Evaluation of Requirement Engineering Best Practices for Secure Software Development in GSD: An ISM Analysis

Rafiq Ahmad Khan[1], Muhammad Azeem Akbar[2], Saima Rafi[3], Alaa Omran Almagrabi[4], and Musaad Alzahrani[5]

[1]University of Malakand
[2]LUT University
[3]University of Murcia
[4]King Abdulaziz University
[5]Albaha University

January 19, 2023

Abstract

Technological advancement makes the world a global village. The immense use of software systems has modernized human society in every aspect. Thus, the security parameter is an important element that needs to be considered while developing software systems. Considering the significance of software security, it is important to consider the security practices from the early phase of the software development life cycle (SDLC), i.e., requirements engineering (RE). Hence, this study aims to identify and categorize RE practices important to apply for secure software development (SSD) in a geographically distributed development environment. To study the RE practices concerning SSD, we conducted a questionnaire survey with industrial experts in the global software development (GSD) context. Furthermore, the interpretive structure modeling (ISM) approach was applied to evaluate the relationship between the RE security practice core categories. This paper identifies 70 practices and classifies them into 11 fundamental dimensions (categories) to assist GSD organizations in specifying the requirements for SSD. The ISM results show the "Awareness of Secure Requirement Engineering (SRE)" category has the most decisive influence on the other ten core categories of the identified RE security practices. With the help of empirical evidence and the ISM approach, this work attempts to identify potential security practices and to give a set of secure RE practices that can be used to improve the security of the software development process.

1. Introduction

In recent years, the software has become an important and integrated part of our daily activities. Software security has gained importance in research due to the increasing popularity of hacking and attacking software systems. Software security flaws and vulnerabilities result from badly written software that hackers can easily exploit. Most software is designed and put into use without considering security needs [1]. The majority of companies consider security to be a post-development process [2]. Every day, new threats from inside and outside the company threaten the availability and integrity of the company's data, resulting in massive financial loss and other damage [3].

Integrating security into the software engineering paradigm is essential to secure the software development life cycle from its early stages [4]. Therefore, many researchers have considered security from the outset of software development, starting with requirement engineering (RE) [5]. The development process needs to shape its security properties by adding security practices to avoid defects in software products [6]. Four stages must be followed to build secure software: Security protocol design, implementation, and Testing for complete software security needs [7]. This process aims to improve security requirements, apply threat modeling during software design, and follow best security practices when developing, reviewing code, and Testing [8]. This process needs to be updated all the time to make sure that software products are safe. Research is needed to discover what methods, notations, tools, and techniques are becoming popular [9]. Vulnerabilities are often caused by neglecting security [10]. The ”fix and penetrate” method, where security is checked after a project is finished, is used by even the most ethical companies [10].

Multiple efforts have been made to design, develop, and maintain secure software systems: Verdon and McGraw [11] designed Microsoft Trustworthy Computing Security Development Lifecycle [12], TSP Secure (Team Software Process for Secure Software Development) [13], Secure Software Development Process Model (S2D-ProM) [14]. Niazi et al. [10] developed the Requirements Engineering Security Maturity Model (RESMM), Comprehensive, Lightweight Application Security Process (CLASP) [15], and Secure Software Development Model (SSDM)education [16]. Al-Matouq et al. [17] designed a Secure Software Design Maturity Model (SSDMM), etc.

The above discussion shows that software security must be improved from the start. Integrating security awareness into the SDLC in the RE stage is a current research topic that needs to be implemented in the real-world software business [10]. The literature findings reveal that little work has been performed on SRE, and no work has been published that uses the Interpretive Structure Modeling (ISM) approach to categorize and find the interrelationship between RE practices for SSD in the context of GSD. Therefore, there is a dire need to study:

State-of-art on software security in the context of secure requirement engineering (SRE).

RE security practices to assist global software development (GSD) organizations in specifying the requirements for secure software development (SSD).

To find the interrelationship between the categories of RE security practices by applying Interpretive Structure Modeling (ISM).

The following research questions were designed to achieve the goals of this research.

RQ1: What software security practices are required to assist GSD organizations in specifying the requirements for SSD processes?

RQ2: What would be the interrelationship among the RE security practices that will assist GSD organizations in better managing SSD activities?

The remaining paper is structured as follows: Section 2 covers the background and related work, whereas Section 3 covers the research methods for this study. Section 4 presents all the results in detail, while Section 5 presents a summary, implications, and future work. Section 6 presents the limitations of the research.

2. Background and Related work

The need to take security requirements into account during system design and modeling has arisen as a result of the security concerns encountered in the connected world. Large corporations have lost millions of dollars due to security breaches, and this cost is rising [18]. According to Khan et al. [4], early security requirements analysis can cut software development and maintenance costs by 12-21%. Previous studies by academia, industry, and standards groups have proposed solutions to these issues [7, 19]. Early capturing of privacy and security requirements is critical to building public trust and promoting secure software development [5].

People around the globe are performing transactions through various channels such as the Internet, ATMs, mobile phone, and email. They make use of software, keeping in mind that it is dependable and trustworthy and that the operations are safe. However, still, due to budget constraints and the demand to deliver software to market quickly, as competition with other brands, many developers treat security as an afterthought, resulting in poor software quality [20].

Mead et al. [21] proposed a method, i.e., Security Quality Requirement Engineering (SQUARE), that elicits and documents security requirements. Goel et al. [22] recommend using Security-Requirements-Elicitation-

and-Assessment-Mechanism (SecREAM) to address security vulnerabilities early in software development. According to Mouratidis et al. [23], the 'Secure Tropos Methodology' can be used to create a unified security protection process that takes into account both the character, purpose, and planning concepts of requirement engineering and the threat, security challenges, and security strategy elements of security engineering. According to Manico [15], OWASP (Security Verification Standard (ASVS) version3.0) is a community effort to provide a framework of security criteria and controls that normalize the functional and nonfunctional security controls needed for designing, creating, and testing modern web applications. The Application Security Verification Standard (ASVS) is a set of requirements or tests used by architects, developers, testers, security professionals, and even users to determine whether or not a given application meets their standards for security [15]. Security requirements identification based on the context of usage, modeling, and risk analysis are the cornerstones of the AEGIS approach proposed by Ivan et al. [24] to build trustworthy systems. Chatterjee et al. [28] discussed "Secure Requirement Engineering (SRE) in terms of the nonfunctional requirement that elicits a control, constraint, safeguard or countermeasure to avoid or remove security vulnerabilities from requirements, design or code." The goal of SRE is to ensure the highest level of protection possible by putting into place all of the necessary security measures that will ensure privacy, integrity, and accessibility [25]. SRE is typically carried out at the SDLC's initial phase, and its successful completion results in a higher-quality software product. The core RE security activities is identification and inception, documentation, elicitation, analysis and negotiation, mapping, verification and validation, prioritizing and management, authentication, and authorization [26].

From the above discussion, we concluded that security must be added to the early stage of the SDLC to make secure software applications. However, there are limited studies conducted on integrating security in the RE phase of the SDLC. Furthermore, we found there is a dire need to study the interrelationship between security best practices in Requirement Engineering (RE) against security risks in the RE phase of the SDLC. To address the research gap, there is a need for an empirical study examining security practices in RE to assist GSD organizations in securing software development processes.

3. Research Methodology

The following research methods have been investigated to meet the study goals:

Step 1: To investigate SRE strategies to help GSD organizations and practitioners manage SSD RE operations. Thereby, we performed a systematic literature review (SLR) approach and the preliminary findings of our published prior work [4].

Step 2: A questionnaire survey to verify and validate the categorization of security best practices of RE for GSD as identified through SLR [4].

Step 3: We have used the ISM technique to assess how categorizing security best practices of RE will help GSD organizations during the SSD and provide a thorough picture of their interaction. The research methodology used in this paper is illustrated in Figure 1.

3.1 Step 1: Identifying RE best Security Practices for GSD Organizations

We used a systematic literature review (SLR) approach to investigate current RE security best practices for GSD, and our initial findings were recently presented at a conference [27]. The current paper extends our previous published study, exploring the RE security practices in the GSD context with industrial practitioners. We performed the following steps to perform the empirical study [4].

3.2 Step 2: Questionnaire Survey Design and Execution

In software engineering, the empirical approaches that are used most frequently are case studies, questionnaire surveys, and experimental analyses [28]. To obtain industry practitioners' opinions on SLR findings, this study used Google Docs to create an online questionnaire [27], identifying security risks and associated practices to mitigate these risks. Collecting data directly from industry experts in different parts of the world is hard. Therefore, we employed a non-methodical approach to data collection through a web-based questionnaire survey. Other software engineering researchers used the same data collection strategy [29-33]. Methods used in the questionnaire survey were as follows:

3.2.2.1 Development of Questionnaire Survey

The survey relies mostly on closed-ended questions to elicit information from professionals in the field. The questionnaire has a few open-ended questions to allow survey participants to add any additional software security risks and practices not identified by the SLR. The feedback was recorded in five-point Likert (strongly-agree to strongly-disagree).

3.2.2.2 Pilot of Questionnaire Survey

To conduct the pilot assessment of the questionnaire survey, we selected experts working in the global software development (GSD) environment (i.e., ”Software Engineering Research Group (SERG UOM) Pakistan,” ”King Fahd University of Petroleum and Minerals, Saudi Arabia,” and ”Qatar University, Doha, Qatar”). This pilot assessment addresses statistical variables and survey question clarity. To improve the questionnaire’s design, experts recommend including extra questions to collect more data from respondents.

A declaration of the researchers’ ethical obligation was included at the survey’s outset to ensure the participants’ anonymity. Participants were told that only the research team would access their data. The research team promised that they wouldn’t leak the data or divulge who the participants or companies were.

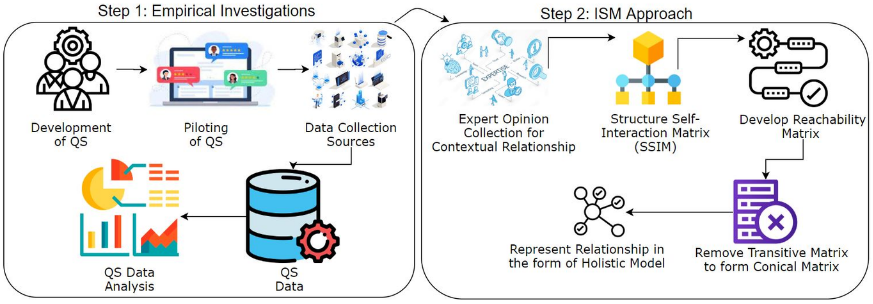

Step 1: Empirical Investigations
Development of QS
Piloting of QS
Data Collection Sources
QS Data Analysis
QS Data
Step 2: ISM Approach
Expert Opinion Collection for Contextual Relationship
Structure Self-Interaction Matrix (SSIM)
Develop Reachability Matrix
Represent Relationship in the form of Holistic Model
Remove Transitive Matrix to form Conical Matrix

Figure 1: Hybrid Research Methodology

3.2.2.3 Data Collection Sources

Our target audience was major global organizations. We resorted to a snowball sampling of relevant professionals [34, 35]. ”Snowballing” refers to a simple, inexpensive method of expanding one’s influence over a targeted group [32]. We contacted professionals via email and social media platforms like Facebook, LinkedIn, and Research Gate. The empirical data was collected online from April 2022 to July 2022. Acquiring all of the data took one month and four days to complete 64 replies were obtained throughout the survey’s deployment. The responses were manually reviewed, and 14 were disregarded because the information provided by their authors was unrelated to GSD and SSD. Final survey responses (n=50) were included in the study.

3.2.2.4 Data Analysis

This study used frequency analysis to analyze survey results [36].

3.3 Step-3 ISM Approach

Sage [37], in 1977, defined Interpretive Structure Modeling (ISM) as an approach that imposes order and direction on the complicated element and system relationships to create a holistic model. This is a dynamic method of learning that allows us to put together a comprehensive representation of the topic by connecting its many parts. The model uses clear patterns to show the structure’s complexity through graphs and words [38]. The ISM approach assists in finding various relationships in complicated situations when the relationship consists of different variables [39, 40]. Several scholars have used this method to construct a better conceptual framework for the system under examination [41-45]. Figure 2 explains the steps taken by the ISM methodology to determine the connection between RE security practices categorization.

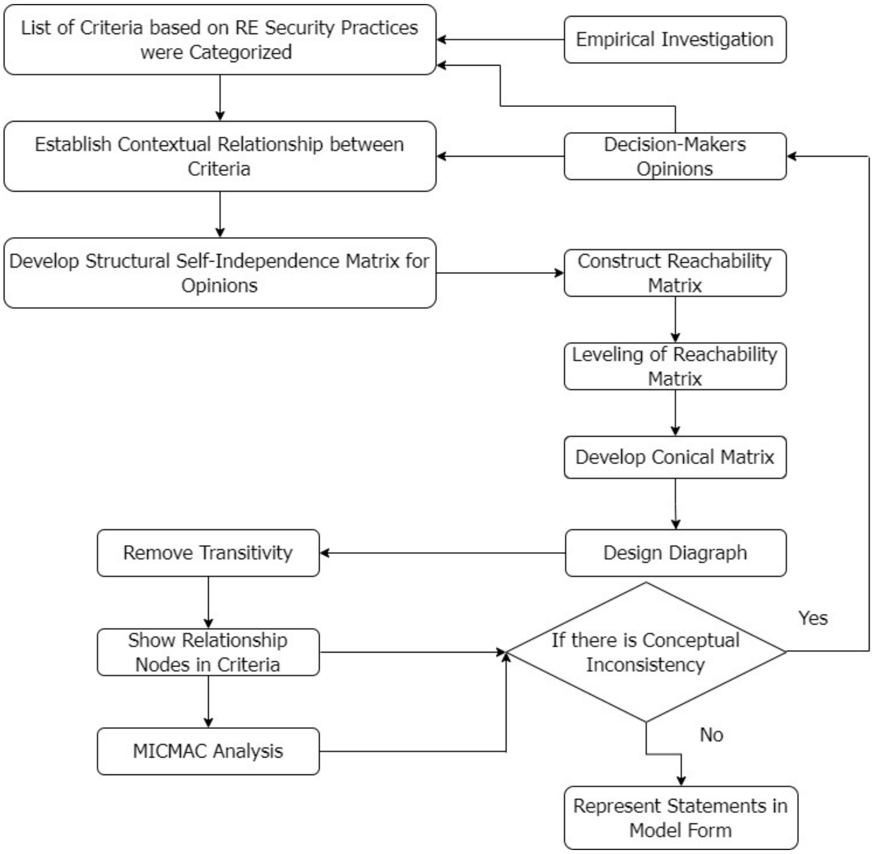

List of Criteria based on RE Security Practices were Categorized
Empirical Investigation
Establish Contextual Relationship between Criteria
Decision-Makers Opinions
Develop Structural Self-Independence Matrix for Opinions
Construct Reachability Matrix
Leveling of Reachability Matrix
Develop Conical Matrix
Remove Transitivity
Design Diagraph
Show Relationship Nodes in Criteria
If there is Conceptual Inconsistency
Yes
MICMAC Analysis
No
Represent Statements in Model Form

Figure 2: ISM Approach

4. Results and Discussion

The study's findings and analysis are presented in the following subsections:

4.1 Findings of SLR Study

A systematic Literature Review (SLR) is a step-by-step procedure that will help to identify the RE security risks and practices that need to be addressed to assist GSD organizations in secure software development (SSD). The identified RE security practices were then mapped into 11 fundamental categories of software security, as presented in Table 1. Later in this study (section 4.5), the mapping categories were used to create a comprehensive model of RE security practices for GSD organizations and their main categories.

To perform mapping, a coding scheme was used to put the RE security practices for GSD. The mapping scheme comprises three main categories: general categorization, sub-categorization, and theoretical framework [46]. Several studies that have been conducted in different areas of software engineering have taken into consideration these mapping approaches [47-49]. To ensure the accuracy of the mapping results, we conducted an inter-rater reliability test. We requested participants in the pilot assessment of the questionnaire survey study to perform the mapping process. We calculated the non-parametric Kendall's coefficient of concordance (W)[50] based on the mapping results of the study authors and external experts. Results (W=0.96) suggested agreement between study authors and external experts. As a result, this demonstrates that the process of mapping is both consistent and unbiased.

4.2 Findings of Empirical Study

The empirical investigation was carried out to gather the response from the experts working with RE security practices for the GSD organizations. The responses were collected through an online questionnaire using a five-point Likert scale. The respondents were asked to indicate their level of agreement using the following statements: ”Strongly Agree (SA),” ”Agree (A),” ”Strongly Disagree (SD)”, ”disagree (D)” and ”neutral (N)”.We divided the responses into three general categories: positive (defined as ”strongly agree and agree”), negative (defined as ”strongly disagree and disagree”), and ”neutral”. The summarized result of the positive category represents the participants in the survey. They agreed with the statement that the identified RE security practices could have a positive impact on the SSD. The survey results are presented in Table 1.

In the following table, ”RE1” means ”Requirement Engineering Practice Category 1 for GSD organizations in SSD process”, ”RE2” means ”Requirement Engineering Practice Category 2”, and so on up to ”RE11”. Similarly, ”P1” means ”Practice 1”. We categorized the identified 70 SRE practices into 11 fundamental categories, as depicted in Table 1. The survey findings present that the category ”SRE1: Awareness of SRE” is the most cited category in the identified practices list, with a percentage of 84. Requirements are gathered in a number of different ways, including through interviews, focus groups, and brainstorming sessions. SRE is distinct in that it strives to ensure full security by enforcing the three pillars of information security—namely, confidentiality, integrity, and availability [25].

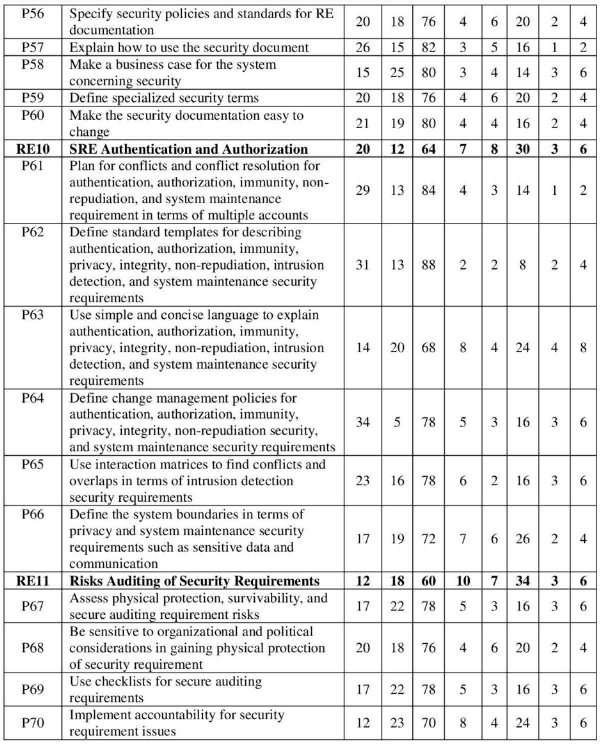
The importance of security requirements in secure software engineering cannot be overstated. The generally used best practices for handling security risks at the requirement engineering stage of the SDLC are listed in Table 1. The survey respondents identified that these practices assist global software development (GSD) organizations in SSD processes.

Table 1 presents that the most common security requirement engineering (SRE) practices are: well-defined client roles and resource capabilities, abuse and misuse cases, record rationale for security requirements, perform security requirements specification, and define standard templates for describing authentication, authorization, immunity, privacy, integrity, non-repudiation, intrusion detection, and system maintenance security requirements. The SQUARE (Security Quality Requirements Engineering) technique enables the elicitation, classification, and prioritizing of security standards for IT systems and applications [51]. Various researchers [10, 26, 52] and the relevance of including SRE in the SSD process have stressed GSD industry practitioners. These operations yield outcomes that are inextricably tied to the software's economic value [53].

Table 1. SRE Practices for GSD in SSD process identified through Empirical Study

Codes	**List of SRE Categories and its Practices**	**Empirical Investigation (N=50)**							
		Positive (N=50)			**Negative (N=50)**			**Neutral (N=50)**	
		SA	**A**	**%**	**SD**	**D**	**%**	**N**	**%**
RE1	**Awareness of SRE**	**34**	**8**	**84**	**4**	**2**	**12**	**2**	**4**
P1	Adopt security and privacy risk assessment policies	30	10	80	5	4	18	1	2
P2	Requirements engineering team members have adequate security training	21	19	80	4	4	16	2	4
P3	Update Requirements Repository	26	14	80	6	3	18	1	2
P4	Adopt established standards for secure RE	17	25	84	4	1	10	3	6
P5	Well-defined client roles and resource capabilities	31	13	88	2	2	8	2	4
P6	Identify requirement dependencies	15	25	80	3	3	12	4	8
P7	Develop artifacts to examine the possible security risk of RE	20	18	76	4	6	20	2	4
P8	Create quality requirements gates/bug bars	14	20	68	8	5	26	3	6
RE2	**SRE Methods and Tools**	**26**	**15**	**82**	**3**	**5**	**16**	**1**	**2**
P9	UMLsec, SecureUML	17	25	84	5	1	12	2	4
P10	Secure Tropos	20	14	68	8	4	24	4	8
P11	Abuse and Misuse Cases	31	13	88	2	3	10	1	2
P12	Structured Object-Oriented Formal Languages	21	19	80	4	4	16	2	4
P13	Secure Machine learning Techniques	32	10	84	2	3	10	3	6
P14	Secure RE Approach	15	25	80	3	3	12	4	8
P15	Secure RE Problem Frames	24	16	80	6	2	16	2	4
RE3	**Elicitation of Secure Requirements**	**12**	**28**	**80**	**5**	**4**	**18**	**1**	**2**
P16	Elicit and categorize safety and security requirements	34	5	78	5	3	16	3	6
P17	Take into consideration organizational and political issues	29	13	84	4	3	14	1	2
P18	Use scenarios to elicit sensitive data and communication	14	20	68	8	4	24	4	8
P19	Use concerns related to business to motivate security requirement elicitation	15	25	80	3	4	14	3	6
P20	Identify functional and non-functional security requirements	20	18	76	4	6	20	2	4
P21	Record rationale for security requirements	31	13	88	2	2	8	2	4
P22	Use hypothetical cases to elicit security requirement	21	19	80	4	4	16	2	4
P23	Assess system security feasibility in terms of RE	17	22	78	5	3	16	3	6
RE4	**Analysis and Negotiation of Secure Requirements**	**23**	**16**	**78**	**6**	**2**	**16**	**3**	**6**
P24	Conduct product security risk analysis to ensure security requirements and constraints	15	25	80	3	3	12	4	8
P25	Adopt attack tree modeling	24	16	80	6	2	16	2	4
P26	Analyze tradeoffs for security requirements between cost and protection	29	13	84	4	3	14	1	2
P27	Perform threat landscaping	14	20	68	8	4	24	4	8
P28	Define security of system boundaries	14	20	68	8	5	26	3	6
P29	Make use of checklists to analyze security requirements	26	15	82	3	5	16	1	2
P30	Sort out security requirements through a multi-dimensional approach	15	25	80	3	3	12	4	8

P31	Provide software to support negotiations	24	16	80	6	2	16	2	4
RE5	**SRE Threat Modeling**	**20**	**18**	**76**	**5**	**6**	**22**	**1**	**2**
P32	Identify threat origin with the help of threat modeling	21	19	80	4	4	16	2	4
P33	Implement STRIDE and DREAD Threat Model	26	14	80	6	3	18	1	2
P34	Perform threat evaluation & prioritization	20	18	76	4	6	20	2	4
P35	Illustrate threat landscaping, risk likelihood, and mitigation strategy for secure RE	14	20	68	8	5	26	3	6
RE6	**Verification & Validation of Security Requirements**	**20**	**17**	**74**	**5**	**4**	**18**	**4**	**8**
P36	Review documentation against the security objectives and needs	29	13	84	4	3	14	1	2
P37	Perform secure requirements review	15	25	80	3	3	12	4	8
P38	Software products and artifacts should be certified and verified	21	19	80	4	4	16	2	4
P39	Identification of attacker's interest and capabilities in the resources/assets of software	20	18	76	4	6	20	2	4
P40	A threshold can be defined by using the security index	21	16	74	6	5	22	1	2
P41	Use multi-disciplinary teams to assess security requirements	26	14	80	6	3	18	1	2
P42	Use prototype to animate security requirements	14	20	68	8	4	24	4	8
RE7	**Prioritization and Management of Security Requirements**	**17**	**19**	**72**	**7**	**6**	**26**	**2**	**4**
P43	Perform security requirements specification	31	13	88	2	2	8	2	4
P44	Identify policies for management of security requirements	17	25	84	4	1	10	3	6
P45	Conduct coherent and cost-effective manner for security prioritization	20	18	76	4	6	20	2	4
P46	Evaluate and manage product security risks throughout the project	30	10	80	5	4	18	1	2
P47	Preservation of confidentiality, Integrity, Availability, and Usability, should be specified to mitigate identified threats	32	10	84	2	3	10	3	6
P48	Establish and manage the project's secure development process	20	18	76	4	6	20	2	4
P49	Define and maintain traceability manual	17	22	78	5	3	16	3	6
P50	Record rejected security requirement	26	14	80	6	3	18	1	2
RE8	**Identification and Inception of Security Requirements**	**12**	**23**	**70**	**8**	**4**	**24**	**3**	**6**
P51	All stakeholders, customers, and clients need to be agreed on the security requirement definition	26	15	82	3	5	16	1	2
P52	Identification of security requirement goals and objectives	20	18	76	4	6	20	2	4
P53	Identification of potential attackers of the software	21	19	80	4	4	16	2	4
P54	Utilize brainstorming technique to aggregate identification of security requirement	15	25	80	3	4	14	3	6
RE9	**SRE Documentation**	**18**	**16**	**68**	**3**	**9**	**24**	**4**	**8**
P55	Incorporate security needs, objectives, and requirements in the final documentation	17	22	78	5	3	16	3	6



P56	Specify security policies and standards for RE documentation	20	18	76	4	6	20	2	4
P57	Explain how to use the security document	26	15	82	3	5	16	1	2
P58	Make a business case for the system concerning security	15	25	80	3	4	14	3	6
P59	Define specialized security terms	20	18	76	4	6	20	2	4
P60	Make the security documentation easy to change	21	19	80	4	4	16	2	4
RE10	**SRE Authentication and Authorization**	**20**	**12**	**64**	**7**	**8**	**30**	**3**	**6**
P61	Plan for conflicts and conflict resolution for authentication, authorization, immunity, non-repudiation, and system maintenance requirement in terms of multiple accounts	29	13	84	4	3	14	1	2
P62	Define standard templates for describing authentication, authorization, immunity, privacy, integrity, non-repudiation, intrusion detection, and system maintenance security requirements	31	13	88	2	2	8	2	4
P63	Use simple and concise language to explain authentication, authorization, immunity, privacy, integrity, non-repudiation, intrusion detection, and system maintenance security requirements	14	20	68	8	4	24	4	8
P64	Define change management policies for authentication, authorization, immunity, privacy, integrity, non-repudiation security, and system maintenance security requirements	34	5	78	5	3	16	3	6
P65	Use interaction matrices to find conflicts and overlaps in terms of intrusion detection security requirements	23	16	78	6	2	16	3	6
P66	Define the system boundaries in terms of privacy and system maintenance security requirements such as sensitive data and communication	17	19	72	7	6	26	2	4
RE11	**Risks Auditing of Security Requirements**	**12**	**18**	**60**	**10**	**7**	**34**	**3**	**6**
P67	Assess physical protection, survivability, and secure auditing requirement risks	17	22	78	5	3	16	3	6
P68	Be sensitive to organizational and political considerations in gaining physical protection of security requirement	20	18	76	4	6	20	2	4
P69	Use checklists for secure auditing requirements	17	22	78	5	3	16	3	6
P70	Implement accountability for security requirement issues	12	23	70	8	4	24	3	6

4.3 Results of the ISM Approach

Interaction between RE practices and the key knowledge areas is determined using the ISM method. Many academic researchers have used a similar methodology to investigate the contextual interaction of elements [40-43, 47]. ISM method helps in presenting a complex system in a simplified way, provides an interpretation of the embedded object, and transforms unclear and poorly articulated mental models of systems into visible, well-defined models, thereby helping in answering *what* and *how* in theory, building facilitates the identification of the structure within a system [40-43, 47]. A structural-self-interaction matrix (SSIM), as provided in the following sections, is required to develop the contextual interaction between the criteria.

4.3.1 Structural-Self-Interaction Matrix (SSIM)

The ISM methodology was applied using experts' perspectives to examine the contextual relationship between the RE security practice's core categories. We formed an expert group to gain their insight on ISM. An invitation letter was used to invite participants to the initial survey. Thirteen of the most knowledgeable individuals in the field offered to participate in the decision-making process. The participants come from various research and development sectors and industry practitioners. We built the SSIM matrix based on the opinions of the experts.

There is a possibility that the study's findings cannot be generalized due to the small sample size. However, we discovered that Kannan et al. [42] utilized the recommendations of five experts to choose reverse logistic providers. Similarly, Soni et al. [54] assembled a group of nine experts to investigate the aspects of an urban rail transit system that contributed to its complexity. Attri et al. [55] decided on the success elements for complete productive maintenance using the data that five experts provided. Azeem et al. [47] applied the ISM approach to study the relationships among the core categories of the challenges in DevSecOps. Similarly, other researchers also use the ISM approach to find the interrelationship between the DevOps testing process [41] and best test practices [40].

The following symbols indicate the direction of a relationship between a RE enabler (m and n) in the appropriate SSD context.

- The letter ”V” denotes the connection between the m and n enablers.
- The letter ”A” denotes the connection between the n and m enablers.
- ”X” when both enablers' m and n reach each other in the same direction.
- ”O” is the scenario that occurs when enabler m and enabler n do not have any connection to one another.

We have designed the SSIM shown in Table 2 based on the comments of industry professionals.

Table 2: SSIM Matrix

	RE11	RE10	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1
RE1	V	V	V	V	V	V	V	V	V	V	*
RE2	O	V	V	V	O	A	V	V	A	*	*
RE3	V	O	V	V	O	V	V	A	*	*	*
RE4	V	V	O	V	V	V	V	*	*	*	*
RE5	V	O	V	A	V	V	*	*	*	*	*
RE6	A	X	O	A	V	*	*	*	*	*	*
RE7	A	A	O	V	*	*	*	*	*	*	*
RE8	V	V	V	*	*	*	*	*	*	*	*
RE9	V	A	*	*	*	*	*	*	*	*	*
RE10	A	*	*	*	*	*	*	*	*	*	*
RE11	*	*	*	*	*	*	*	*	*	*	*

According to the results presented in Table 2, there is a relationship between RE1 ”Awareness of SRE” and RE11 ”Risks Auditing of Security Requirements”, as ”V” denotes the relationship between both the enablers. Hence RE1 helps to improve the RE11 by following practices (P67-P70). Also, the relationship between RE2 ”SRE Methods and Tools” and RE11 ”Risks Auditing of Security Requirements” is shown to be ”O” which means that there is no connection between the two. In addition, it has been observed that the RE2 ”SRE Methods and Tools” contributes to the RE6 ”Verification and Validation of Security Requirements” enhancement. This is because, in the opinions of the SSD experts, RE2 and RE6 have a relationship of the type ”A”. We have also noticed from Table 2 that RE6, ”Verification and Validation of Security Requirements,” and RE10, ”Authentication and Authorization of Security Requirements,” have the same direction because their relationship denotes a letter ”X”.

4.3.2 Reachability Matrix

We transformed V, A, X, and O for the reachability matrix in binary form (0, 1). The following protocols are taken into consideration in the development of the reachability matrix.

- If the value of m and n in SSIM is V, then we replace it with 1; if not, the value allocated is 0.
- If the value of m and n in SSIM is A, it is changed to 0; if not, it is changed to 1.
- If the value of m and n in SSIM is X, it is substituted with 1; and 1 is assigned to the m and n entries.
- If the value of m and n in the SSIM is O, it will be replaced with 0; the value allocated to m and n is 0.

Table 3 shows the reachability matrix based on the protocols mentioned above. The determination of the transitivity, discussed in Section 3.3, was used in developing the final reachability matrix. The transitivity is implemented using the 1* value. This removes the error in data collected for SSIM.

Table 3: Reachability Matrix

	RE1	RE2	RE3	RE4	RE5	RE6	RE7	RE8	RE9	RE10	RE11
RE1	1	1	1	1	1	1	1	1	1	1	1
RE2	0	1	0	1	1	0	0	1	1	1	0
RE3	0	1	1	0	1	1	0	1	1	0	1
RE4	0	0	1	1	1	1	1	1	0	1	1
RE5	0	0	0	0	1	1	1	0	1	0	1
RE6	0	1	0	0	0	1	0	1	0	1	0
RE7	0	0	0	0	0	0	1	1	0	0	0
RE8	0	0	0	0	1	1	1	1	1	1	1
RE9	0	0	0	0	0	0	0	0	1	0	1
RE10	0	0	0	0	0	1	1	0	1	1	0
RE11	0	0	0	0	0	1	1	0	0	1	1

Table 4 details the addition of a transitivity check and shows how each criterion drives, depends on, and ranks the others. The identified driving power shows all the requirements for that RE security practice category (criteria). The dependent power indicates the criteria that may aid in attaining the objective. This reliance and driving power will help in MICMAC analysis, which divides criteria into four clusters: autonomous, dependent, linking, and independent.

Table 4: Transitivity Check

	RE1	RE2	RE3	RE4	RE5	RE6	RE7	RE8	RE9	RE10	RE11	DRI	Rank
RE1	1	1	1	1	1	1	1	1	1	1	1	11	6
RE2	0	1	*1	1	1	*1	*1	1	1	1	*1	10	5
RE3	0	1	1	*1	1	1	*1	1	1	*1	1	10	5
RE4	0	*1	1	1	1	1	1	1	*1	1	1	10	5
E5	0	*1	0	0	1	1	1	*1	1	*1	1	8	3
RE6	0	1	0	*1	*1	1	*1	1	*1	1	*1	9	4
RE7	0	0	0	0	*1	*1	1	1	*1	*1	*1	7	2
RE8	0	*1	0	0	1	1	1	1	1	1	1	8	3
RE9	0	0	0	0	0	*1	*1	0	1	*1	1	5	1
RE10	0	*1	0	0	0	1	1	*1	1	1	*1	7	2
RE11	0	*1	0	0	0	1	1	*1	*1	1	1	7	2
Dep	1	9	4	5	8	11	11	10	11	11	11	92	
Rank	1	5	2	3	4	7	7	6	7	7	7		

4.3.3 Portioning the Reachability Matrix

According to Warfield [56], ”the reachability set for a particular variable consists of the variable itself and the other variables, which help the variable itself and to form reachability set.” After that, the intersection of these sets is computed for each of the components individually. The elements with the same reachability and intersection sets occupy the top level of the ISM hierarchy. The top-level element in the hierarchy does not contribute to completing any other levels above it. After the top-level element has been determined, it is isolated from the other elements in the structure. The same method is utilized once more to determine the components of the subsequent level. This procedure is repeated until the level of each element is figured out. These levels give support to the construction of the diagram and ISM model. Table 5 presents the reachability set, antecedent set, intersection set, and levels for this study’s 11 criteria (RE practices categories for GSD organizations in the SSD process).

Table 5: Levels of Final Reachability Matrix

Leveling of RE Practices Categories

	Reachability Set	Antecedent Set	Intersection set	Level	RE Practices Categories
Iteration 1	Iteration 1	Iteration 1	Iteration 1	Iteration 1	Iteration 1
RE1	1,2,3,4,5,6,7,8,9,10,11	1,10,11	1,10,11		
RE2	2,3,4,5,6,7,8,9,10,11	1,2,3,4,5,6,8,10,11	2,3,4,5,6,8,10,11		
RE3	2,3,4,5,6,7,8,9,10,11	1,2,3,4	2,3,4		
RE4	2,3,4,5,6,7,8,9,10,12	1,2,3,4,6	2,3,4,6		
RE5	2,5,6,7,8,9,10,11	1,2,3,4,5,6,7,8	2,5,6,7,8		
RE6	2,4,5,6,7,8,9,10,11	1,2,3,4,5,6,7,8,9,10,11	2,4,5,6,7,8,9,10,11	Level 1	2,4,5,6,7,8,9,10,11
RE7	5,6,7,8,9,10,11	1,2,3,4,5,6,7,8,9,10,11	5,6,7,8,9,10,11		
RE8	2,5,6,7,8,9,10	1,2,3,4,5,6,7,8,10,11	2,5,6,7,8,10		
RE9	6,7,9,10,11	1,2,3,4,5,6,7,8,9,10,11	6,7,9,10,11		
RE10	2,6,7,8,9,10	1,2,3,4,5,6,7,8,9,10,11	2,6,7,8,9,10		
RE11	2,6,7,8,9,10	1,2,3,4,5,6,7,8,9,10,11	2,6,7,8,9,10		
Iteration 2	Iteration 2	Iteration 2	Iteration 2	Iteration 2	Iteration 2
RE1	1,3	1	1		
RE2	3	3	3	Level 2	3
Iteration 3	Iteration 3	Iteration 3	Iteration 3	Iteration 3	Iteration 3
RE1	1	1	1	Level 3	1

4.3.4 Interpretation of ISM Model

The final version of the ISM model was developed based on the results of the reachability matrix. The interconnections between the criteria are illustrated by arrows that point from one criterion to another. After the digraph was successfully converted to the ISM model, the transitivity analysis was carried out to determine whether or not the data contained any ambiguity (see Figure 3). In this figure RE1 ”Awareness of SRE” category stands on the top for selecting RE practices for GSD organizations in the context of SSD. This shows that RE1 is an independent category in the identified list of RE practices for GSD organizations. All the other categories are dependent on RE1.

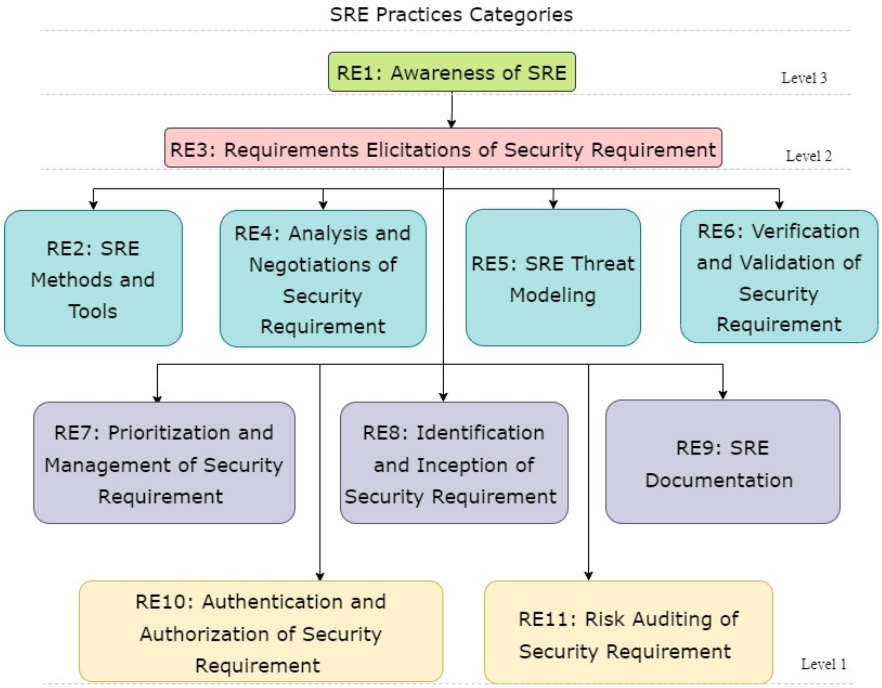

SRE Practices Categories
RE1: Awareness of SRE
Level 3
RE3: Requirements Elicitations of Security Requirement
Level 2
RE2: SRE Methods and Tools
RE4: Analysis and Negotiations of Security Requirement
RE5: SRE Threat Modeling
RE6: Verification and Validation of Security Requirement
RE7: Prioritization and Management of Security Requirement
RE8: Identification and Inception of Security Requirement
RE9: SRE Documentation
RE10: Authentication and Authorization of Security Requirement
RE11: Risk Auditing of Security Requirement
Level 1

Figure 3: Leveling of SRE Practices Categories

Figure 3 presents that RE3 ”Requirement Elicitations of Security Requirements” is dependent only on level 3 (RE1: Awareness of SRE), but all the coming categories (RE2, RE4-RE11) of level 1 depend on level 2 category (RE3). The findings of this figure depict that RE2, RE4-RE11 practices categories depend on RE3 and RE1. This indicates that to fulfill the need of RE2, RE4-RE11, all the practices associated with these categories must be implemented, including the categories on which they depend on them, i.e., RE3 and RE1.

4.3.5 MICMAC Analysis

MICMAC stands for matrix cross-impact matrix classification. The MICMAC analysis examines the system’s key categories. Attri et al. [55] say that the MICMAC ”analysis involves the development of a graph that classifies factors based on driving power and dependence power.” ”MICMAC analysis is used to classify the factors and validate the interpretive structural model factors in the study to reach their results and conclusions” [54]. Enablers are divided into four groups based on their driving and dependence power.

1. **Autonomous Cluster:** This cluster contains a category with low driving and dependency power. They are mostly disconnected due to weak links. As a result, their influence on the system as a whole is negligible [76].
2. **Linkage Cluster:** This cluster has great driving and dependency power and affects other enablers due to strong connectivity [76].
3. **Dependent Cluster** : This cluster’s enablers have a high degree of dependence but a low level of driving power [76].
4. **Independent Cluster:** The enablers in this cluster have weak, dependent power but significant driving power; they are also known as ”key enablers” [76].

4.3.6 Development of Conical Matrix

The main goal of developing a conical matrix is to execute a MICMAC analysis. The conical matrix (Table 6) was generated based on the data presented in Tables 4 and 5. In Table 6, ”Dri” and ”Dep” show the driving and dependence power criteria. Initially, each criterion was arranged according to its level number (Table 5). Second, the values of every criterion were considered based on Table 4.

Table 6: Conical Matrix after Clustering Enablers

	RE2	RE4	RE5	RE6	RE7	RE8	RE9	RE10	RE11	RE3	RE1	DRI
RE2	1	1	1	1	1	1	1	1	1	1	0	10
RE4	1	1	1	1	1	1	1	1	1	1	0	10
RE5	1	0	1	1	1	1	1	1	1	1	0	9
RE6	1	1	1	1	1	1	1	1	1	0	1	10
RE7	1	0	1	1	1	1	1	1	1	0	0	8
RE8	1	0	1	1	1	1	1	1	1	0	0	8
RE9	1	0	0	1	1	0	1	1	1	0	0	6
RE10	1	0	0	1	1	1	1	1	1	0	0	7
RE11	1	0	0	1	1	1	1	1	1	0	1	8
RE3	1	1	1	1	1	1	1	1	1	1	0	10
RE1	1	1	1	1	1	1	1	1	1	1	1	11
Dep	11	5	8	11	11	10	11	11	11	5	3	

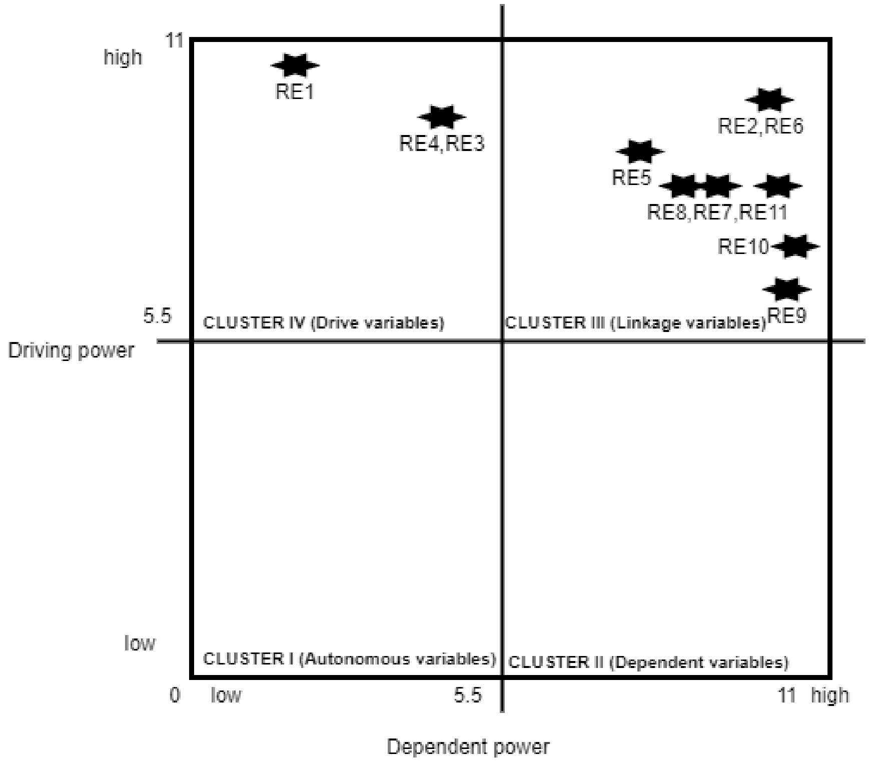


Figure 4: Graphical View of MICMAC Analysis

Figure 4 depicts the MICMAC analysis findings. The RE security practices were grouped into four distinct clusters for the MICMAC analysis. A clustering of RE security practices categories is shown in Figure 4. The first cluster comprises (autonomous enablers), the second cluster includes (dependent enablers), and the third and fourth clusters include (independent enablers). The results show that the RE1 ”Awareness of SRE,” RE3 "Requirements Elicitation of Security Requirements," and RE4"Analysis and Negotiations of Security Requirements" criteria are considered driving variable categories and have, thus, been isolated from the system. It is noted that RE2 "Methods and Tools," RE5-RE11, have strong driving and dependency power and influence other enablers owing to a strong relationship. This renders all the categories interlinked with each other but not fully dependent on any category. Thereby, we need practices from various categories to meet security requirements in the software development process (GSD context). Interestingly, no categories belong to a dependent cluster or autonomous clusters.

5. Summary, Implications and Future Work

In software engineering, it is important to carefully consider the practices with the intent to develop secure software projects at the beginning. Requirements engineers need to examine the best practices that come with the GSD paradigm, which is being considered by the vast majority of software development companies. Since this research investigates and evaluates the security practices that need to be adopted by requirements engineering teams in the context of GSD, this paper is an extension of our previously published systematic literature review.

In the first phase of this research, a questionnaire survey was conducted with GSD experts. The results of this survey were used to assess the importance of the highlighted RE security practices in real-world practice. The data collection process for the survey yielded 50 responses that were considered for the final data sample. According to the frequency analysis, these 70 RE security practices and their primary 11 categories are linked to industry practices. Survey results depict that the most common security requirement engineering (SRE) practices are well-defined client roles and resource capabilities, abuse and misuse cases, record rationale for security requirements, perform security requirements specification, and define standard templates for describing authentication, authorization, immunity, privacy, integrity, non-repudiation, intrusion detection, and system maintenance security requirements. These operations yield outcomes that are inextricably tied to the software's economic value.

Secondly, in the third phase, we used the ISM technique to investigate the links between GSD organizations in the SSD process 11 major RE security practices categories. According to the findings, the RE1 "Awareness of SRE" category is the top for selecting RE practices for SSD. This shows that RE1 is an independent category in the identified list of RE practices for SSD. All the other categories are dependent on RE1. The ISM approach results also present that RE3 "Requirement Elicitations" is dependent only on level 3 (RE1: Awareness of SRE), but all the coming categories (RE2, RE4-RE11) of level 1 depend on level 2 category (RE3). The findings further depict that RE2, RE4-RE11 practices categories depend on RE3 and RE1.

The study implications for researchers and practitioners are as follows:

- **For Researchers:** By conducting a thorough assessment of both academic and literature, the study offers a state-of-the-art summary of the RE security practices that potentially positively impact GSD organizations in SSD procedures. The study findings give a body of knowledge for researchers to use in developing RE security practices to deploy SSD approaches. In addition, the study presents a ranked framework for the observed RE security practices categories. The security practices are investigated within the framework of their priority ranking and the link between the fundamental categories of the identified RE security practices. We believe that a prioritization-based ranking will assist researchers in thinking about the most significant RE security practice category in their ongoing and future work.
- **For Practitioners:** An in-depth literature review and empirical studies provide a body of information to industry specialists regarding the RE security practices for the GSD organizations in the SSD process. This research provides 70 RE security practices and categorizes them into 11 core categories, each of which calls on industry practitioners to focus on them throughout the implementation of RE initiatives for the SSD process. Prioritization and categorization of identified practices will assist GSD practitioners in considering the most significant RE security practice category aspect on priority. The practitioners will be assisted in revising and developing new strategies for successfully implementing RE practices if the security risks they face are first identified and then prioritized. In addition, this study

presents a comprehensive view of RE security practices categories, enlightening practitioners regarding which category is critically important for SRE.ISM was also introduced as a unique methodology to help RE industry experts fix any ambiguous viewpoints of GSD experts in the SSD domain.

- **Future Work:** The development of security models and techniques for RE procedures in the real-world industry has not received much scholarly attention. In the future, we will use a fuzzy analytical hierarchy process (FAHP) to design a framework/model that supports RE in software development by identifying critical security risks, best practices, levels of RE practices categories, and a road map. Various areas, including political, economic, and management sciences, have extensively used AHP to solve complicated problems. When measuring multiple criteria's relative importance, classical AHP cannot handle the ambiguity and obscurity of the decision-maker. Because of this, fuzzy AHP was developed, which outperformed AHP in terms of accuracy and efficiency [57-59]. With these insights in mind, we have chosen to use them in future work on fuzzy AHP over other approaches. This is the case even though integrating security into RE is extremely important. Given the importance of security concerns in software development, we are driven to create a secure RE maturity model (SREMM) that will aid GSD firms in measuring their security maturity level and recommending best practices for successfully executing RE activities. SREMM will be engineered on SLR and empirically discover RE security risks, its best practices, and taking guidance from existing security models in software engineering disciplines. The security maturity level components will be used to assess the GSD organization's maturity level in the RE process and recommend best practices to improve its RE capabilities. The proposed model will be helpful in the GSD industry's efforts to carry out SRE activities in the actual world.

6. Research Limitations

Initially, we conducted SLR to study the relevant literature to find out the RE security practices to address the security risks faced by GSD organizations in the RE phase of the SDLC. In the conduction of SLR, the first author searched the literature with the help of a defined search string and completed the primary and final selection of the papers. In contrast, the second co-author reviewed the selection and data extraction processes. This process may be biased. To mitigate this risk, this paper's third co-author carried out the inclusion, exclusion, quality assessment, and data extraction processes for a total of fifteen random studies selected from 121 final papers. In addition, we conducted the inter-rater reliability test with Software Engineering Research Group (University of Malakand) (SERG_UOM) experts. The findings demonstrate no major bias and that the data collected and the analysis are compatible.

As a second consideration, relevant published materials were likely overlooked throughout the data collection procedure. This shortcoming is not systematic, as our investigation includes 121 representative literature items [48, 60]. The questionnaire survey method has been implemented to conduct an empirical investigation of the identified RE security practices with the assistance of GSD industry experts in the SSD domain. When developing survey instruments, there is always the possibility of encountering a risk. This concern was mitigated by piloting and evaluating the development questionnaire with "Software Engineering Research Group (SERG UOM) Pakistan", "King Fahd University of Petroleum and Minerals, Saudi Arabia", and "Qatar University, Doha, Qatar."

For the third concern, the ISM approach findings are based on thirteen experts' decisions, which may be a tiny data set. This is because these investigations are subjective, and other studies [43, 47] have also used a small data set for this analysis. As a result, the outcomes of the ISM technique are generalizable.

7. Conclusion

The software has become an indispensable part of human life, and we live in the internet of everything. Thus, software security is critical because a malware attack can cause extreme damage to any piece of software while compromising integrity, authentication, and availability, and it results in to breach the personal information, etc. It is important to consider the security practices from the beginning of the software development life cycle to develop secure software. This paper investigates the important practices to consider in the requirements engineering phase for SSD in the GSD domain. Conducting an empirical study with experts, we explored 70 practices and were taxonomized into 11 fundamental dimensions (categories)to assist GSD organizations in specifying the requirements for SSD.

Additionally, we analyzed the interrelationship among core dimensions of identified practices aiming to check their dependency, interdependency, and independency. The results depict the ”awareness of secure requirement engineering” category has the most decisive influence on the other ten core categories of the identified security practices. The ”requirements elicitation” category is fully dependent just on one category, i.e., ”awareness of secure requirement engineering,” and other categories are fully dependent on both these categories. We further performed the MICMAC analysis to check the right cluster of requirements engineering categories. The results show that the ”awareness of secure requirement engineering”, ”requirements elicitation”, and ”analysis and negotiations of security requirements” categories are considered driving variable categories and have, thus, been isolated from the system. It is noted that ”methods and tools”, have strong driving and dependency power and influence other enablers owing to a strong relationship. This renders all the categories interlinked with each other but not fully dependent on any category. We believe the results and discussion of this study will serve as a body of knowledge for research and practitioners' community to develop effective strategies towards considering security from the requirements engineering phase of software development.

References:

[1] Srivastava. Amit. Kumar and K. Shishir, ”An effective computational technique for taxonomic position of security vulnerability in software development,” *Journal of Computational Science,* vol. 25, pp. 388-396, March 2018.

[2] Li. Jin, Zhang. Yinghui, Chen. Xiaofeng, and X. Yang, ”Secure attribute-based data sharing for resource-limited users in cloud computing,” *Computers & Security,* vol. 72, pp. 1-12, January 2018.

[3] Y. Lee and G. Lee, ”HW-CDI: Hard-Wired Control Data Integrity,” *IEEE Access,* vol. 7, pp. 10811-10822, 2019.

[4] R. A. Khan, S. U. Khan, H. U. Khan, and M. Ilyas, ”Systematic Literature Review on Security Risks and its Practices in Secure Software Development,” *IEEE Access,* vol. 10, pp. 5456-5481, 2022.

[5] O. Olukoya, ”Assessing frameworks for eliciting privacy & security requirements from laws and regulations,” *Computers & Security,* vol. 117, p. 102697, 2022/06/01/ 2022.

[6] N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. A. Babar, ”Software security patch management - A systematic literature review of challenges, approaches, tools and practices,” *Information and Software Technology,* vol. 144, p. 106771, 2022/04/01/ 2022.

[7] H. Nina, J. A. Pow-Sang, and M. Villavicencio, ”Systematic Mapping of the Literature on Secure Software Development,” *IEEE Access,* vol. 9, pp. 36852-36867, 2021.

[8] S. V. Solms and L. A. Futcher, ”Adaption of a Secure Software Development Methodology for Secure Engineering Design,” *IEEE Access,* vol. 8, pp. 125630-125637, 2020.

[9] A. Ramirez, A. Aiello, and S. J. Lincke, ”A Survey and Comparison of Secure Software Development Standards,” in *2020 13th CMI Conference on Cybersecurity and Privacy (CMI) - Digital Transformation - Potentials and Challenges(51275)* , 2020, pp. 1-6.

[10] M. Niazi, A. M. Saeed, M. Alshayeb, S. Mahmood, and S. Zafar, ”A maturity model for secure requirements engineering,” *Computers & Security,* vol. 95, p. 101852, 2020/08/01/ 2020.

[11] D. Verdon and G. McGraw, ”Risk Analysis in Software Design,” *IEEE Security and Privacy,*vol. 2, pp. 79–84, 2004.

[12] S. Lipner, ”The Trustworthy Computing Security Development Lifecycle,” presented at the Proceedings of the 20th Annual Computer Security Applications Conference, 2004.

[13] S. Gupta, M. Faisal, and M. Husain, ”Secure Software Development Process for Embedded Systems Control,” *International Journal of Engineering Sciences & Emerging Technologies,* vol. 4, pp. 133-143, 12/01 2012.

[14] M. Essafi, L. Jilani, and H. Ben Ghezala, *S2D-ProM: A Strategy Oriented Process Model for Secure Software Development* , 2007.

[15] J. Manico, ”OWASP ” in*Application Security Verification Standard 3.0.1* , ed, 2016, pp. 1-70.

[16] A. S. Sodiya, ”Towards Building Secure Software Systems,” 01/01 2006.

[17] H. Al-Matouq, S. Mahmood, M. Alshayeb, and M. Niazi, ”A Maturity Model for Secure Software Design: A Multivocal Study,” *IEEE Access,* vol. 8, pp. 215758-215776, 2020.

[18] T. Li and Z. Chen, ”An ontology-based learning approach for automatically classifying security requirements,” *Journal of Systems and Software,* vol. 165, p. 110566, 2020/07/01/ 2020.

[19] R. Kumar and R. Goyal, ”On cloud security requirements, threats, vulnerabilities and countermeasures: A survey,” *Computer Science Review,* vol. 33, pp. 1-48, 2019/08/01/ 2019.

[20] Sharma. Anuradha and M. P. Kumar, ”Aspects of Enhancing Security in Software Development Life Cycle,” *Advances in Computational Sciences and Technology,* vol. 10, pp. 203-210, 2017.

[21] M. N. R, ”Identifying security requirements using the security quality requirements engineering (SQUARE) method,” *Integrating Security and Software Engineering,* pp. 44–69, 2006.

[22] M. Alam, J. P. Seifert, and X. Zhang, ”A Model-Driven Framework for Trusted Computing Based Systems,” in *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)* , 2007, pp. 75-75.

[23] H. Mouratidis, P. Giorgini, and G. Manson, ”When Security Meets Software Engineering: A Case of Modeling Secure Information Systems,” *Journal of Information Systems,* vol. 30, pp. 609-629, 2005.

[24] Flechais. Ivan, M. Angela. Sasse, and S. M. V. Hailes, ”Bringing Security Home: A process for developing secure and usable systems ” in *New Security Paradigms Workshop* , Ascona, Switzerland, 2003, pp. 49-57.

[25] M. Khari, Vaishali, and P. Kumar, ”Embedding security in Software Development Life Cycle (SDLC),” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* , 2016, pp. 2182-2186.

[26] M. Younas, M. A. Shah, D. N. A. Jawawi, M. K. Ishfaq, M. Awais, K. Wakil*, et al.* , ”Elicitation of Nonfunctional Requirements in Agile Development using Cloud Computing Environment,” *IEEE Access,* pp. 1-1, 2020.

[27] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, ”Systematic literature reviews in software engineering – A systematic literature review,” *Information and Software Technology,* vol. 51, pp. 7-15, 2009/01/01/ 2009.

[28] L. Zhang, J. Tian, and J. Jiang, ”Empirical Research in Software Engineering — A Literature Survey,” *Jounrla of Computer Science Technology,* vol. 33, pp. 876–899, 2018.

[29] S. Wagner, D. M. Fernández, M. Felderer, A. Vetrò, M. Kalinowski, R. Wieringa*, et al.* , ”Status Quo in Requirements Engineering: A Theory and a Global Family of Surveys,” *ACM Trans. Softw. Eng. Methodol.,* vol. 28, p. Article 9, 2019.

[30] M. Niazi, D. Wilson, and D. Zowghi, ”Critical success factors for software process improvement implementation: an empirical study,” *Software Process: Improvement and Practice,* vol. 11, pp. 193-211, 03/01 2006.

[31] H. U. Rahman, M. Raza, P. Afsar, and H. U. Khan, ”Empirical Investigation of Influencing Factors Regarding Offshore Outsourcing Decision of Application Maintenance,” *IEEE Access,* vol. 9, pp. 58589-58608, 2021.

[32] M. A. Akbar, W. Naveed, A. A. Alsanad, L. Alsuwaidan, A. Alsanad, A. Gumaei*, et al.* , ”Requirements Change Management Challenges of Global Software Development: An Empirical Investigation,” *IEEE Access,* vol. 8, pp. 203070-203085, 2020.

[33] H. Mumtaz, M. Alshayeb, S. Mahmood, and M. Niazi, ”An empirical study to improve software security through the application of code refactoring,” *Information and Software Technology,* vol. 96, pp. 112-125, 2018/04/01/ 2018.

[34] B. Kitchenham and S. L. Pfleeger, ”Principles of survey research part 6: data analysis,” *SIGSOFT Softw. Eng. Notes,* vol. 28, pp. 24–27, 2003.

[35] A. A. Khan, M. Shameem, M. Nadeem, and M. A. Akbar, ”Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping,” *Applied Soft Computing,* vol. 102, p. 107090, 2021/04/01/ 2021.

[36] B. Martin,*Introduction to Medical Statistics* , 4th Edition ed., 2015.

[37] S. A. P, ”Interpretive structural modeling: Methodology for large scale systems. New York, McGraw-Hill,” pp. 1-445, 1977.

[38] V. Ravi and R. Shankar, ”Analysis of interactions among the barriers of reverse logistics,” *Technological Forecasting and Social Change,* vol. 72, pp. 1011-1029, 2005/10/01/ 2005.

[39] H. Shakeri and M. Khalilzadeh, ”Analysis of factors affecting project communications with a hybrid DEMATEL-ISM approach (A case study in Iran),” *Heliyon,*vol. 6, p. e04430, 2020/08/01/ 2020.

[40] S. Rafi, M. A. Akbar, S. Mahmood, A. Alsanad, and A. Alothaim, ”Selection of DevOps best test practices: A hybrid approach using ISM and fuzzy TOPSIS analysis,” *Journal of Software: Evolution and Process,* vol. 34, p. e2448, 2022.

[41] S. Rafi, M. A. Akbar, W. Yu, A. Alsanad, A. Gumaei, and M. U. Sarwar, ”Exploration of DevOps testing process capabilities: An ISM and fuzzy TOPSIS analysis,” *Applied Soft Computing,* vol. 116, p. 108377, 2022/02/01/ 2022.

[42] G. Kannan, S. Pokharel, and P. Sasi Kumar, ”A hybrid approach using ISM and fuzzy TOPSIS for the selection of reverse logistics provider,” *Resources, Conservation and Recycling,* vol. 54, pp. 28-36, 2009/11/01/ 2009.

[43] A. Agarwal and P. Vrat, ”Modeling Attributes of Human Body Organization Using ISM and AHP,” *Jindal Journal of Business Research,* vol. 6, pp. 44-62, 2017.

[44] C. Sakar, B. Koseoglu, A. C. Toz, and M. Buber, ”Analysing the effects of liquefaction on capsizing through integrating interpretive structural modelling (ISM) and fuzzy Bayesian networks (FBN),” *Ocean Engineering,* vol. 215, p. 107917, 2020/11/01/ 2020.

[45] M. N. Patel, A. A. Pujara, R. Kant, and R. K. Malviya, ”Assessment of circular economy enablers: Hybrid ISM and fuzzy MICMAC approach,” *Journal of Cleaner Production,* vol. 317, p. 128387, 2021/10/01/ 2021.

[46] S. Salinger, L. Plonka, and L. Prechelt, ”A Coding Scheme Development Methodology Using Grounded Theory For Qualitative Analysis Of Pair Programming,” *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments,*vol. 4, 05/31 2008.

[47] M. Azeem Akbar, S. Mahmood, A. Alsanad, and A. Com, ”Toward Successful DevSecOps in Software Development Organizations: A Decision-Making Framework,” *Information and Software Technology,* vol. 147, 02/27 2022.

[48] M. Niazi, S. Mahmood, M. Alshayeb, A. M. Qureshi, K. Faisal, and N. Cerpa, ”Toward successful project management in global software development,” *International Journal of Project Management,* vol. 34, pp. 1553-1567, 2016/11/01/ 2016.

[49] A. Khan, M. Niazi, and S. Hussain, ”Systematic Literature Study for Dimensional Classification of Success Factors Affecting Process Improvement in Global Software Development: Client-Vendor Perspective,” *IET Software,* vol. 12, 04/04 2018.

[50] W. Afzal, R. Torkar, and R. Feldt, ”A systematic review of search-based testing for nonfunctional system properties,” *Information and Software Technology,* vol. 51, pp. 957-976, 2009/06/01/ 2009.

[51] N. Mead and T. Stehney, ”Security quality requirements engineering (SQUARE) methodology,” *ACM SIGSOFT Software Engineering Notes,* vol. 30, pp. 1-7, 07/01 2005.

[52] Y. Mufti, M. Niazi, M. Alshayeb, and S. Mahmood, ”A Readiness Model for Security Requirements Engineering,” *IEEE Access,* vol. 6, pp. 28611-28631, 2018.

[53] K. Rindell, J. Ruohonen, J. Holvitie, S. Hyrynsalmi, and V. Leppänen, ”Security in agile software development: A practitioner survey,” *Information and Software Technology,* vol. 131, p. 106488, 2021/03/01/ 2021.

[54] M. Soni, *End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery* , 2015.

[55] R. Attri, S. Grover, N. Dev, and D. Kumar, ”Analysis of barriers of total productive maintenance (TPM),” *International Journal of System Assurance Engineering and Management,* vol. 4, pp. 365-377, 2013/12/01 2013.

[56] J. N. Warfield, ”Developing Interconnection Matrices in Structural Modeling,” *IEEE Transactions on Systems, Man, and Cybernetics,* vol. SMC-4, pp. 81-87, 1974.

[57] C.-K. Kwong and H. Bai, ”A fuzzy AHP approach to the determination of importance weights of customer requirements in quality function deployment,” *Journal of intelligent manufacturing,* vol. 13, pp. 367-377, 2002.

[58] C.-K. Kwong and H. Bai, ”Determining the importance weights for the customer requirements in QFD using a fuzzy AHP with an extent analysis approach,” *iie Transactions,* vol. 35, pp. 619-626, 2003.

[59] M. A. Akbar, M. Shameem, S. Mahmood, A. Alsanad, and A. Gumaei, ”Prioritization based taxonomy of cloud-based outsource software development challenges: Fuzzy AHP analysis,” *Applied Soft Computing,* vol. 95, p. 106557, 2020.

[60] M. A. Akbar, J. Sang, A. A. Khan, S. Mahmood, S. F. Qadri, H. Hu*, et al.* , ”Success factors influencing requirements change management process in global software development,” *Journal of Computer Languages,* vol. 51, pp. 112-130, 2019.

Appendix A: Demographics of Survey Respondents

Responses No	**Country of Job**	**Your position in the Organization**	**Work Experience**	**Size of your organization in terms of the number of employees working**	**What type of SPI standards or models your company has achieved?**
	Pakistan	Software Developer	6 to 10 Years	Small-to-Medium Enterprise (50-200 Employees)	Not Sure
	USA	Software Security Expert, Software Developer	3 to 6 Years	Small-Enterprise (10-50 Employees)	ISO
	Malaysia	Project Leader, Academician	10 Years or more	Large-Enterprise (More than 200 Employees)	ISO
	Pakistan	Software Security Expert	1 to 3 Years	Micro-Enterprise (1-10 Employees)	Not Sure
	United Kingdom	Manager, Software Developer, Project Leader	10 Years or more	Micro-Enterprise (1-10 Employees)	CMMI
	Pakistan	Manager	3 to 6 Years	Small-Enterprise (10-50 Employees)	CMMI
	Pakistan	Security Tester	6 to 10 Years	Micro-Enterprise (1-10 Employees)	Not Sure
	Australia	Manager	6 Months to 1 Year	Small-Enterprise (10-50 Employees)	CMMI
	USA	Software Security Expert	1 to 3 Years	Large-Enterprise (More than 200 Employees)	CMMI
	United Kingdom	Software Developer	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI
	China	Software Developer	3 to 6 Years	Micro-Enterprise (1-10 Employees)	Not Sure
	India	Project Leader	1 to 3 Years	Small-Enterprise (10-50 Employees)	ISO
	Germany	Software Designer, Analyst	1 to 3 Years	Small-to-Medium Enterprise (50-200 Employees)	CMMI

Responses No	**Country of Job**	**Your position in the Organization**	**Work Experience**	**Size of your organization in terms of the number of employees working**	**What type of SPI standards or models your company has achieved?**
	Ireland	Software Designer	10 Years or more	Small-to-Medium Enterprise (50-200 Employees)	CMMI
	Baghdad, Iraq	Analyst	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Malaysia	Product Owner	1 to 3 Years	Small-to-Medium Enterprise (50-200 Employees)	ISO
	New Dehli, India	Manager	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Pakistan	Software Developer	3 to 6 Years	Micro-Enterprise (1-10 Employees)	Not Sure
	UK	Software Security Expert, Security Tester	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI
	Malaysia	Software Security Expert	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Japan	Manager	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Saudi Arabia	Manager	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	USA	Software Security Expert, Software Developer	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Tehran, Iran	Software Developer, Member of the Security Group	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Malaysia	Software Developer, Security Tester	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Australia	Software Security Expert	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI

Responses No	Country of Job	Your position in the Organization	Work Experience	Size of your organization in terms of the number of employees working	What type of SPI standards or models your company has achieved?
	Malaysia	CEO, Software Security Expert, Software Developer	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI, ISO
	Malaysia	CEO, Software Security Expert	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Germany	Member of the Security Group	1 to 3 Years	Large-Enterprise (More than 200 Employees)	CMMI
	India	Software Security Expert, Software Developer, Project Leader	3 to 6 Years	Small-to-Medium Enterprise (50-200 Employees)	CMMI
	Depok, Indonesia	CEO, Software Security Expert	1 to 3 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Japan	CEO	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	China	Project Leader	3 to 6 Years	Large-Enterprise (Morethan 200 Employees)	ISO
	China	CEO, Software Security Expert, Member of the Security Group	6 to 10 Years	Micro-Enterprise (1-10 Employees)	CMMI
	Kuala Lumpur, Malaysia	Member of the Security Group, Software Designer	1 to 3 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Malaysia	CEO, Software Security Expert, Software Developer	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Australia	Software Developer	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI
	Limerick, Ireland	Software Security Expert, Manager, Software Developer	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI

Responses No	**Country of Job**	**Your position in the Organization**	**Work Experience**	**Size of your organization in terms of the number of employees working**	**What type of SPI standards or models your company has achieved?**
	Sydney, Australia	CEO, Software Security Expert, Software Developer	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI, ISO
	India	CEO, Software Security Expert, Security Tester	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI
	India	CEO, Software Security Expert, Software Developer	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Rabat, Morocco	CEO, Software Security Expert, Software Developer	3 to 6 Years	Large-Enterprise (Morethan 200 Employees)	CMMI
	USA	CEO, Software Security Expert, Software Developer, Project Leader	10 Years or more	Micro-Enterprise (1-10 Employees)	CMMI
	USA	CEO, Security Tester	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI
	United Kingdom	CEO, Software Security Expert	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI
	United Kingdom	CEO, Software Security Expert, Software Developer, Project Leader	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI, ISO
	UK	CEO, Software Security Expert, Manager	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI, ISO
	Germany	CEO, Software Security Expert, Software Developer, Member of the Security Group	6 to 10 Years	Large-Enterprise (More than 200 Employees)	CMMI
	Germany	CEO, Member of the Security Group	10 Years or more	Large-Enterprise (More than 200 Employees)	CMMI

Responses No	**Country of Job**	**Your position in the Organization**	**Work Experience**	**Size of your organization in terms of the number of employees working**	**What type of SPI standards or models your company has achieved?**
	Rabat, Morocco	CEO, Software Security Expert	3 to 6 Years	Large-Enterprise (More than 200 Employees)	CMMI