IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 9, 2018, accepted May 21, 2018, date of publication May 24, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840322

A Readiness Model for Security Requirements Engineering

YUSUF MUFTI, MAHMOOD NIAZI, MOHAMMAD ALSHAYEB, AND SAJJAD MAHMOOD
Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Sajjad Mahmood (smahmood@kfupm.edu.sa)

This work was supported by the Deanship of Scientific Research at the King Fahd University of Petroleum and Minerals, Saudi Arabia, under Project IN161024.

ABSTRACT The focus on secure software development has been growing steadily in all phases of the software development life cycle. Security awareness in the requirements engineering stage of software development is important in building secure software. One of the major issues faced by the software industry is that many organizations undertake secure software development initiatives without knowing whether they are ready to undertake them. Currently, there is no model to measure the readiness of security requirements engineering in an organization. The objective of this paper is to develop a security requirements engineering readiness model (SRERM) to enable organizations to assess their security requirements engineering (SRE) readiness levels. In order to achieve this goal, a systematic mapping study was conducted to identify the relevant studies in the SRE domain. A total of 104 primary studies were identified, and available evidence was synthesized into 12 security requirements categories and 76 best practices to build a SRERM. Initially, two case studies were conducted in order to evaluate the SRERM in a real-world environment. Based on the outcomes of the two case studies, some modifications were proposed to further improve the SRERM. After modifying the SRERM, two more case studies were conducted in order to evaluate the modifications made to the SRERM. The case study results indicate that the SRERM has the ability to identify the readiness levels of SRE in the software industry.

INDEX TERMS Readiness model, secure requirements engineering.

I. INTRODUCTION

The number of software vulnerabilities is increasing with the growth of Internet-enabled software [1], [2]. Security awareness in the requirements engineering (RE) stage of the software development lifecycle (SDLC) is important in building secure software. Currently, security issues are gaining more attention because of the popularity of social networking systems and cloud computing. Due to the increasing number of users around the world, both cloud computing and social networking systems face more challenges in securing the availability of the system, the integrity of transferred data and the confidentiality of information control [3], [4].

There are a number of common challenges to building secure software. Flaws, bugs, and defects in software are urgent issues and generally demand high attention. According to McGraw [5], it is motivated by the connectivity, complexity and extensibility of the software. Then, various attacks, such as buffer flows, race conditions and incomplete mitigation, could utilize software flaws to disclose access.

In addition, malware (malicious software) is also a challenge to building secure software. Stamp [6] lists various types of malware that are harmful to software, such as viruses, worms, Trojan Horses, trap doors, rabbits and spyware. Several solutions are available to mitigate the risk of malware: signatures, changes, and anomaly detection. For example, to filter malware, users are encouraged to install antivirus software on desktops, network devices, mail gateways and network gateways [4]. However, these are not sufficient because software requirements change over time, so various existing security mechanisms become irrelevant [3]. Therefore, security awareness in RE activities should be encouraged.

Integrating security awareness into the RE stage of the SDLC is an active area of research and needs to be applied to the real-world software industry [7]–[9]. This topic is popularly known as security requirements engineering (SRE). For instance, capturing SR has been a popular area of research, discussed by dozens of researchers for more than

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

two decades [10]–[12]. Recently, it has been applied to cloud computing [13] and the Internet of Things (IoT) [14] research.

The objective of this study is to develop a readiness model for security requirements engineering. The developed model is expected to have the ability to determine an organization's SRE readiness to encompass relevant security requirements, which are reliable in various software organizations. To do this, we address the following research questions:

RQ1. How can a practical and robust readiness model for SRE be developed?

RQ2. Is the developed readiness model robust in term of measuring SRE readiness?

This study provides a consolidated knowledge base of the literature and the main contribution is the software requirements engineering readiness model (SRERM). This model will help the organization to assess the readiness of their SRE. The SRERM provides a practical structure integrated with the assessment of SRE process of organizations. We believe that this research will contribute to the knowledge on SRE for industrial practices.

The remainder of this paper is organized as follows: Section II presents the background and related work. Section III describes our research methodology. In Section IV, we present the SRERM. In Sections V, VI, VII and VIII, we discuss the overall study results and limitations. Finally, Section IX provides the conclusion and discusses how the results from this study can be further used in future research endeavors.

II. BACKGROUND

The definition of security requirements engineering in this research consists of two main terms in software development; security and requirements engineering.

A. SECURITY

In terms of computers and software, security has meant a way of thinking to protect the essential assets of the system, such as information, the operating system, networking and programs. There are three types of security implementation: defense, detection and deterrence [4]. The most effective approach to include security into software development is donning a black hat and thinking like a bad guy [1]. However, software organizations commonly prefer to utilize existing security standards as a guideline to secure their system.

Various security standards are employed to assist information security management. COBIT, ISO 27001 and 27002, National Institute of Standards and Technology (NIST) and common criteria are the most widely discussed security standards in the published studies as these have been developed by well-known organizations and have attracted the attention of more security practitioners than the other types [4].

B. SECURITY REQUIREMENTS (SR)

There are two common definitions of security requirements (SR) in the published studies. The first definition states that SR is a constraint on the functions of the system, whose purpose is to satisfy one or more security goals [12], [15], [16]. SR as a constraint will specify urgent notes or restrictions relating to relevant security concerns to the functional requirements. For example, a functional requirement states a user needs to insert their username and password to log in to the system. SR then ensures the system verifies this information before allowing the user to access the system.

The second definition argues that SR should be considered as a functional requirement [15], [17]. This is similar to the common criteria concept [18] which recommends several security mechanisms as a requirement, and includes a section to discuss the reasons behind them. For example, there is a consideration that ''the user is authenticated by using biometric devices'' as a requirement. When this is documented in software requirement specifications (SRS), it will encourage people to focus on the technical security architectural mechanism and design rather than why biometric devices are selected. In this study, the definition of SR as a constraint is adopted rather than as a functional requirement. In other words, security requirements will document various important assets linked to running software such as information, communication data and the software itself.

C. SECURITY REQUIREMENTS ENGINEERING

Typically, SRE is performed in the first stage of the software development lifecycle. The main activities of SRE include eliciting, analyzing and specifying the security requirements. To support these main activities, SRE also involves validating and managing the collected security requirements. The outcomes of SRE are a security requirement specification which describes the identified assets, detected threats, potential vulnerabilities, analyzed risks and practices [16], [19].

Salini and Kanmani [16] state there are several published SRE methods in real software development. Some of these are McGraw's SSDL process, Microsoft's Trustworthy Computing SDLC, Aprville and Purzandi's SDLC, CLASP (Comprehensive, Lightweight Application Security Process), SQUARE (Security Quality Requirement Engineering), Haley and his colleague's framework, Security Requirement Engineering Process (SREP), and Secure Tropos. One difference between these SRE methods is the number of activities covered. For example, SQUARE has a misuse modeling activity while Secure Tropos and CLASP do not. SREP performs an asset identification activity while SQUARE does not. SREP involves validation activities while Trustworthy Computing SDLC does not. In the authors' opinion, the most recommended SRE method is SREP because it covers the most activities of SRE.

In addition, SRE will heighten people's awareness of the need to improve and ensure the security of software from the beginning of the development phase. It can be interpreted by analyzing the potential threats, such as abuser, attack, malware and theft. As a result, it will lead to protecting the confidentiality, integrity and availability of the software and its information.

D. READINESS MODELS

In software engineering research, a readiness model has been utilized in several studies. It was used by Niazi *et al.* [20] to assess organizational readiness in terms of software process improvement. Their readiness model has several levels: aware, defined, and optimizing. Critical factors and barriers support each level. The researchers validated their readiness model by performing case studies in three software organizations.

Similarly, Ali and Khan [21] presented a model to measure the readiness of a software organization to form outsourcing relationships. To develop a readiness model, they utilized critical partnership factors and examined their practical implementation. Their readiness model also has several levels: contract, success, readiness, conversion and maturity. By utilizing case studies in two software organizations, they argue that their readiness model has the ability to assist software development outsourcing.

As a result, a readiness model can be defined as a technique to assess an organization or team based on the specified criteria to represent their level of readiness. The aforementioned studies utilize the Motorola assessment tool and a case study to show the usability of their readiness model. The challenge learned from the literature is how to construct the levels with practices that can be applied to real software organizations.

E. EXISTING LITERATURE ON SRE

SRE focuses on the early phase while software security covers security knowledge and how to integrate it in the software development lifecycle [1]. This section describes some of the published research, which motivated this study.

Capturing security requirements is a popular topic in the elicitation step of SRE. Several studies describe a technique to elicit security requirements in a systematic way. El-Hadary and El-Kassas [12] proposed a technique for eliciting security requirements based on problem frames and abuse frames. They used problem frames to build a security catalog and to represent security requirements, while the abuse frames are used for threat modeling. Abuse frames and problem frames were also previously utilized by Lin *et al.* [22], [23] to collect threats and vulnerabilities to enhance security requirements engineering.

Another technique for eliciting security requirements is misuse cases. Sindre and Opdahl [24] proposed misuse cases to capture security threats and requirements. Misuse cases provide a visualization of the connection between use cases and misuse cases. Although misuse cases have a trustworthy capability to analyze threats to functional requirements, they also have some weaknesses, such as requiring the developer to have a high level of understanding to know how to improve the misuse case, and furthermore, it does not cover some kinds of threats.

Tøndel *et al.* [25] highlighted the strong potential of combining misuse cases with attack trees [26] to improve security requirements elicitation. They argued that attack trees could provide references of threats in more detail to support the misuse cases. Similarly, Gandotra *et al.* [27] combine the strength of misuse case and attack trees.

Similar to misuse cases, abuse cases were previously proposed by McDermott and Fox [28]. Although both misuse cases and abuse cases employ the concept of the use case, they have an essential difference. While misuse cases are visualized in one single diagram with the use case, abuse cases are separated.

Recently, research has offered different frameworks to overcome some of the challenges of SRE. For instance, Dalpiaz *et al.* [29] proposed a SecCo framework which focuses on elicitation and specification activities to document security requirements. SecCo works by utilizing a commitment view between actors. In addition, Saleem *et al.* [30] presented a framework for eliciting and modeling the security requirements from the business process model. They stated their framework is able to model the security requirements on SOA-based applications.

Furthermore, Salini [11] presented a model-oriented security requirements engineering (MOSRE) framework. They utilized a use case diagram to elicit security requirements. MOSRE has been applied to E-Health web applications. To determine the security requirements, it has the ability to identify, quantify and rank the risks of security threats and vulnerabilities.

Mellado *et al.* [31] proposed SRE a process for the software product line (SREPPLine) framework. They utilized XML grammar and the security reference model in their framework. They argued their framework conforms to ISO/IEC 27001 and common criteria linked to security requirements management concerns. In addition, common criteria [18] as a standardized guideline for eliciting, specifying, and analyzing SR was also utilized in research by Ware *et al.* [32] in combination with use cases for eliciting SR.

Recently, several well-known studies have discussed how to build a framework for SRE [14], [33], [34]. Other fruitfuldiscussions talk about how to implement SRE in cloud system development [35]–[37]. In general, every new technology such as the Internet of Things (IoT) has its own security challenges. From this discussion of these published studies, it can be seen that SRE is an active area of research.

Many research papers have been published that discuss SRE in term of techniques, guidelines, and frameworks. However, they face the challenge of how to assess the strength of SRE implementation in the software industry. There is still no study, which provides a technique, or a tool to identify which security area has been overlooked in software development. Due to the high number of technological challenges and security threats that will be faced in the future, the software industry needs an assistant or tool to indicate the readiness of their SRE process.

III. RESEARCH METHODOLOGY

The research methodology of this research consists of the four following steps:

- Phase 1: Systematic Mapping Study (SMAPS). In this research, we undertook a SMAPS to collect and analyze the studies that are relevant to the following research question:

  RQ. How can a practical and robust readiness model for SRE be developed?

We followed the SMAPS guidelines presented by Petersen *et al.* [38]. There are several processes in the SMAPS: defining the research questions, conducting the search, screening the studies, filtering the abstracts by keyword, extracting the data and mapping.

There are three steps to develop a search string. In the first step, we built the search terms by defining the population, the intervention, the outcome of relevance, and the experimental design that is suitable for our research.

1. Population: secure requirements engineering in software development
2. Intervention: available techniques, models and approaches to satisfy secure requirements engineering
3. The outcome of relevance: secure requirement-engineering techniques, SRE models, SRE approaches.
4. Experimental design: SMS, case studies, empirical studies, theoretical studies.

Based on the results, the search string was configured by using certain keywords such as "secure", "security", "requirement", "software", "engineering", and "approach".

In the second step, we looked for synonyms of the keywords to enhance the quality of the search string. We undertook this step as some studies often utilize different words with the same meaning.

Secure Requirements Engineering: "Security Requirements" OR "Securing Requirements" OR "Secured Requirements Engineering"

Approaches: "guideline" OR "technique" OR "technology" OR "tool" OR "model" OR "framework" OR "approach"

The word "secure" has a similar meaning to "security", "securing" and "secured" in terms of requirements, whereas the word "approaches" contains many potential meanings, such as "technique", "guideline", "model", "tool", and "framework".

In the final step, after identifying the synonyms of each keyword, we then described a general search string that has been applied in research sources. The full search string is defined as follows.

Software AND requirement AND (secure OR security OR securing OR secured) AND (technique OR method OR technology OR tool OR model OR diagram OR approach OR framework OR guideline).

The SMAPS was conducted between December 2016 and March 2017 on five popular digital libraries (IEEE, ACM, Springer, Science Direct, Wiley). The review protocol was developed rigorously to obtain the result. Initially, 924 studies were collected. Based on the analysis process, we selected 104 primary studies that corresponded to the research questions. Then, the outcomes of the SMAPS were utilized to develop the SRERM.

We included those papers which were relevant to our research question and published in English. We excluded those papers that were without bibliographic information.

We used the grounded theory-based coding scheme to review the literature and identify security components. We labelled and grouped the related security components into categories.

- Phase 2: Developing an SRE readiness model (SRERM). The SRERM development was influenced by several published pieces of research that present a readiness model [20], [21], [39], [40]. This study utilizes the outcomes of SMAPS to develop security requirements components, including specifying relevant practices, in constructing the SRERM. In order to evaluate SRERM, we used the Motorola assessment tool [41].
- Phase 3: Conducting a case study. In this phase, a case study was conducted in two software organizations in order to validate the usability of the SRERM. The organizations recommended changes, offered criticisms and suggested modifications. This phase of the SRERM also included a modification section to accommodate feedback from the respondents.
- Phase 4: Performing evaluation and modification. In this phase, the post case study evaluation was performed to gain feedback. The SRERM was modified based on the respondents' suggestions.
- Phase 5: Performing post evaluation. After modifying the SRERM, two software organizations were asked to participate in a case study, the purpose being to analyze the improvement and the usability of the SRERM after modification.

IV. SECURITY REQUIREMENTS ENGINEERING READINESS MODEL (SRERM)

A. DEVELOPMENT PROCESS of SRERM

SRERM is based on the concepts from the software process improvement readiness model (SPIRM) [20], software outsourcing vendor readiness model (SOVRM) [42] and software outsourcing partnership model (SOPM) [21]. There are five security requirements engineering levels in this model, which were adapted from the aforementioned literature. In the previous work, the researchers utilized critical success factors (CSFs) in each level; however, this research uses security requirements categories (SRCs). The identified SRE practices are then modified and distributed into suitable SRCs. Internal reviews and iterative changes were made before external evaluation was undertaken. The flow process of SRERM development is shown in Figure 1.

B. THE STRUCTURE OF SRERM

We develop the SRERM structure on the following four dimensions:

- Levels of SRERM
- Security components (SCs) for each level.

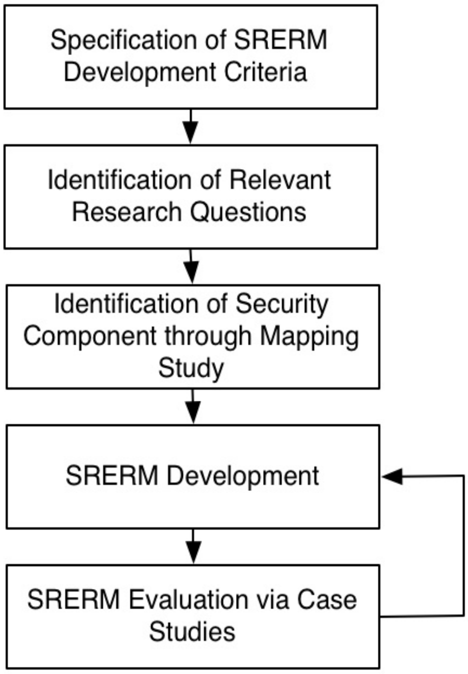
Specification of SRERM Development Criteria
Identification of Relevant Research Questions
Identification of Security Component through Mapping Study
SRERM Development
SRERM Evaluation via Case Studies

FIGURE 1. SRERM development flow process.

- Practices for each SCs.
- Assessment method

The SRERM structure is based on the readiness model concepts from SPIRM, SOVRM, and SOPM [21], [43], [44]. We developed the SRERM to assist organizations in quantifying the readiness of their SRE activities. Figure 2 shows the relationship among levels, components, and the practices of SRERM. It represents how SRERM levels recognize an organization's performance and how the findings from the SMAPS feed into these levels.

1) THE LEVELS OF SRERM

Following are the five preliminary levels of the SRERM for software development organizations.

- Initial: This readiness level has a confused status. At this level, organizations are not prepared for security requirements engineering.
- Standard: This readiness level indicates concern for developing basic security requirements for software development. At this level, organizations realize security requirements in software development are mandatory.
- Protected: This level analyzes the security requirements related to information assets.
- Anticipated: At this level, prevention and greater awareness are emphasized.
- Monitored: This is the highest readiness level. At this level, organizations have a high focus on maintaining security requirements built at the previous level.

The readiness levels in the SRERM require evaluation and feedback so that an organization's SRE readiness can be analyzed. When a conflict among security requirements components is found, or some suggestions relating to the representation of the readiness model are received, correction and improvement should be rapidly undertaken. Figure 3 represents how SRERM levels recognize the organization's performance and how the findings from the SMAPS are distributed into the SRERM levels.

2) THE SECURITY COMPONENTS OF SRERM

The SPI readiness model [20] was used to distribute the critical success factors and the barriers to software process improvement to each level. The SOPM [21] utilized the critical success factor of the outsourcing relationship. This research follows these concepts in order to develop the levels of the SRERM, which distribute the security requirement components. The security requirements components in our research refer to the security requirements categories, which were collected through the SMAPS. These security requirements categories were presented by Donald [45] and discussed by Salini and Kanmani [16]. 76 SRE practices were also developed based on the outcomes of the SMAPS and RE activities. The detail of the SRE practices is available in appendix A.

Twelve security requirements (SR) categories were distributed into five preliminary readiness levels, as depicted in Table 1. The distribution of these security components was based on the prioritization, which was obtained from the SMAPS. Each level contains some security requirements categories except the initial one. A focus column is added in Table 1 to describe the motivation or situation of each readiness level.

Identification SR, authentication SR, and authorization SR were distributed in the basic level because they are mandatory for each system. Immunity SR, privacy SR, and integrity SR were placed in the protected level because they are suitable for protecting important assets. Physical protection SR, non-repudiation SR, and intrusion SR were placed in the anticipated level because they require high effort. System maintenance SR, secure auditing SR, and survivability SR were placed in the monitored level because the cost of these SRs is very high and are commonly purposed for the sustainability of the organizations' long-term goals.

3) PRACTICES FOR EACH SCs

We propose various practices for each security requirements category. In total, 76 practices were identified from the mapping study as shown in Table 1 and Appendix A.

4) ASSESSMENT METHOD

The Motorola assessment tool [41] is the measurement tool used in the SRERM. As shown in Table 2, it is utilized to assess the practices for each security requirements component. This tool has been used in CMMI [42], SOVRM [21] and SOPM [21] publications. The Motorola assessment tool requires three assessment aspects [41]:

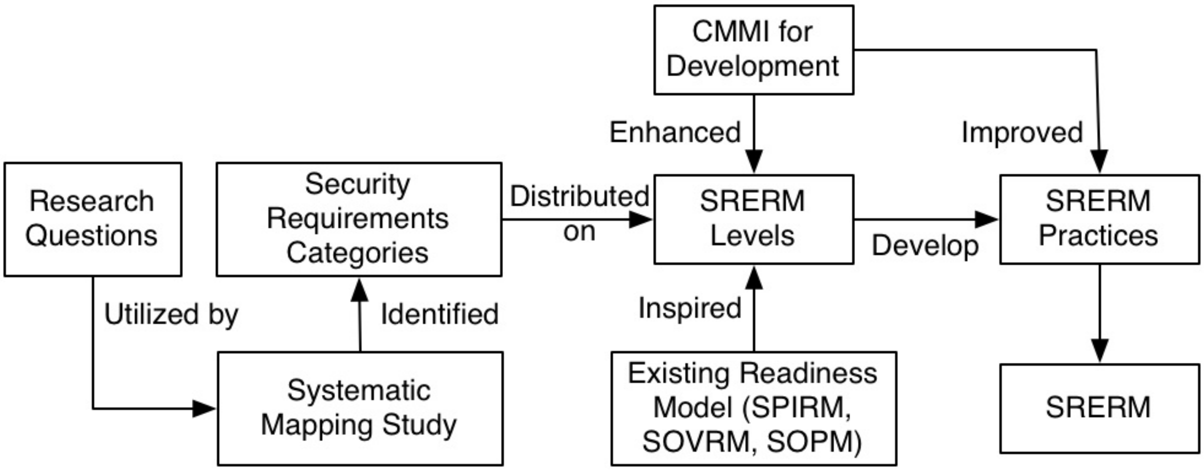

FIGURE 2. The structure of SRERM.

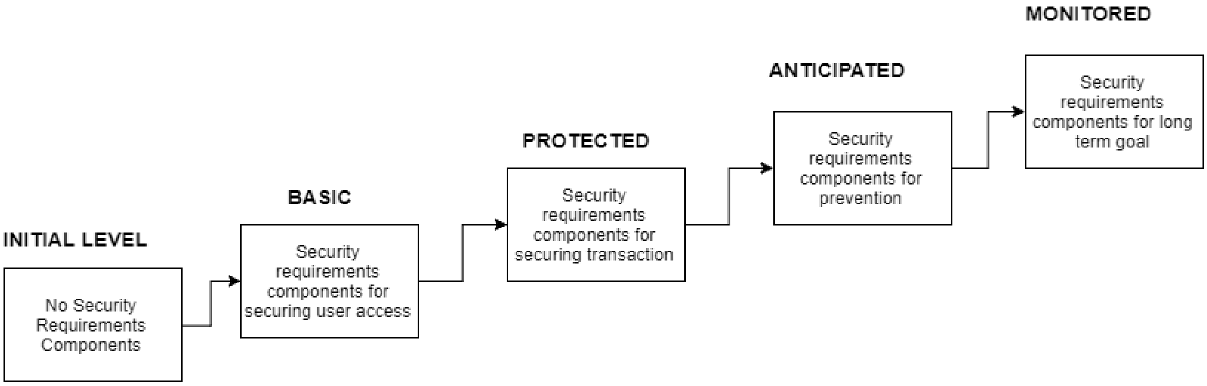
MONITORED
Security requirements components for long term goal
ANTICIPATED
Security requirements components for prevention
PROTECTED
Security requirements components for securing transaction
BASIC
Security requirements components for securing user access
INITIAL LEVEL
No Security Requirements Components

FIGURE 3. Preliminary SRERM Levels.

TABLE 1. Preliminary levels of the SRERM.

No	SRERM Level	Focus	Security Components	# Practices
1	Initial	No security requirement	Nil	-
2	Basic	Securing user's access	Identification Security Requirements	7
			Authentication Security Requirements	6
			Authorization Security Requirements	6
3	Protected	Securing transactions	Immunity Security Requirements	6
			Privacy Security Requirements	6
			Integrity Security Requirements	6
4	Anticipated	Prevention and high-quality security	Physical Protection Security Requirements	6
			Non-repudiation Security Requirements	6
			Intrusion Detection Security Requirements	6
5	Monitored	Security for long-term goals	System Maintenance Security Requirements	6
			Secure Auditing Security Requirements	7
			Survivability Security Requirements	8

- Approach: This aspect focuses on the support of management and the commitment of the organization relating to their practices.
- Deployment: This aspect focuses on the comprehensiveness and consistency of practice deployment.
- Results: This aspect focuses on the positive results in terms of the effect scale of the project.

For each aspect, we select a value (0, 2, 4, 6, 8, and 10) which can be determined by referring to the criteria provided in Table 2. Here, we explain how to utilize the Motorola

TABLE 2. Motorola assessment tool.

Score	Approach (A)	Deployment (D)	Results (R)
Poor (0)	• Management does not discover any need (OR) • Ability of organization is zero (OR) • Commitment of organizational is zero	• Practices are not utilized in the organization (OR) • Organization interest is zero	• Ineffective
Weak (2)	• The need is discovered by management (OR) • Support for practices starts (OR) • A few parts of the organization have the ability to undertake the practices	• Usage is fragmented (OR) • Usage is not appropriate (OR) • Implemented in several parts of the organization (OR) • Restricted to monitoring/verification of use	• Unstable result (OR) • Inappropriate result (OR) • Several parts of the organization gain a number of effectiveness outcomes.
Fair (4)	• Wide but not full commitment by management (OR) • Planning for practice establishment described (OR) • A number of supporting items for the practice exists	• Usage is fragmented (OR) • Some consistency in the usage (OR) • Implemented in a number of major parts of the organization (OR) • Usage in some parts of the organization is monitored/verified.	• Reliable and positive outcomes for a number of parts of the organization (OR) • Inappropriate outcomes for a number of parts of the organization
Marginally qualified (6)	• Some commitment from management is evident (OR) • Some of the management group becomes proactive (OR) • Practice deployment well underway across parts of the organization (OR) • Supporting items exist (OR)	• Some parts of the organization implement it (OR) • Suitable use across many parts of the organization (OR) • The usage of many parts of the organization is monitored or verified (OR)	• Positive measurable outcomes in many parts of the organization (OR) • Suitable positive results over time across many parts of the organization
Qualified (8)	• All management is committed (OR) • Almost all management is proactive (OR) • Practice developed as an integral part of the process (OR) • Supporting items motivate and improve the use of practice (OR)	• Implemented in almost all parts of the organization (OR) • Appropriate use across almost all parts of the organization (OR) • Usage for almost all parts of the organization monitored and verified.	• Almost all parts of the organization generate positive assessment outcomes (OR) • Appropriately positive outcomes over time across almost all parts of the organization
Outstanding (10)	• Management has a leadership commitment (OR) • Organizational excellence in practices identified not only within the organization	• General and stable deployed in all parts of the organization (OR) • Suitable use over time across all parts of the organization (OR) • All parts of the organization monitored and verified	• Requirements achieved (OR) • Suitable world-class outcomes (OR) • Guidance sought by others

TABLE 3. The example of security component evaluation.

No	Practices	Approach 0,2,4,6,8,10	Deployment 0,2,4,6,8,10	Result 0,2,4,6,8,10	Average
1	Utilize brainstorming technique to aggregate identification security requirement	10	10	10	10
2	Identify system stakeholders to improve identification security requirement	8	8	8	8
3	Plan for conflicts and conflict resolution for identification security requirement in terms of stakeholders	8	8	8	8
4	Define standard templates for describing identification security requirement	8	8	8	8
5	Use simple and concise language to explain identification security requirement	8	8	8	8
6	Check that identification security requirement meets your standard	8	8	8	8
7	Define change management policies for identification security requirement	4	4	4	4
Total of average scores (calculate all scores in the average column)					46
Final score (Total of average scores divided by number of practices) = 46/7 = 7.7					8

assessment tool by assuming the scores have been computed as shown in Table 3.

- First, for each practice, calculate the total score of the three aspects (approach, deployment, and result), then divide the total by three to find the average and round to a whole number.
- Second, repeat the first step for overall practice in one security component.
- Third, sum the average of every practice and divide by the number of practices for each security component.
- Fourth, repeat the third step and find the average for each level. If the level has an average score which is less than seven, it is regarded as weak, whereas a score equal to seven or higher is strong.

V. EVALUATION OF SRERM

The evaluation step for the SRERM is an important stage to validate and improve the applicability of the SRERM for the real software industry. Two case studies in the software industry were rigorously conducted. The respondents of the evaluation were selected once they were deemed to have the capability and experience to answer questions on the SRERM. In the evaluation agreement, it is also stated for privacy and business considerations that their affiliated organizations will not be published. The participants then completed both the SRERM and the feedback section.

Once the case studies were completed, the respondents were requested to complete the questionnaire to assess the quality of the SRERM. The evaluation criteria were explained in the evaluation criteria section. The outcomes of the SRERM evaluation were analyzed for identify weaknesses. The respondents suggested some changes to aid the future improvement of the model.

A. EVALUATION CRITERIA

The criteria in the feedback section are as follows:

- Ease of use: This criterion assesses and evaluates the usability of the SRERM structure. It requires the SRERM structure to have flexibility and be unambiguous because complex models will require a higher effort and training.
- Satisfaction of user: This criterion assesses and evaluates users' satisfaction according to the outcomes of the SRERM. They should be able to utilize the SRERM without any misunderstanding or difficulties to achieve the goals related to the SRE domain.
- Structure of the SRERM: This criterion's purpose is to identify any gaps in the SRERM structure and how to improve these.

B. CASE STUDIES

A case study has the ability to provide more information based on real-world perspectives. This advantage suits our requirement for the SRERM to be evaluated by practitioners in the software industry. We conducted a case study in this research for the following reasons:

- To demonstrate that the SRERM can be adapted to real software development.
- To highlight the areas where the SRERM requires improvement.
- To demonstrate the benefit of applying the SRERM.
- To achieve confidence in the evaluation, this research conducted two case studies in two different software development organizations. The selected organizations have clear software development processes. They also allowed the research to be released providing their identity was protected.

TABLE 4. The SCs implementation score of organization A.

SRERM Levels		Security Components	Organization A	
No	Level		Score	Status
1	Initial	Nil		
2	Basic	Identification Security Requirement	7.5	Implemented
		Authentication Security Requirement	8.2	Implemented
		Authorization Security Requirement	8.5	Implemented
3	Protected	Immunity Security Requirement	7.6	Implemented
		Privacy Security Requirement	7.1	Implemented
		Integrity Security Requirement	0.8	Not Implemented
4	Anticipated	Physical Protection Security Requirement	3.2	Not Implemented
		Non-repudiation Security Requirement	2.1	Not Implemented
		Intrusion Detection Security Requirement	5.1	Not Implemented
5	Monitored	System Maintenance Security Requirement	0	Not Implemented
		Secure Auditing Security Requirement	1	Not Implemented
		Survivability Security Requirement	0	Not Implemented

TABLE 5. The SCs implementation score of organization B.

SRERM Levels		Security Components	Organization B	
No	Level		Score	Status
1	Initial	Nil		
2	Basic	Identification Security Requirement	5.7	Not Implemented
		Authentication Security Requirement	8	Implemented
		Authorization Security Requirement	7.7	Implemented
3	Protected	Immunity Security Requirement	2.4	Not Implemented
		Privacy Security Requirement	4.4	Not Implemented
		Integrity Security Requirement	2.1	Not Implemented
4	Anticipated	Physical Protection Security Requirement	7.3	Implemented
		Non-repudiation Security Requirement	2.7	Not Implemented
		Intrusion Detection Security Requirement	1.3	Not Implemented
5	Monitored	System Maintenance Security Requirement	3.3	Not Implemented
		Secure Auditing Security Requirement	3.1	Not Implemented
		Survivability Security Requirement	4.7	Not Implemented

Initially, we personally communicated with each respondent from the different organizations, introducing the concept of the SRERM and inviting them to participate in our case study. Depending on the quality of the respondents' feedback, training and introductory discussion were carried out in the first instance. Although they were unfamiliar with security requirements engineering research, due to their knowledge of security mechanisms, they rapidly learned how to utilize the SRERM.

The outcomes of each organization assessment are shown in Table 4 and Table 5. Respondents was required to utilize their experience on completed projects to undertake the assessment. Due to quality concerns and independent feedback, the respondent was requested to complete the questionnaire at their place of business. In a short period, they submitted the assessment outcomes including the SRERM evaluation form via email.

1) ORGANIZATION 'A'

Organization A is a well-established software development organization with customers around the world. They have branches in Asia, Australia, Europe, and America. They support a number of oil companies by developing services such as real-time monitoring, data analytics, and reporting.

The respondent from organization A has 17 years of experience in developing and managing software. Currently, his position is a software development manager. One of his responsibilities is ensuring the requirements analysis process is implemented and satisfies the customer's need.

2) ASSESSMENT OUTCOMES OF ORGANIZATION 'A'

At the basic level, organization A obtains a score of 7.5 for identification SR, 8.2 for authentication SR and 8.5 for authorization SR. Therefore, organization A achieved the basic level because the scores are higher than 7. This indicates that

TABLE 6. Ease of learning evaluation of organization A and B.

Ease of Learning	Organizations' perception (n=2)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM representation is easy to learn	0	2	100%	0	0	0	0	0
A little knowledge of security requirements engineering is required to learn how to use SRERM	0	2	100%	0	0	0	0	0
It is necessary to learn the practices arranged for each security requirement component	0	2	100%	0	0	0	0	0
It is necessary understand the assessment method	0	2	100%	0	0	0	0	0
It is necessary to utilize the SRERM to measure an organization's readiness for security requirement engineering.	0	2	100%	0	0	0	0	0
It is necessary to distribute security requirement components among various levels, e.g. Identification, Authentication, and Authorization at the basic level	0	2	100%	0	0	0	0	0
Some training should be provided to assist the utilization of SRERM	0	2	100%	0	0	0	0	0

their security awareness is established from the earliest point of the software development lifecycle. At the protected level, they achieved a score higher than 7 in two SRs (immunity SR and privacy SR), which shows that they have an awareness of the threat of malware and of privacy issues. However, their integrity SR score is very low. The SCs at the monitored level are very low as they are still planning and discussing these concerns. This information was used to evaluate and improve the SRERM.

3) ORGANIZATION 'B'

Organization B is a growing software development organization of a university. They develop various integrated software such as a class registration system, an e-learning system, a payment system, a graduation system, an attendance system, a network settings system, and a library system. The main core of the organization's service is data center management and software development.

We selected this organization to represent a non-international organization with fewer employees. In addition, this organization has only one customer, which is the university. As a result, organization B was expected to add more usability value to the SRERM.

The selected respondent is a senior developer/team lead in organization B. He has more than five years' experience developing and managing systems. He plays a strong role in software development, especially the requirements analysis process. He has the responsibility to analyze customer's need and manage systems' development.

4) ASSESSMENT OUTCOMES OF ORGANIZATION 'B'

Organization B is in the initial level because they have not completed the security components at the basic level. For a growing organization, this level of achievement shows that their awareness of security requirements engineering has not yet started. They need more support and commitment from management to encourage their team to achieve a higher level of SRE readiness.

Based on Table 5, the results for organization B also indicate there is sufficient concern for physical protection SR and survivability SR. In contrast, they obtained a low score in identification SR. One of the possible reasons for this is because they only have a few stakeholders for their systems. As a result, they are able to describe the identification security requirements without completing all the practices provided in the SRERM. These findings were used to improve the SRERM.

5) FEEDBACK SUMMARY

Both respondents from organizations A and B completed the feedback forms to evaluate various aspects of the SRERM. As described in the section on SRERM development, there are three key aspects (ease of use, satisfaction of the user, and the structure of the SRERM). These were evaluated using quantitative measurement. In addition, questions were provided to collect the participants' reviews and suggestions for improving the SRERM.

First, they were asked to evaluate the ease of learning aspect. Based on Table 6, organizations A and B positively agreed that the SRERM is clear and easy to learn. However, training is still required to understand how to utilize the SRERM properly. Although the participants are familiar with the requirements engineering process and security mechanisms, they have only recently learnt about SRE.

Second, user satisfaction was assessed. As described in the evaluation criteria section, this criterion assesses and evaluates users' satisfaction corresponding to the results of the SRERM. As Table 7 shows, both organizations agreed that the SRERM could be useful in other organizations. They were interested in utilizing this SRERM in their work if it

TABLE 7. User satisfaction evaluation of organization A and B.

User Satisfaction	Organizations' perception (n=2)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM can be executed in most organizations	0	2	100%	0	0	0	0	0
Every practice is easy to learn and clear	0	2	100%	0	0	0	0	0
The SRERM is able to identify the strong and weak areas in organizations corresponding to security requirements engineering	0	2	100%	0	0	0	0	0
Using the SRERM would improve our security requirements engineering	0	2	100%	0	0	0	0	0
If SRERM were accessible in my occupation, I anticipate that I would utilize it.	0	2	100%	0	0	0	0	0
I agree with the readiness issues identified by SRERM.	0	2	100%	0	0	0	0	0
It is critical to actualize SRERM as an automated software tool to encourage security requirements engineering in measuring an organization's readiness.	0	2	100%	0	0	0	0	0

TABLE 8. SRERM structure evaluation of organization A and B.

Structure of SRERM	Organizations' perception (n=2)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
Every component of the SRERM is self-explanatory and requires no further clarification for adequate utilization	0	2	100%	0	0	0	0	0
Every component of the SRERM is feasible and suited to the security requirements engineering process	0	2	100%	0	0	0	0	0
The SRERM can be used effectively to identify security requirements engineering readiness issues with the goal of increasing an organization's readiness for security requirements engineering.	0	2	100%	0	0	0	0	0
The distribution of security components among various readiness levels (e.g. identification, authentication, and authorization) is valuable	0	2	100%	0	0	0	0	0
The five readiness levels of SRERM are valuable	0	2	100%	0	0	0	0	0

were available in their organizations. They were satisfied with the ability of the SRERM to recognize the area of their SRE, which needs further improvement.

Third, the structure aspect of the SRERM was evaluated by the two organizations, as shown in Table 8. They agreed that the arrangement of the SRERM structure is suitable and clear. The various levels and the distribution of the security requirements categories cause no confusion or ambiguity. Based on their evaluation outcomes, SRERM can be utilized to effectively measure the SRE readiness of software development organizations.

Fourth, we received a suggestion and criticism from organization A only. The respondent from organization A suggested a modification related to the levels of the SRERM, recommending that the SRERM have four levels instead of five. A criticism was made that the document needed improvement because the respondent had already utilized a requirements engineering template from JIRA. In addition, it was recommended that the design of the questionnaire be improved in the future. This feedback was utilized in the next section which discusses the modification of the SRERM.

VI. MODIFICATION TO SRERM

Based on the outcomes of the two case studies, we then made some modifications to the SRERM. The modifications were intended to ensure higher usability of the SRERM in the software industry. When the usability value of the SRERM is high, it will be easier to attract more software organizations to utilize the SRERM. The changes were related to moving the position of a security requirements component across levels and merging one level into another.

First, the physical protection security requirements component was moved from the anticipated level (third level) to the protected level (second level). This was motivated by the outcome of organization B that indicated high interest in securing the physical protection SR. They considered the

TABLE 9. Feedback results (essay answer) of organization A and B.

Question	Response		
	Organization A	Organization B	
Do you suggest any correction or enhancement to the SRERM?	Need to review the levels. For clarity, it would be better to merge the anticipated level with the monitored level	No	Positive
Do you suggest any new components for the SRERM in the future and if so, provide the reason?	No	No	Very Positive
What is your opinion on the assessment method?	Perhaps need to clearly define which documents are required during assessment	No	Positive
What is your opinion on the distribution of various security requirement practices?	No	No	Very Positive
What is your opinion on SRERM usability with respect to the time it takes the respondent to quantify security requirement engineering readiness?	Probably it is better to redesign the format when the parameters are the same for each level.	No	Positive
What is your opinion on the review of the organization's practices?	No need for changes	No changes suggested	Very Positive

physical properties of the server as important to the security of information. The physical server should be protected from any challenges such as theft, vandalism, fire, and natural disasters.

Second, due to the suggestion by the respondent of organization A, we merged the anticipated level with the monitored level. Therefore, non-repudiation SR and intrusion detection SR were distributed at the monitored level. We considered non-repudiation SR to be at the monitored level because it had the purpose of advancing the quality of software security. In addition, intrusion detection SR was considered to be at the monitored level due to its provision of high-quality security.

Finally, these modifications were updated in the SRERM as shown in Table 10 and Figure 4. The modifications were motivated by the assessment results obtained in the case studies of organizations A and B. The modifications also affected the practices for each security component without reducing its usability.

As shown in Table 10, the modified SRERM has four levels. The initial level indicates that the organization has no interest in implementing SRE in software development. The basic level indicates the organization has an awareness of SRE for mandatory security components. The protected level indicates the organization has a high concern to implement SRE to ensure the security of their data. Lastly, the monitored level indicates the organization is motivated to implement SRE by adding advanced services.

VII. CASE STUDIES AFTER MODIFICATION

We conducted additional case studies using organization A and C to evaluate the modified SRERM. As in the previous case studies, one of the anticipated advantages was to assess SRERM applicability and usability in real software organizations. The detailed explanation of the specific objectives and the outcomes are described in the following sections.

A. ASSESSMENT OUTCOMES OF CASE STUDY IN ORGANIZATION A

There are two main objectives of conducting these additional case studies. First, we tried to observe the improvement of SRE in this organization after six months. We compared the previous case study outcomes with the outcomes of the second case study. Second, we aimed to check the user satisfaction in organization A with the improved SRERM. Since we incorporated several recommendations from organization A to improve the SRERM, their reviews were essential in the second case study.

B. AN ADDITIONAL CASE STUDY IN ORGANIZATION A

There are two main objectives of conducting these additional case studies. First, we tried to observe the improvement of SRE in this organization after six months. We compared the previous case study outcomes with the outcomes of the second case study. Second, we aimed to check the user satisfaction in organization A with the improved SRERM. Since we incorporated several recommendations from organization A to improve the SRERM, their reviews were essential in the second case study.

As shown in Table 11, a comparison of the results of the previous case study with the second case study shows that

TABLE 10. The modified levels of SRERM.

No	SRERM Levels	Focus	Security Components
1	Initial	The situation without any security requirement component included	Nil
2	Basic	Securing user's access	Identification Security Requirement
			Authentication Security Requirement
			Authorization Security Requirement
3	Protected	Securing transactions and other important assets	Immunity Security Requirement
			Privacy Security Requirement
			Integrity Security Requirement
			Physical Protection Security Requirement
4	Monitored	Prevention, providing high-quality security, and supporting long-term goals	Survivability Security Requirement
			Intrusion Detection Security Requirement
			Non-repudiation Security Requirement
			Secure Auditing Security Requirement
			System Maintenance Security Requirement

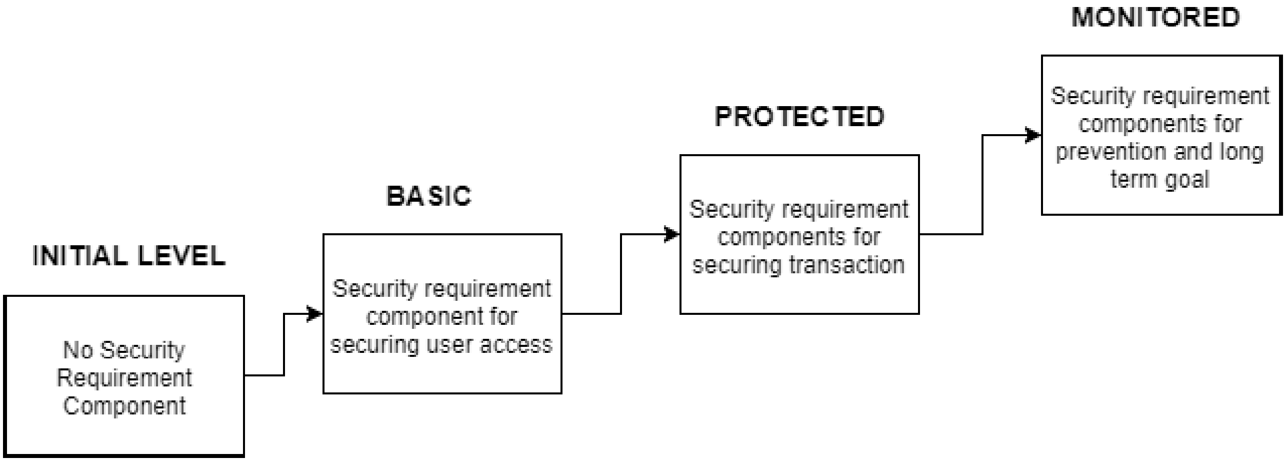
MONITORED
Security requirement components for prevention and long term goal
PROTECTED
Security requirement components for securing transaction
BASIC
Security requirement component for securing user access
INITIAL LEVEL
No Security Requirement Component

FIGURE 4. The modified levels of SRERM.

organization A has improved in some of the security requirements categories. For example, the integrity SR increased from 0.8 to 5.4. Although their SRE position is still at the basic level, this indicates there is an improvement in the organization. In addition, some security requirements categories have been improved such as physical protection, non-repudiation, intrusion detection, system maintenance, secure auditing, and survivability security requirements categories.

The respondent from organization A recognized and agreed with the modifications to SRERM, especially the modified levels. In the feedback section, he did not add any comments or suggestions regarding the modified SRERM. However, he agreed that the modified SRERM satisfied the ease of learning evaluation, user satisfaction evaluation, and structure evaluation.

C. AN ADDITIONAL CASE STUDY IN ORGANIZATION 'C'

We conducted a case study in organization C to evaluate the modified SRERM. Organization C has the main responsibility of providing IT services for a university in Saudi Arabia, with around 160 employees. They develop several education systems such as a student registration system, academic portal system, room booking system, and library system.

The selected respondent from organization C is a senior developer. He has more than five years' experience in software development. In addition, he conducted research in the field of software engineering when he was a graduate student at university. As a result, he was able to understand the purpose and the importance of the SRERM.

Similar to the previous case studies, we introduced the SRERM in a meeting with the respondent. The questionnaire was conducted online and the respondent's answers were collected. The questionnaire required an assessment of SRERM in terms of ease of learning, user satisfaction, structure evaluation and a feedback form.

Once the respondent had completed the questionnaire, their answers were extracted and analyzed based on the defined

TABLE 11. Implementation score for each SCs in organization A.

SRERM Level		Security Component	Organization A	
No.	Level		Previous Score	New Score
1.	Initial	Nil		
2.	Basic	Identification Security Requirements	7.5	7.5
		Authentication Security Requirements	8.2	8.2
		Authorization Security Requirements	8.5	8.5
3.	Protected	Immunity Security Requirements	7.6	7.6
		Privacy Security Requirements	7.1	7.1
		Integrity Security Requirements	0.8	5.4
		Physical Protection Security Requirements	3.2	4.3
5.	Monitored	Non-repudiation Security Requirements	2.1	4.2
		Intrusion Detection Security Requirements	5.1	6.1
		System Maintenance Security Requirements	0	2.3
		Secure Auditing Security Requirements	1	2.7
		Survivability Security Requirements	0	2.6

TABLE 12. Implementation score for each SCs in organization C.

SRERM Level		Security Component	Organization B	
No.	Level		Score	Status
1.	Initial	Nil		
2.	Basic	Identification Security Requirements	5.1	weak
		Authentication Security Requirements	4.7	weak
		Authorization Security Requirements	5.1	weak
3.	Protected	Immunity Security Requirements	8.0	strong
		Privacy Security Requirements	6.1	weak
		Integrity Security Requirements	5.3	weak
		Physical Protection Security Requirements	7.3	strong
4.	Monitored	Non-repudiation Security Requirements	2.3	weak
		Intrusion Detection Security Requirements	3.0	weak
		System Maintenance Security Requirements	4.0	weak
		Secure Auditing Security Requirements	4.6	weak
		Survivability Security Requirements	4.2	weak

evaluations. The following information describes the case study outcomes of respondent from organization C.

As shown in Table 12, organization C is at the initial level because it did not achieve a basic level score for the security requirement categories. However, they achieved a high score in two security requirements categories: immunity and physical protection. One reason why they had many low scores is due to the unspecified format of their security requirements. They did not provide a clear format or policy on how to manage security requirements in the software development process although they have a high concern about security requirements. They might improve their SRE readiness level by utilizing the template provided in several security requirements engineering frameworks.

The evaluation results of the SRERM are shown in Table 13, Table 14, and Table 15. The respondent from organization C strongly agreed that the form of SRERM is clear and easy to learn. He agreed with the arrangement of the SRERM structure, including the creation of levels and that the distribution of the security requirement categories was not confusing or ambiguous. In addition, SRERM can be utilized to effectively measure the SRE readiness of software development organizations. He agreed that the SRERM could be useful in other organizations. He left no suggestion or correction to the SRERM.

VIII. THREATS TO VALIDITY

This research has a few limitations regarding the outcomes of the SMAPS that was utilized to develop the SRERM. When selecting primary studies and extracting the data, subjective decisions may be made. A reason for this is that some primary studies do not have enough clear description, discussion and contribution. Consequently, we mitigated these limitations by

TABLE 13. Ease of learning evaluation of organization C.

Ease of Learning	Organization's perception (n=1)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM representation is clear	0	1	100%	0	0	0	0	0
A little knowledge of security requirements engineering is required to learn how to use SRERM	1	0	100%	0	0	0	0	0
It is necessary to learn the practices arranged for each security requirements component	0	1	100%	0	0	0	0	0
It is necessary to understand the assessment method	0	1	100%	0	0	0	0	0
It is necessary to utilize the SRERM to measure an organization's readiness for security requirements engineering.	0	1	100%	0	0	0	0	0
It is necessary to distribute security requirements components among various levels, e.g. Identification, Authentication, and Authorization at the basic level	0	1	100%	0	0	0	0	0
Some training should be provided for the utilization of SRERM	1	0	100%	0	0	0	0	0

TABLE 14. User satisfaction evaluation from organization C.

User Satisfaction	Organizations' perception (n=1)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM can be executed in most organizations	1	0	100%	0	0	0	0	0
Every practice is easy to learn and clear	0	1	100%	0	0	0	0	0
The SRERM is able to identify the strong and weak areas in organizations corresponding to security requirements engineering	1	0	100%	0	0	0	0	0
Using the SRERM would improve our security requirements engineering	1	0	100%	0	0	0	0	0
If SRERM were accessible for my occupation, I anticipate that I would utilize it.	1	0	100%	0	0	0	0	0
I agree with the readiness issues identified by SRERM.	1	0	100%	0	0	0	0	0
It is critical to actualize SRERM as an automated software tool to encourage SRE in measuring an organization's readiness.	0	1	100%	0	0	0	0	0

utilizing mapping study assistant software, and involved three other researchers to review the primary studies, undertaking an iterative selection process, and extracting the data comprehensively.

Another limitation is that this study retrieves publications from five electronic research databases only. Some relevant publications may exist in other research electronic databases that are not included in this research. Studies, which were published since this research was undertaken, could have been missed. Nevertheless, we believe our outcomes cover the most relevant published literature.

The case study to validate the proposed model was conducted using three organizations with different characteristics, which has external validity. Therefore, generalizing the findings on the SRERM into other organizations needs careful consideration.

TABLE 15. SRERM structure evaluation of organization C.

Structure of SRERM	Organizations’ perception (n=1)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
Every component of the SRERM is self-explanatory and requires no further clarification to be utilized adequately	0	1	100%	0	0	0	0	0
Every component of the SRERM is feasible and is suited to the security requirement engineering process	0	1	100%	0	0	0	0	0
The SRERM can be used effectively to identify SRE readiness issues with the goal of increasing an organization’s readiness for security requirement engineering.	0	1	100%	0	0	0	0	0
The distribution of security components among various readiness levels (e.g. identification, authentication, and authorization) is valuable	0	1	100%	0	0	0	0	0
The four readiness levels of SRERM are valuable	0	1	100%	0	0	0	0	0

TABLE 16. Identification security requirement practices.

No	Practices
1	Utilize brainstorming technique to aggregate identification security requirement
2	Identify system stakeholders to improve identification security requirement
3	Plan for conflicts and conflict resolution for identification security requirement in terms of stakeholders
4	Define standard templates for describing identification security requirement
5	Use simple and concise language to explain identification security requirement
6	Check that identification security requirement meets your standard
7	Define change management policies for identification security requirement

TABLE 17. Authentication security requirement practices.

No	Practices
1	Use scenarios to elicit authentication security requirement
2	Plan for conflicts and conflict resolution for authentication security requirement in terms of multiple accounts
3	Define standard templates for describing authentication security requirement
4	Use simple and concise language to explain authentication security requirement
5	Check that authentication security requirement meets your standard
6	Define change management policies for authentication security requirement

IX. CONCLUSION AND FUTURE WORK

In this study, we developed a security requirements engineering readiness model (SRERM). The purpose of the SRERM is to help organization measure their readiness level for SRE activities in software development organizations. The organizations are expected to be able to reduce their vulnerabilities

TABLE 18. Authorization security requirement practices.

No	Practices
1	Use scenarios to elicit the roles of stakeholders in terms of authorization security requirement
2	Plan for conflicts and conflict resolution for authorization security requirement in term of multiple roles
3	Define standard templates for describing authorization security requirement
4	Use simple and concise language to explain authorization security requirement
5	Check that authorization security requirement meets your standard
6	Define change management policies for authorization security requirement

TABLE 19. Immunity security requirement practices.

No	Practices
1	Define the system's operation environment to gain immunity security requirement
2	Assess immunity security requirement in terms of undesirable programs
3	Define standard templates for describing immunity security requirement
4	Use simple and concise language to explain immunity security requirement
5	Check that immunity security requirement meets your standard
6	Define change management policies for immunity security requirement

TABLE 20. Privacy security requirement practices.

No	Practices
1	Use scenarios to elicit sensitive data and communication in terms of privacy security requirement
2	Define the system boundaries in terms of privacy security requirement such as sensitive data and communication.
3	Define standard templates for describing privacy security requirement
4	Use simple and concise language to explain privacy security requirement
5	Check that privacy security requirement meets your standard
6	Define change management policies for privacy security requirement

TABLE 21. Integrity security requirement practices.

No	Practices
1	Define the operational processes to gain integrity security requirement
2	Assess integrity security requirement risks
3	Define standard templates for describing integrity security requirement
4	Use simple and concise language to explain integrity security requirement
5	Check that integrity security requirement meets your standard
6	Define change management policies for integrity security requirement

in terms of SRE in order to produce and maintain secure software.

A comprehensive systematic mapping study (SMAPS) was first conducted. In total, 104 primary studies were analyzed and the available evidence was used to identify the security requirement categories that are utilized in the SRERM development.

In order to assess the usability of the SRERM, two case studies were conducted before and after modification involving three software organizations. The outcomes of the first case study and the respondents' feedback motivated some modifications to the SRERM. The changes included moving the security requirements component from one level to another level and merging the anticipated level with the

TABLE 22. Physical protection security requirement practices.

No	Practices
1	Be sensitive to organizational and political considerations in gaining physical protection of security requirement
2	Assess physical protection requirement risks
3	Define standard templates for describing physical protection security requirement
4	Use simple and concise language to explain physical protection security requirement
5	Check that physical protection security requirement meets your standard
6	Define change management policies for physical protection security requirement

TABLE 23. Non-repudiation security requirement practices.

No	Practices
1	Define operational processes to gain non-repudiation security requirement
2	Plan for conflicts and conflict resolution in terms of non-repudiation security requirement
3	Define standard templates for describing non-repudiation security requirement
4	Use simple and concise language to explain non-repudiation security requirement
5	Check that non-repudiation security requirement meets your standard
6	Define change management policies for non-repudiation security requirement

TABLE 24. Intrusion detection security requirement practices.

No	Practices
1	Define operational processes to gain intrusion detection security requirement
2	Use interaction matrices to find conflicts and overlaps in terms of intrusion detection security requirement
3	Define standard templates for describing intrusion detection security requirement
4	Use simple and concise language to explain intrusion detection security requirement
5	Check that intrusion security requirement meets your standard
6	Define change management policies for intrusion detection security requirement

monitored level. Thus, the SRERM levels were reduced to four instead of five as initially designed. Each level contains various security components that are referred to as security requirement categories in the SMAPS. These changes can enhance the usability and applicability of the SRERM to assess the SRE process in other organizations. We used the Motorola assessment tool [41] to assess each practice in the SRERM. The calculated result for each practice defines the level of organizational readiness in terms of SRE.

To further enhance this work, SRERM needs to generate comprehensive outcomes due to different security mechanisms and facilities in various organizations. It needs more collaboration with several software development organizations. Some organizations that have security third parties or a large number of security experts will have a different priority for implementing SRE to growing organizations. Furthermore, we plan to extend SRERM with specific characteristics suited to various recent technologies, such as cloud computing and the Internet of Things (IoT). In addition, we plan

TABLE 25. System maintenance security requirement practices.

No	Practices
1	Use scenarios to elicit system maintenance security requirement
2	Define system boundaries in terms of system maintenance security requirement
3	Define standard templates for describing system maintenance security requirement
4	Use simple and concise language to explain system maintenance security requirement
5	Check that system maintenance security requirement meets your standard
6	Define change management policies for system maintenance security requirement

TABLE 26. Secure auditing security requirement practices.

No	Practices
1	Define operational processes in order to gain secure auditing requirement
2	Use checklists for secure auditing requirement
3	Assess security requirement risks to support secure auditing requirement
4	Define standard templates for describing secure auditing security requirement
5	Use simple and concise language to explain security auditing requirement
6	Check that security auditing requirement meets your standard
7	Define change management policies for security auditing requirement

TABLE 27. Survivability security requirement practices.

No	Practices
1	Define the system's operation environment to gain survivability security requirement
2	Assess system feasibility in terms of survivability security requirement
3	Plan for conflicts and conflict resolution in terms of survivability security requirement
4	Assess survivability security requirement risk
5	Define standard templates for describing survivability security requirement
6	Use simple and concise language to explain survivability security requirement
7	Check that survivability security requirement meets your standard
8	Define change management policies for survivability security requirement

to enhance SRERM to aid organizations in achieving ISO certification.

ACKNOWLEDGEMENT

The authors are grateful to the participants who evaluated the proposed model and recommended improvements.

APPENDIX
SECURITY REQUIREMENT ENGINEERING READINESS MODEL PRACTICES

See Tables 16–27.

REFERENCES

[1] G. McGraw, *Software Security Building Security In*. Boston, MA, USA: Pearson Education Ltd., 2006.

[2] R. Brown. (2014). *Computer Security Threats: A Brief History*. Accessed: Dec. 8, 2017. [Online]. Available: https://blog.dell.com/en-us/computer-security-threats-a-brief-history/

[3] C. Wong, *Security Metrics: A Beginner's Guide*. New York, NY, USA: McGraw-Hill, 2012.

[4] M. Rhodes-Ousley, *Information Security: The Complete Reference*, 2nd ed. New York, NY, USA: McGraw-Hill, 2013.

[5] G. McGraw, "Software Security," *Datenschutz Datensicherheit*, vol. 36, no. 9, pp. 662–665, 2012.

[6] M. Stamp, *Information Security: Principles and Practice*. Hoboken, NJ, USA: Wiley, 2011.

[7] I. El Kassmi and Z. Jarir, "Security requirements in Web service composition: Formalization, integration, and verification," in *Proc. Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Jun. 2016, pp. 179–184.

[8] R. Jindal, R. Malhotra, and A. Jain, "Automated classification of security requirements," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2016, pp. 2027–2033.

[9] N. Rjaibi and L. Ben Arfa Rabai, "Developing a novel holistic taxonomy of security requirements," *Procedia Comput. Sci.*, vol. 62, pp. 213–220, Mar. 2015.

[10] C. Gutiérrez, D. G. Rosado, and E. Fernández-medina, "The practical application of a process for eliciting and designing security in Web service systems," *Inf. Softw. Technol.*, vol. 51, no. 12, pp. 1712–1738, Dec. 2009.

[11] P. Salini and S. Kanmani, "Elicitation of security requirements for e-health system by applying model oriented security requirements engineering (MOSRE) framework," in *Proc. ACM 2nd Int. Conf. Comput. Sci. Eng. Inf. Technol. (CCSEIT)*, 2012, pp. 126–131.

[12] H. El-Hadary and S. El-Kassas, "Capturing security requirements for software systems," *J. Adv. Res.*, vol. 5, no. 4, pp. 463–472, Jul. 2014.

[13] K. Beckers, I. Côté, and L. Goeke, "A catalog of security requirements patterns for the domain of cloud computing systems," in *Proc. 29th Symp. Appl. Comput.*, 2014, pp. 337–342.

[14] I. Alqassem, "Privacy and security requirements framework for the Internet of Things (IoT)," in *Proc. 36th Int. Conf. Softw. Eng.*, 2014, pp. 739–741.

[15] C. B. Haley, "Arguing security: A framework for analyzing security requirements," Ph.D. dissertation, Open Univ., Milton Keynes, U.K., Mar. 2007.

[16] P. Salini and S. Kanmani, "Survey and analysis on security requirements engineering," *Comput. Elect. Eng.*, vol. 38, no. 6, pp. 1785–1797, Nov. 2012.

[17] P. Jaferian, G. Elahi, M. R. A. Shirazi, and B. Sadeghian, "RUPSec: Extending business modeling and requirements disciplines of RUP for developing secure systems," in *Proc. 31st EUROMICRO Conf. Softw. Eng. Adv. Appl.*, Aug./Sep. 2005, pp. 232–239.

[18] *Common Criteria for Information Technology Security Evaluation—Part 2: Security Functional Components*, Common Criteria Implement. Board, Sep. 2012.

[19] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*. Berlin, Germany: Springer-Verlag, 2010.

[20] M. Niazi, D. Wilson, and D. Zowghi, "Organisational readiness and software process improvement," in *Product-Focused Software Process Improvement* (Lecture Notes in Computer Science), vol. 4589. Berlin, Germany: Springer-Verlag, 2007, pp. 96–107.

[21] S. Ali and S. U. Khan, "Software outsourcing partnership model: An evaluation framework for vendor organizations," *J. Syst. Softw.*, vol. 117, pp. 402–425, Jul. 2016.

[22] L. Lin, B. Nuseibeh, D. Ince, and M. Jackson, "Using abuse frames to bound the scope of security problems," in *Proc. 12th IEEE Int. Requirements Eng. Conf.*, Sep. 2004, pp. 354–355.

[23] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett, "Introducing abuse frames for analysing security requirements," in *Proc. 11th Int. Conf. Requirements Eng. Conf.*, Sep. 2003, pp. 371–372.

[24] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Eng.*, vol. 10, no. 1, pp. 34–44, Jan. 2005.

[25] I. A. Tøndel, J. Jensen, and L. Røstad, "Combining misuse cases with attack trees and security activity models," in *Proc. 5th Int. Conf. Availability, Reliab. Secur.*, 2010, pp. 438–445.

[26] B. Schneier, "Attack trees," *Dr. Dobb's J.*, vol. 24, no. 12, pp. 21–23, 1999.

[27] V. Gandotra, A. Singhal, and P. Bedi, "Identifying security requirements hybrid technique," in *Proc. 4th Int. Conf. Softw. Eng. Adv. (ICSEA)*, 2009, pp. 407–412.

[28] J. McDermott and C. Fox, "Using abuse case models for security requirements analysis," in *Proc. 15th Annu. Comput. Secur. Appl. Conf.*, 1999, pp. 55–64.

[29] F. Dalpiaz, E. Paja, and P. Giorgini, "Security requirements engineering via commitments," in *Proc. 1st Workshop Socio-Tech. Aspects Secur. Trust (STAST)*, Sep. 2011, pp. 1–8.

[30] M. Q. Saleem, J. Jaafar, and M. F. Hassan, "Model driven security framework for definition of security requirements for SOA based applications," in *Proc. Int. Conf. Comput. Appl. Ind. Electron. (ICCAIE)*, Dec. 2010, pp. 266–270.

[31] D. Mellado, E. Fernández-Medina, and M. Piattini, "Security requirements engineering framework for software product lines," *Inf. Softw. Technol.*, vol. 52, no. 10, pp. 1094–1117, Oct. 2010.

[32] M. S. Ware, J. B. Bowles, and C. M. Eastman, "Using the common criteria to elicit security requirements with use cases," in *Proc. SoutheastCon*, Mar./Apr. 2005, pp. 273–278.

[33] D. Mellado, H. Mouratidis, and E. Fernández-Medina, "Secure Tropos framework for software product lines requirements engineering," *Comput. Standards Interfaces*, vol. 36, no. 4, pp. 711–722, Jun. 2014.

[34] M. Riaz, J. Stallings, M. P. Singh, J. Slankas, and L. Williams, "DIGS: A framework for discovering goals for security requirements engineering," in *Proc. Emprical Softw. Eng. Meas.*, Sep. 2016, Art. no. 35.

[35] K. Beckers, M. Heisel, I. Cote, L. Goeke, and S. Guler, "Structured pattern-based security requirements elicitation for clouds," in *Proc. Int. Conf. Availability, Rel. Secur. (ARES)*, Sep. 2013, pp. 465–474.

[36] A. Akinbi and E. Pereira, "Mapping security requirements to identify critical security areas of focus in PaaS cloud models," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.; Ubiquitous Comput. Commun.; Dependable, Auton. Secure Comput.; Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM)*, Oct. 2015, pp. 789–794.

[37] S. Islam, M. Ouedraogo, C. Kalloniatis, H. Mouratidis, and S. Gritzalis, "Assurance of security and privacy requirements for cloud deployment model," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2015.2511719.

[38] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. 12th Int. Conf. Eval. Assess. Softw. Eng. (EASE)*, 2008, pp. 68–77.

[39] S. Khan, M. Niazi, and R. Ahmad, "A readiness model for software development outsourcing vendors," in *Proc. IEEE Int. Conf. Global Softw. Eng.*, Aug. 2008, pp. 273–277.

[40] C. Wu, "A readiness model for adopting Web services," *J. Enterprise Inf. Manag.*, vol. 17, no. 5, pp. 361–371, 2004.

[41] M. K. Daskalantonakis, "Achieving higher SEI levels," *IEEE Softw.*, vol. 11, no. 4, pp. 17–24, Jul. 1994.

[42] S. U. Khan, M. Niazi, and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: A systematic literature review," in *Proc. IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Jul. 2009, pp. 207–216.

[43] S. Khan, M. Niazi, and R. Ahmad, "A readiness model for software development outsourcing vendors," in *Proc. Global Softw. Eng. Conf.*, Aug. 2008, pp. 273–277.

[44] M. Niazi, D. Wilson, and D. Zowghi, "A framework for assisting the design of effective software process improvement implementation strategies," *J. Syst. Softw.*, vol. 78, no. 2, pp. 204–222, Nov. 2005.

[45] F. Donald, "Engineering security requirements," *J. Object Technol.*, vol. 2, no. 1, pp. 53–68, 2003.

YUSUF MUFTI received the M.S. degree in software engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2017. His research interests include empirical software engineering and evidence-based software engineering.

MAHMOOD NIAZI is currently an Associate Professor of software engineering with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia. He has spent over a decade with leading technology firms and universities as a process analyst, a senior systems analyst, a project manager, a lecturer, and a professor. He has participated in and managed several software development projects. He has published over 100 articles in peer-reviewed conferences and journals. His research interests include evidence-based software engineering, requirements engineering, sustainable, reliable, and secure software engineering processes, global system development and management, project management, and software process improvement. His work has received over 3000 citations and has received awards for best papers at several conferences.

MOHAMMAD ALSHAYEB received the B.S. degree in computer science from Mutah University, Jordan, in 1995, and the M.S. and Ph.D. degrees in computer science and a certificate of software engineering from the University of Alabama in Huntsville in 2000, 2002, and 1999 respectively. He was a senior researcher and a software engineer, and managed software projects in USA and the Middle East. He taught and coordinated industrial training courses. He provided consulting services to major industrial and educational institutes. He is currently an Associate Professor with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. His research interests include software quality and quality improvements, software measurement and metrics, object-oriented design, and empirical studies in software engineering. He is a member of a number of professional associations. He is a certified Project Manager (PMP). He received a number of certificates of excellence and appreciation from many companies. He also received the Excellence in Teaching Award from KFUPM, in 2007, the Excellence in Advising Award in 2008, the Instructional Technology Award in 2012 and 2017, the Excellence in Research Award in 2014, and the Khalifa Award for Education as a Distinguished University Professor in the field of teaching within Arab World in 2016.

SAJJAD MAHMOOD received the Ph.D. degree from La Trobe University, Melbourne, Australia, in 2008. Prior to pursing his Ph.D. degree, he was a Software Engineer in USA and Australia. He taught and designed a number of courses related to software engineering at the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. He is currently an Associate Professor of software engineering with the Information and Computer Science Department, KFUPM. He is an active researcher in the field of software engineering and has published over 45 articles in peer-reviewed journals and international conferences. He was a principal and co-investigators in a number of research projects that investigate issues related to component-based software development and global software development projects. His research interests include empirical software engineering, evidence-based software engineering, component-based systems, global software development, and software process improvement in general. He received the Excellence in Teaching Award, the Excellence in Instructional Technology Award, and the Excellence in Academic Advising Award from KFUPM.