurance, security requirements elicitation and analysis and security requirement engineering. The survey was based on closed-ended questions and distributed among the SDO's participants. The participants were contacted by their emails and shared online google form to gather the responses. The participants of the study are software engineers, system analysts, security analysts, business analysts, and product managers from SDE's. The SDE's participants of this study were from Pakistan, Nigeria, and Qatar's small and medium enterprises.

Data Collection

The data of this study was collected through a survey. In this study, a quantitative method is employed to evaluate the amount of knowledge in KOS, SRA, SREA, and the level of awareness regarding the SRE domain of software development teams. The concepts of software engineering existing theories, literature, modified Sommerville's requirement engineering practices, SRE and SREA existing frameworks were adopted in designing and categorizing the survey and its questions (Mufti et al., 2018). This is to ensure that the questions in the survey effectively cover various concepts, theories and practices regarding SRE, SRA, KOS, and SREA. The survey was validated through feedback from several industrial practitioners who conducted a rigorous review of the questions in the survey. This approach aligns with the methodology employed in prior surveys, where feedback from industry practitioners has proven valuable (Ghani & Besrour, 2012). Notably, the survey's question set was organized into four distinct categories: (SRE), (KOS), (SRA), and (SREA). By adopting this categorization, we aimed to comprehensively address the relevant areas within the field of software engineering while ensuring the survey remained focused and efficient (Garousi et al., 2016). This survey contains a total of 28 closed-ended questions, 10 of which pertain to SREA, where 5 questions were specifically grouped for security requirement elicitation and 5 questions for security requirement analysis, 7 security questions belonged to SRE domain in which 4 questions were designed for approaches, techniques, and methodologies, and the remaining 3 were about general SRE awareness, 3 questions were designed about security requirements assurance approaches. The rest of the questions belonged to general knowledge of security. To measure the survey responses, the Likert scale was used to collect the participants' knowledge and awareness. The Likert scale ranges from 1 to 5 in which 1 refers to no knowledge on SRE, SRA, KOS, and SREA and 5 refers to very knowledgeable on SRE, SRA, KOS, and SREA.

Data Analysis

The data was analyzed using descriptive statistics technique which includes mean, standard deviation, Pearson's correlation coefficient and regression analysis. The mean provides a measure of central tendency, allowing for a better understanding of the distribution and means of the SRE, SRA, KOS, and SREA variables. The Pearson correlation coefficient was used to understand the relationship between SRE, SRA, KOS, and SREA variables. This is to determine if there is a positive, negative or no correlation between SREA, KOS, SRA and SRE

domains. Regression analysis is used to understand the effect of SRA variables towards SRE variables. From regression analysis, the understanding of the relationship between the SRA domain and SRE domain can be known. In regression the b-coefficient is considered statistically significant, when the P value is below the acceptable level of significance, or below 0.05. Having a significant value below 0.05 indicates the impact and significance of variable. IBM SPSS were used to perform all descriptive statistics adopted in this study.

Results and Discussions

A total of 58 participants that comprises of software engineers, system analyst, security analysts, business analysts, and product managers participated in this survey. Participants of the survey are practitioners and belongs to well-known enterprises in Pakistan and Qatar. There was a good mix of people who worked for different software development enterprises. Participants belong to various sectors i.e., Chemical, petroleum, Finance, Education, Healthcare, IT industry, Oil and Gas, and Telecom industry, while having responsibilities and operating in IT domains. IT was the most common target sector for the companies that hired the participants.

Table 2
Participants Profile Demographics

Description	Frequency	%
Position		
Software engineer	4	6.8
Business/System Analyst	5	8.6
Developer (ML, Android, IOS, Flutter)	6	10.3
SOC Analyst	9	15.5
Product managers	3	5.1
Test Engineer / Quality control	2	3.4
Others	29	50.3
Industry		
Software and IT	21	36.4
Oil and Gas	1	1.7
Banking	2	3.4
Education	12	20.6
Media	2	3.4
Telecom	9	15.5
Healthcare	1	1.7
Others	10	17.2
IT sector		
All participants are from IT and software sector.	58	100

Knowledge of Security

Knowledge of security refers to the participant's knowledge with respect to requirements engineering. General questions about security were asked. The Mean value of the Knowledge of security is 3.79, which means most of the responses have agreed to the given scenario of Knowledge of security in the survey.

Table 3
Knowledge of Security with Mean and Standard Deviations

Security Questions	Mean	Standard Deviation
Security awareness in RE	3.9	1.3
Security role in RE	2.7	1.3
Security Importance in RE	4.5	0.6
Prior knowledge about security	3.8	1.0
Security maintenance in RE	4.1	0.8
Risk analysis and budget about security	3.3	0.9
Protect Customer/ supplier data	4.2	1.0
Protect Intellectual property	4.3	0.8
Overall Mean of Knowledge of Security	3.79	0.46

Security Requirement Engineering

Participants have a good understanding of security requirements engineering. They have the knowledge and awareness of security requirements, security goals and the identification of assets to be protected. However, they have average knowledge and awareness of SRE techniques and methodologies. Despite having a good understanding of SRE, the overall mean for the SRE questions is 3.61, which indicates the participants have an overall average knowledge and understanding of SRE.

Table 4
Security Requirement Engineering with Mean and Standard Deviations

Security Questions	Mean	Standard Deviation
Awareness about SRE domain	3.6	1.0
Gathering security requirements using SRE	3.5	1.1
Knowledge about important security Goals in SRE	3.7	1.0
Effective SRE techniques for vulnerabilities	3.8	1.2
Awareness about important assets to be identified	3.7	1.0
SRE methodologies for security requirements	3.2	1.3
SRE techniques for security requirements	3.5	1.2
Overall Mean of Security Requirement Engineering	3.61	0.88

Security Requirement Elicitation and Analysis Knowledge and Awareness

In Table 5, questions related to risk assessment, prioritization techniques, checklist, protypes for vague requirements and feasibility studies show a higher mean as compared to the rest of the security question. This indicates that software development teams are aware of security requirements before the commencement of new software development projects. Like the SRE domain, questions that are related to techniques and methods of SREA are considered as average with a score of 3.5. Nonetheless, the overall mean for SREA domain question is 3.67 which indicates that the participants have good overall knowledge and awareness of SREA.

Table 5
Security Requirement Elicitation and Analysis with Mean and Standard Deviations

Security Questions	Mean	Standard Deviation
Feasibility study of new projects	3.9	1.0
Secure prototype for vague requirements	3.9	1.1
Secure scenario before elicitation process	3.6	0.9
Security plan for unexpected situation in elicitation	3.7	0.8
Elicitation techniques reusability for security requirements	3.4	1.1
Requirement analyst team design security boundary	3.4	0.9
Security checklist used in RE analysis phase	3.8	1.1
Prioritization techniques for security requirements	3.9	0.9
Interaction matrix for security measures in analysis	3.5	1.2
Risk assessment process in analysis phase	4.3	0.7
Overall Mean of Security Requirement Engineering	3.67	0.75

Security Requirement Assurance

From Table 6 the participants have a below average understanding of general security requirements assurance. They have very little knowledge and awareness of security requirements, assurance approaches and techniques. The overall mean for the SRA questions is 2.7, which indicates the participants have very low knowledge and understanding of SRA.

Table 6
Security Requirement Assurance Mean and Standard Deviations

Security Questions	Mean	Standard Deviation
Assurance of security requirements knowledge.	2.8	1.1
Security assurance techniques for security requirements.	2.6	1.1
Security assurance metrics techniques for security requirements?	2.6	1.2
Overall mean of Security assurance	2.71	1.1

Correlation among Variables

This study has observed a significant positive association between different variables. The KOS mean has a moderate positive correlation with SRE and ASR (r=.460**)(r=.234**) and a strong positive association of (r= .512**) with SREA. The SRE mean has a strong positive correlation with SREA (r= .596**) and moderate positive correlation with KOS and SRA (r=.460**)(r=.446**). The SREA mean has a moderate positive correlation with SRA of (r= .267**) and a strong positive association of (r= .512** and r=.596**) with KOS and SRE. The SRA mean has a moderate positive correlation with KOS, SRE, and SREA of (r= .234**) (r=.446**)(r= .267**). The overall result of the correlation reflects a positive association between different variables which means they are interconnected and likely to predict an association with each other.

Table 7
Correlation among all Variables

		KOS Mean	SRE Mean	SREA Mean	SREA Mean
KOS Mean	Pearson correlation	1	.460**	.512**	0.234
	Sig. (2-tailed)		0	0	0.077
	N	58	58	58	58
SRE Mean	Pearson correlation	.460**	1	.596**	.446**
	Sig. (2-tailed)	0		0	0
	N	58	58	58	58
SREA Mean	Pearson correlation	.512**	.596**	1	.267*
	Sig. (2-tailed)	0	0		0.042
	N	58	58	58	58
Assurance Mean	Pearson correlation	0.234	.446**	.267*	1
	Sig. (2-tailed)	0.077	0	0.042	
	N	58	58	58	58

Regression Coefficient among Variables

Table 8 interprets that the regression values of security knowledge produce a positive outcome and having a detail knowledge about security in security requirement engineering, this will significantly improve the security perspective in early stages of software development, as b-coefficient is statistically significant if its "Sig." or $p < 0.05$. In this case regression coefficients, the P value is significant. It can significantly improve the security level and avoid security challenges in the software development process.

Table 8
Regression Coefficients Table

Model	Unstandardized B	Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
Constant	0.728	0.455		1.6	0.115
SREA mean	0.602	0.122	0.514	4.924	0
SRA mean	0.247	0.083	0.309	2.96	0.005

Discussion

Security requirement engineering is important in addressing security concerns in building systems that are secure and resistant to attacks. The methods in SRE assist software development teams in aligning the development efforts with security goals, facilitate risk mitigation and promote a systematic to security throughout the software development lifecycle. Thus, it is important for software development teams to have the knowledge and awareness of SRE, SRA and SREA. The findings of the study indicate that software development teams have a good overall understanding of KOS, SRE and SREA but lacking in practical aspects of SRA. Software development teams have good knowledge and awareness of the fundamental concepts in SRE and SREA such as identification assets, threats, and security goals but average understanding in the aspects of SRE and SREA methods and techniques. This can be attributed to the various methods such as SQUARE, Secure Tropos,

SREP and others with each of these methods adopts different approaches and perspectives. This could be a challenge for software development teams to use the most suitable technique for a system development project. Even lacking in SRA knowledge is big challenge for development process because having inadequate security measures and unidentified risks can lead to several challenges, primarily related to the security of software, systems, or products being developed or deployed.

The findings of the study also demonstrate that software development teams do acknowledge the necessity to address security concerns in the elicitation and analysis phase of a software development project. The findings show a high mean score for the questions regarding the need to conduct a feasibility study and risk assessment for new software development projects. This shows that software development teams possess the knowledge to handle the security requirement issues in the elicitation and analysis phase of a software development project. The results show a significant positive association between SRE and SRA in terms of the knowledge between the two fields. Given that the SRA and SREA already belong to the SRE domain, this is not surprising. This would suggest that requirement engineers would be familiar with SRA and SREA if they are familiar with SRE and vice versa. This is supported further with the values from regression analysis which indicates a positive relationship between SRA and SRE. When software development teams increase their knowledge on SRA, SREA, the knowledge on SRE will increase as a result.

Conclusion

Addressing security concerns early in the development of the lifecycle has become a must-have feature in software systems. Addressing security in the early stages of software development lifecycle is crucial. Thus, it is important for software development teams to be aware and knowledgeable in security requirements engineering domain. This research examines the security awareness of software development teams with respect to requirement engineering. The results demonstrate that the software development team have a comprehensive understanding of security and are familiar with SREA and SRE but lacking in practical aspects with SRA. But still researchers need to collaborate with SDOs to enhance the level of security knowledge and awareness about SRE and SRA approaches of software development teams during the secure development process. In future, this research can be enhanced to use it in security requirement elicitation with security requirement assurance for better understanding of the security concerns early in the SDLC.

Acknowledgments

This study is supported via funding from Prince sattam bin Abdulaziz University project number (PSAU/2023/R/1445).

References

Almadani, B. (2022). Structure of security requirements: Insights from requirements elicitation. In.

Ambreen, T., Ikram, N., Usman, M., & Niazi, M. (2016). Empirical research in requirements engineering: trends and opportunities. *Requirements Engineering*, *23*(1), 63-95. https://doi.org/10.1007/s00766-016-0258-2

Anderson, R. (2020). *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons.

Ansari, M. T. J., Pandey, D., & Alenezi, M. (2022). STORE: Security Threat oriented requirements engineering methodology. *Journal of King Saud University - Computer and Information Sciences*, *34*(2), 191-203. https://doi.org/10.1016/j.jksuci.2018.12.005

Anwar Mohammad, M. N., Nazir, M., & Mustafa, K. (2019). A systematic review and analytical evaluation of security requirements engineering approaches. *Arabian Journal for Science and Engineering*, *44*(11), 8963-8987. https://doi.org/10.1007/s13369-019-04067-3

Batta, A., & Srivastava, D. K. (2021). *A novel approach in requirement engineering during software build-up* 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT),

Bloomfield, R., Bishop, P., Butler, E., & Netkachova, K. (2017). Using an assurance case framework to develop security strategy and policies. Computer Safety, Reliability, and Security: SAFECOMP 2017 Workshops, ASSURE, DECSoS, SASSUR, TELERISE, and TIPS, Trento, Italy, September 12, 2017, Proceedings 36,

Calinescu, R., Weyns, D., Gerasimou, S., Iftikhar, M. U., Habli, I., & Kelly, T. (2017). Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Transactions on Software Engineering*, *44*(11), 1039-1069.

Canedo, E. D., Bandeira, I. N., Calazans, A. T. S., Costa, P. H. T., Cançado, E. C. R., & Bonifácio, R. (2022). Privacy requirements elicitation: A systematic literature review and perception analysis of IT practitioners. *Requirements Engineering*. https://doi.org/10.1007/s00766-022-00382-8

Fernández, D. M., Franch, X., Seyff, N., Felderer, M., Glinz, M., Kalinowski, M., Volgelsang, A., Wagner, S., Bühne, S., & Lauenroth, K. (2019). Do we preach what we practice? Investigating the practical relevance of requirements engineering syllabi-the ireb case. *arXiv preprint arXiv:1902.01822*.

Fujdiak, R., Mlynek, P., Mrnustik, P., Barabas, M., Blazek, P., Borcik, F., & Misurec, J. (2019). Managing the secure software development. 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS),

Garousi, V., Coşkunçay, A., Demirörs, O., & Yazici, A. (2016). Cross-factor analysis of software engineering practices versus practitioner demographics: An exploratory study in Turkey. *Journal of Systems and Software*, *111*, 49-73. https://doi.org/10.1016/j.jss.2015.09.013

Ghani, I., & Besrour, S. (2012). Questionnaire based approach to measure security in requirement engineering. *International Journal of Computer Applications*, *54*(9), 31-34. https://doi.org/10.5120/8596-2359

Harris, K. D., & General, A. (2016). California data breach report. *Retrieved August*, *7*, 2016.

Humayun, M., Niazi, M., Assiri, M., & Haoues, M. (2023). Secure global software development: A practitioners' perspective. *Applied Sciences*, *13*(4). https://doi.org/10.3390/app13042465

Jahan, S., Marshall, A., & Gamble, R. (2019). Evaluating security assurance case adaptation.

Jahan, S., Pasco, M., Gamble, R., McKinley, P., & Cheng, B. (2019). *MAPE-SAC: A framework to dynamically manage security assurance cases* 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W),

Jahan, S., Riley, I., Walter, C., Gamble, R. F., Pasco, M., McKinley, P. K., & Cheng, B. H. (2020). MAPE-K/MAPE-SAC: An interaction framework for adaptive systems with security assurance cases. *Future Generation Computer Systems*, *109*, 197-209.

Kabir, S. (2021). Internet of things and safety assurance of cooperative cyber-physical systems: Opportunities and Challenges. *IEEE Internet of Things Magazine*, *4*(2), 74-78. https://doi.org/10.1109/iotm.0001.2000062

Katt, B., & Prasher, N. (2018). Quantitative security assurance metrics.

Khan, R. A., & Khan, S. U. (2018a). A preliminary structure of software security assurance model. Proceedings of the 13th International Conference on Global Software Engineering,

Khan, R. A., & Khan, S. U. (2018b). *A preliminary structure of software security assurance model* Proceedings of the 13th International Conference on Global Software Engineering,

Khan, R. A., Khan, S. U., Ilyas, M., & Idris, M. Y. (2020). *The State of the Art on Secure Software Engineering* Proceedings of the Evaluation and Assessment in Software Engineering,

Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Systematic Mapping study on security approaches in secure software engineering. *IEEE Access*, *9*, 19139-19160. https://doi.org/10.1109/access.2021.3052311

Laporte, C., & O'Connor, R. (2016). Software process improvement in industry in a graduate software engineering curriculum. *Software Quality Professional Journal*, *18*(3), 4-17.

Li, H., Li, X., Hao, J., Xu, G., Feng, Z., & Xie, X. (2017). *FESR: A framework for eliciting security requirements based on integration of common criteria and weakness detection formal Model* 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS),

Lin, C.-L., Shen, W., & Cheng, B. (2020). Measuring confidence of assurance cases in safety-critical domains.

Lopez, T., Sharp, H., Tun, T., Bandara, A., Levine, M., & Nuseibeh, B. (2019). *Talking about security with professional developers* 2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP),

Mahmood, W., Rizvi, S., & Munir, S. (2022). Hindrance to requirements engineering during software development with globally distributed teams. *International Journal of Information Engineering and Electronic Business*, *14*(2), 39-46. https://doi.org/10.5815/ijieeb.2022.02.03

Maksimov, M., Kokaly, S., & Chechik, M. (2019). A survey of tool-supported assurance case assessment techniques. *ACM Computing Surveys (CSUR)*, *52*(5), 1-34.

Mall, R. (2018). *Fundamentals of software engineering*. PHI Learning Pvt. Ltd.

Melegati, J., Goldman, A., Kon, F., & Wang, X. (2019). A model of requirements engineering in software startups. *Information and Software Technology*, *109*, 92-107. https://doi.org/10.1016/j.infsof.2019.02.001

Mohamad, M., Steghöfer, J.-P., & Scandariato, R. (2021). Security assurance cases—state of the art of an emerging approach. *Empirical Software Engineering*, *26*(4). https://doi.org/10.1007/s10664-021-09971-7

Mufti, Y., Niazi, M., Alshayeb, M., & Mahmood, S. (2018). A readiness model for security requirements engineering. *IEEE Access*, *6*, 28611-28631. https://doi.org/10.1109/access.2018.2840322

Nazir, N., & Nazir, M. K. (2018). A review of security issues in SDLC. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *46*(1), 247-259.

Niazi, M., Saeed, A. M., Alshayeb, M., Mahmood, S., & Zafar, S. (2020). A maturity model for secure requirements engineering. *Computers & Security*, *95*. https://doi.org/10.1016/j.cose.2020.101852

Sonmez, O. F., & Kilic, B. G. (2021). Reusable security requirements repository implementation based on application/system components. *IEEE Access*, *9*, 165966-165988. https://doi.org/10.1109/access.2021.3133020

Prabhakaran, S., & Selvadurai, K. (2018). Performance analysis of security requirements engineering framework by measuring the vulnerabilities. *Int. Arab J. Inf. Technol.*, *15*(3), 435-444.

Qadir, N., & Ahmad, R. (2022). SecRS template to aid novice developers in security requirements identification and documentation. *International Journal of Software Engineering and Computer Systems*, *8*(1), 45-52.

Ragkhitwetsagul, C., Krinke, J., Choetkiertikul, M., Sunetnanta, T., & Sarro, F. (2022). Identifying software engineering challenges in software SMEs: A case study in Thailand.

Rehman, S. U., Allgaier, C., & Gruhn, V. (2018). *Security requirements engineering: A framework for cyber-physical systems* 2018 International Conference on Frontiers of Information Technology (FIT),

Sadiq, M., Devi, S. V., Ahmad, J., & Mohammad, C. W. (2021). Fuzzy logic driven security requirements engineering process. *Journal of Information and Optimization Sciences*, *42*(7), 1685-1707. https://doi.org/10.1080/02522667.2021.1972618

Steinmann, J., & Ochoa, O. (2022). *Supporting security requirements engineering through the development of the secure development ontology* 2022 IEEE 16th International Conference on Semantic Computing (ICSC),

Villamizar, H., Kalinowski, M., Viana, M., & Fernandez, D. M. (2018). *A systematic mapping study on security in agile requirements engineering* 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA),

Weir, C., Becker, I., & Blair, L. (2021). *A passion for security: Intervening to help software developers* 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP),

Weir, C., Becker, I., & Blair, L. (2022). Incorporating software security: Using developer workshops to engage product managers. *Empirical Software Engineering*, *28*(2). https://doi.org/10.1007/s10664-022-10252-0

Yeng, P., Wolthusen, S., & Yang, B. (2020). *Comparative analysis of software development methodologies for security requirement analysis: Towards Healthcare Security Practice* Proceedings of the 13 th IADIS International Conference Information Systems 2020,

Zareen, S., Akram, A., & Khan, A. S. (2020). Security requirements engineering framework with BPMN 2.0.2 extension model for development of information systems. *Applied Sciences*, *10*(14). https://doi.org/10.3390/app10144981