



A reference measurement framework of software security product quality (SPQ^{NFSR})

Khalid T. Al-Sarayreh¹  | Mamdouh Alenezi² | Mohammed Zarour²  | Kenza Meridji³

¹Department of Software Engineering, Hashemite University, Zarqa, Jordan

²Department of Computer Science, Prince Sultan University, Riyadh, Saudi Arabia

³Department of Software Engineering, University of Petra, Amman, Jordan

Correspondence

Khalid T. Al-Sarayreh, Department of Software Engineering, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan.

Funding information

Hashemite University

Abstract

Currently, the customer's demands have expressively amplified their expectations of getting software at a high-quality level. However, the non-functional requirements of the software products attention have been expanded in both the academic and the industrial fields; so, there is no framework for specifying and measuring such kinds of quality constraints for the security requirements of software product quality. This paper presents an integrated framework of the early specification and measurement of the functional and non-functional software security requirements. Such a measurement framework would help software and systems engineers to improve product qualities whether the software has already been delivered or has yet to be built. The main steps that have been followed include: identify, specify and measure the software security requirements based on ISO/IEC SQuaRE series of international standards for software product quality. A standard measurement framework used to measure the functional size of the software product quality to develop a functional size measurement of the functional and non-functional security requirements is described. As a result, a functional size measurement framework of the functional and non-functional security requirements (SPQ^{NFSR}) using international standards is proposed. An automatic teller machine case study for the measurement of security requirements based on perspectives of a software functional user requirements is presented. Finally, it is concluded that it is essential to develop such a functional size measurement framework for functional and non-functional security requirements to support developers to face the challenges derived from early dealing with such requirements.

1 | INTRODUCTION

Currently, software engineers are required to identify their measures and implementations for all requirements of software products. More specifically, software engineers are responsible for identifying, specifying and measuring all different types of software product requirements. This includes internal and external security measures [1,2]. Specifying security requirements early in the software process is a prime concern for many software development organizations [1].

Many software products failed to deliver because of poorly identified and measured requirements, including security requirements. Software engineers have no consensual reference model [3–6] that is built based on consolidated different industrial views and on different types of international standards to

justify the need for such software requirements [4–8]. Accordingly, the identification of security requirements, for example, is delayed in the software process, hence they are specified vaguely.

Software security requirements show the essentials of software components security in its environment to perform its tasks correctly, accurately and completely within the specified time. Besides, it directs the limitations of software security associated with system security applications and software security awareness, such as access auditability, access controllability and data corruption/prevention as well as data encryptions [1]. It will help the system to guarantee software suitability and availability for every task executed in such software products.

Software 'functionalities fall under the concept of functional user requirements (FURs) and refer to the set of functions or services required from the software system,

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. *IET Information Security* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

whereas constraints fall under the concept of non-functional requirements (NFRs). It should be noted that a number of such constraints, while referred to as software-NFRs by some authors, are referred to as quality aspects by others' [1].

The software quality requirements and evaluation (SQuARE) standards ISO/IEC 25010 [9] and ISO/IEC 25012 [10], as well as the software product quality ISO/IEC 25021 [11] standards, illustrate software security requirements and their measures as part of the software functionality to define the software product quality. Moreover, the capability of the software product to protect information and data from unauthorized persons.

The ISO/IEC/IEEE 29148 [12] outlines example might be used to specify of security and privacy requirements containing 'access limitations to the system, such as existence of log-on procedures and passwords and of data protection and recovery methods. This could include the factors that would protect the system from accidental or malicious access, use, modification, destruction or disclosure'.

The ISO/IEC/IEEE 12207 [13] defines security as the 'protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them' and ISO/IEC 25010 [9] defines the security as the 'degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization'.

Precise security requirements could include essential cryptographic techniques; to keep a specific log or history data sets; assign certain functions to different modules; restrict communications between some areas of the program and check data integrity for critical variables.

Moreover, European standards (ECSS) series [14–16] define security as the specifications, including related factors, which might compromise sensitive information.

The different perspectives of these ISO international standards and European standards, such as the (ECSS) series of standards express into their contents general common view regarding security; the security can be internal (within the same software item) or external (in another software item). Although the ECSS standards deals with security specific to the software-embedded system developed as part of a space project, the proposed measurement method is applicable for non-embedded software security.

ISO/IEC 14143-6:2012 [17] standardizes the basic concepts and definitions of functional size measurement. Detailed descriptions of various functional size measurement methods are recently published as standards. For instance, common software measurement international consortium (COSMIC) ISO/IEC 19761 [18], NESMA ISO/IEC 24570 [19] and international function point users group (IFPUG) ISO/IEC 20926 [20] Functional size measurement is used for many purposes, for example, to help to estimate the effort of a starting development project or measuring the actual productivity of a finished development endeavour.

The ISO/IEC 19761 (COSMIC standard) [18] defines the principles, rules and a process for measuring a standard

functional size of a piece of software. 'Functional size' is a measure of the 'amount of functionality' provided by the software. The purpose of the measurement is to determine the COSMIC (functional size of the security requirement of a software application).

Despite the existence of several measures for software security requirements, most of these measures are still built based on personal views. They are still unsuccessfully demonstrated in one requirement framework, which resulted in having the security measures stated informally. Moreover, they used the NFRs in a faulty context or stated in poorly techniques. Consequently, such requirements' specifications cannot be acceptable by software engineers or software project managers to use them in the estimation context or on benchmarking of software products.

The motivation of this work is to support software and systems engineers with a way of using a reference framework for early identification, specification and measurements of the functional and NFRs for the security requirements. The results of the proposed framework will be used in the future in an estimation effort and benchmarking. The International Software Benchmarking Group can use, that is, the measurement results of the proposed reference framework.

The paper also reports the design of the measurement method. To identify the functional size of the software product quality based on international standards and using the COSMIC standard as a free method to identify the functional size of the software security independently of the software languages, which avoids the weaknesses observed in the product quality measures currently available.

The measurement scope is to identify separately all functionality allocated to software security requirements as a piece of the application.

Furthermore, the main contribution of this paper is the proposed 'reference measurement framework for security requirements' of the software product quality.

The measurement framework proposed in this paper; is considered as a 'reference framework' in the sense of an 'etalon' standard that is being used for the measurement of product quality.

This paper is organized as follows. Section 2 presents related works. Section 3 presents to design a standards-based reference framework for measuring security requirements. Section 4 presents the numerical assignment rules for functional and non-functional security. Section 5 presents the evaluation of the proposed measures for security. Section 6 presents the proposed reference framework among software etalon. Section 7 presents the automatic teller machine (ATM) case study, and a conclusion is presented in section 8.

2 | RELATED WORK

This section presents the related works from previous studies in the literature followed by the most recent methods and techniques proposed up to date by the industry for the functional size measurement.

2.1 | Functional and non-functional security requirements

Recently, there are many published research studies on system and software security issues at the level of functional and NFRs, for instance, in Ref. [1,2] authors proposed a reference model for system NFRs allocated to software security requirements at different levels of details (software, system and product levels). Followed by a reference model of security requirements for early identification and measurement of security awareness program [6] where a trade-off model is proposed for software requirements to balance between security and usability issues [7].

Recently, a new security management framework for open-source software (OSS) projects [21] has referred to classes of NFRs related to system confidentiality, integrity and availability for OSS projects.

Yang et al. [22] proposed an approach for the anonymity of roaming users. They defined an authentication function to avoid the real-time involvement of the home network control centre when authenticating the roaming users. The results of the security and performance analysis show that the proposed scheme can provide the required security features while providing a small authentication delay.

More recently, researchers in [23] proposed a model to offer secure and user-friendly authentication for a large number of identity-based user authentication mechanisms for the wireless mobile environment. Using many situations where a user's private key and some other sensitive data can be exposed if an attacker remotely or physically controls his/her mobile device. Then they analysed the security requirements in practical applications.

Vistbakka et al. [24] proposed an integrated approach to derive and formalize safety and security requirements systematically. To facilitate requirements elicitation, they proposed a way to adapt and integrate traditional safety and security analysis techniques that will formally specify and verify the requirements.

The interplay between safety and security has been addressed in Ref. [24] by proposing 'an attack injection framework, based on model-implemented fault injection, suitable for model-based design'. The framework helps in evaluating the impact of cybersecurity attacks on system safety early in the development process. The results show that the modelled security attacks could successfully influence system safety by violating their defined safety requirements.

Consequently, Zhang et al. in [25] have analysed the industrial chain coordinating SaaS platform of security encryption configuration requirements of multi-tenant business data that takes corporate champion as the core. The platform data is used in verification, authorization, configuration management, key configuration management and hierarchical decryption query of user's identity authentication. The results show that corporate champion configuration can realize personalized data encryption requirements of different alliances.

Setyoko et al. [26] proposed a framework using ISO/IEC 15408 [27] for the security of a smart card that makes security

design. This research is consisting of three steps: first analysing threats, second designing security objectives and then designing functional security requirements. Threats assessment and analysis in this research has resulted in 10 threats.

However, Hovorushchenko and Pavlova [28] proposed the development method for the activity of ontology-based intelligent agent (OBIA) for evaluating the software requirements specifications (SRS). OBIA evaluates the sufficiency of information in the SRS for assessing the non-functional software features.

Recently, Maskani and El Houssaini [29] followed a previous work that presents a model for security requirements based on a SysML extension. They presented a revision of this extension based on those observations. Then, they applied the revised extension to model the security requirements for a telemedicine system. Followed by Kunakov [30] that analysed the technological process improvement utilizing the introduction of digital technologies and information security requirements.

Ahanger et al. in [31] analysed the 'security requirements related to IoT by exploring the existing experimental studies to get an insight into the security requirements of the IoT'. The results of the study exposed that security threats are one of the main challenges for IoT, and it is essential, to mitigate them for the success of this platform substantially.

Finally, Subburaj and Urban in [32] analysed the security requirements for multi-agent systems (MAS). They proposed solutions to secure MAS and the use of formal methods to specify security requirements. They proposed a security requirements model of MAS early on in the development process. Functional specifications of MAS are modelled along with the non-functional security requirements using the Descartes-Agent specification language.

Finally, ISO/IEC 19515 [33], expresses a technique for systematizing the 'counting of function points that is generally consistent with the function point counting practices manual' (IFPUG CPM) produced by the IFPUG.

The motivation of this research paper is to support software engineers with a way of using a reference model for early identification, specification, and measurements of the functional and non-functional security requirements.

This paper also reports the design measurement method to identify the functional size of the software product quality based on international standards and using COSMIC standard. The main contribution of this paper is the proposal of a reference framework of measuring the functional and non-functional security requirements for software product quality (SPQ^{NFSR}).

2.2 | The international standard for software functional size measurement: ISO/IEC 19761

The COSMIC functional size measurement method [17] is supported by the COSMIC and is a recognized international standard (ISO/IEC 19761). 'In the measurement of functional software size using COSMIC, the functional software processes and their triggering events must be identified' [18].

The unit of measurement in this method is the data movement, which is a base functional component that moves one or more data attributes belonging to a single data group. 'Data movements can be of four types: Entry (E), Exit (X), Read (R) or Write (W)' [18].

The functional process is a primary component of a set of user requirements triggered by one or more triggering events. 'The triggering event is an event occurring outside the boundary of the measured software and initiates one or more functional processes' [18]. The sub-processes of each functional process constitute sequences of events. See Figure 1 for an illustration of the generic flow of data groups through software from a functional perspective.

3 | DESIGN A REFERENCE FRAMEWORK FOR MEASURING SECURITY REQUIREMENTS

This section presents the design approach used to build this reference measurement framework of software security requirements early during the software development process. In this section, four design methodology steps are used. These design steps are recommended by Ref. [34] as follows:

Step 1: Purpose of measurement objectives for system or software product

Step 2: Classification of software concepts to be measured

Step 3: Design or selection of the meta-model throughout the Identification and specification software entities, entity types and entity relationship

Step 4: Numerical assignment rules for system or software requirements

The proposed framework for measuring the security requirements is built using entities (attributes); each entity type is represented by two or more entities or attributes. The set of entity types have built the meta-model for the system from a user point of views to produce a conceptual data model of security information system.

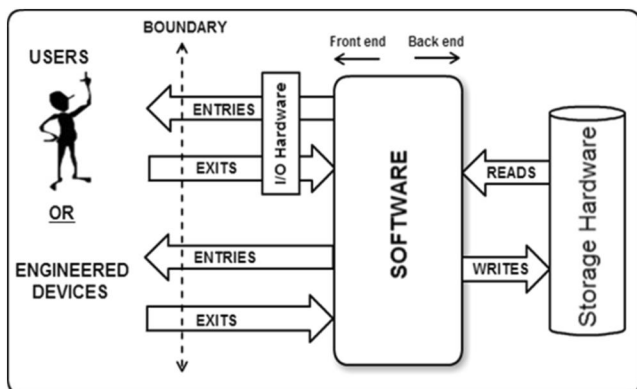


FIGURE 1 Generic ISO/IEC 19761 model for functional size measurements [18]

3.1 | Purpose of measurement objectives for software security requirements

This section presents the measurement objective of the security requirements, followed by security measurement point of view and the intended uses of the measurement results as follows:

3.1.1 | Measurement objective of security requirements

The main objective of this measurement is to measure the functional and non-functional security requirements as defined in ECSS and international ISO standards and using the ISO/IEC 19761 (COSMIC) as a measurement method.

3.1.2 | Measurement point of view for security

The measurement point of view in this paper measures the security requirements from the software perspective view for the functional and NFRs.

3.1.3 | Intended use of measurement results

The measurement of the functional security size (functional and non-functional) during the software process is intended to be used for software cost estimation and to identify the additional effort used in the absence of such NFRs.

3.2 | Classification of software security requirement concepts to be measured

In this section, security requirements are classified based on the consensual views on ISO international standards and European standards, such as the ECSS series of standards as follows: external and internal security requirements. The software security entities to be measured used in the reference framework includes the functional and non-functional security requirements of the software product quality as follows:

3.2.1 | External security entities to be measured

Security entities should be able to measure the following:

- Failing to prevent the leak of secure output information or data; this includes one entity to measure: the entity name is (*Auditability Entity*)
- Failing to prevent loss of relevant data or data corruption, this includes one entity to measure: the entity name is (*Integrity Entity*)

- Failing to defend against illegal access or illegal operation; this includes one entity to measure: the entity name is (*Confidentiality Entity*)

3.2.2 | Internal security entities to be measured

These security entities should be able to measure a set of attributes for assessing the capability of the software product to avoid illegal access to the system and data, hence maintain the confidentiality; this includes one entity to measure: the entity name is *Data Encryption Entity*.

3.3 | Design or selection of the security meta-model

This section illustrates the security requirements' entities to be measured into security requirements entity types and the relationships among such entity types. Four candidate security requirements entity types with their relationships are identified to be used by system and software engineers to measure the software security requirements. The practical specifications of such security entity types are presented using the following design template:

3.3.1 | Entity Type 1: External auditability

Entity type 1 is used to measure the external security requirements for auditability

- **Entity name:** auditability
- The **input** is the user access to the software
- The **output** is the recorded user access in the software
- The **process** should identify the number of accesses that the system recorded in the access history database for each user access.
- **An object of interest** is user access
- **Data sources** are: user access and recorded access to/in the software
- **Data distention** is access auditability
- Entity Type 1 is used to **measure** the functional size of the access auditability.
- **Entity relationship:** many to many of #users access to software with # of recorded user access in the software

3.3.2 | Entity Type 2: External integrity

Entity type 2 is used to measure the external security requirements for integrity.

- **Entity name:** integrity
- The **input** is the types of illegal operations as in the specifications

- The **output** is the detected different types of illegal operations
- The **process** should identify the detected types of different illegal operations on the software compared with types of illegal operations as in the specifications with the system
- **An object of interest** is an operation
- **Data sources:** are illegal operation type and detected illegal operation type to/in the software
- **Data distention** is access controllability
- Entity Type 2 is used to **measure** the functional size of the access controllability
- **Entity relationship:** many to many of # user control of illegal operation as defined in the specification with the detected # of illegal operation on the software

3.3.3 | Entity Type 3: External confidentiality

Entity Type 3 is used to measure the external security requirements for confidentiality

- **Entity name:** confidentiality
- The **input** is the frequency of data corrupted events to the software
- The **output** is the occurred major and minor data corrupted in the software
- The **process** should identify the occurrences of major and minor data corruption events in the software with the frequency of data corrupted events to the software
- **An object of interest** is data
- **Data sources** are: frequently of data corrupted and major and minor data corruption events to/in the software
- **Data distention** is data corruption/prevention
- Entity type 3 is used to **measure** the functional size of data corruption/prevention
- **Entity relationship:** many to many of detection of the data corruption in the software

3.3.4 | Entity Type 4: Internal data encryption

Entity type 4 is used to measure the internal security requirements for data encryption

- **Entity name:** data encryption
- The **input** is the required encryptable/decryptable data items to the software
- The **output** is the encryptable/decryptable data items in the software
- The **process** should identify the number of the encryptable/decryptable data items with the required encryptable/decryptable data items to/in the software
- An object of interest is data
- **Data sources** are required encryptable/decryptable data items with actual encryptable/decryptable data items to/in the software
- **Data distention** is data encryption

- Entity type 4 is used to **measure** the functional size of the data encryption
- **Entity relationship:** many to many of required encryptable/decryptable data items to software with actual encryptable/decryptable data items in the software

3.4 | The meta-model of functional security requirements

The section presents the meta-model of the functional security requirements, which is built based on the previous sections of the identified and specified of the four entity types and their relationships mapping with ISO/IEC 19761.

Figure 2 illustrates the meta-model of functional security requirements. The meta-model combines the security requirements as defined on different standards and the functional size measurement method listed in ISO/IEC 19761 in order to measure the functional security size as a piece of the software product.

3.5 | The meta-model of software non-functional security requirements

Figure 3 illustrates the meta-model of non-functional security requirements. This figure combines the security NFRs as defined on different standards and the functional size measurement method listed in ISO/IEC 19761 in order to measure the security non-functional size as piece of the software product.

4 | NUMERICAL ASSIGNMENT RULES FOR FUNCTIONAL AND NON-FUNCTIONAL SECURITY

This section presents a numerical assignment rules for the proposed meta-models and the characterization of the concepts illustrated in Figures 2 and 3, respectively. Numerical assignment rules can be described through a descriptive text (a practitioner's description) or mathematical expressions (a formal theoretical viewpoint), in this paper we used the descriptive text and numerical assignment rules.

The extension of the software functional security to software non-functional security requirements is used to build mathematical assignments rules based on mathematical expressions. The numerical assignment rules are added to the auditability, integrity, confidentiality and data encryption/decryption entities.

4.1 | Security measurement procedure

The functional size measurement procedures have been developed by applying the ISO/IEC 19761 (COSMIC) method. A subset of these measurement procedures is centred on the measurement of the functional size. From their conceptual meta-models in Figures 2 and 3, respectively.

ISO/IEC 19761 gives for software and systems engineers the ability to measure the functional requirements of the software or part of it, using the FURs perspectives. The fundamental technique for the FUR is the functional process for the software component(s) by implementing an independent data movement types for each component.

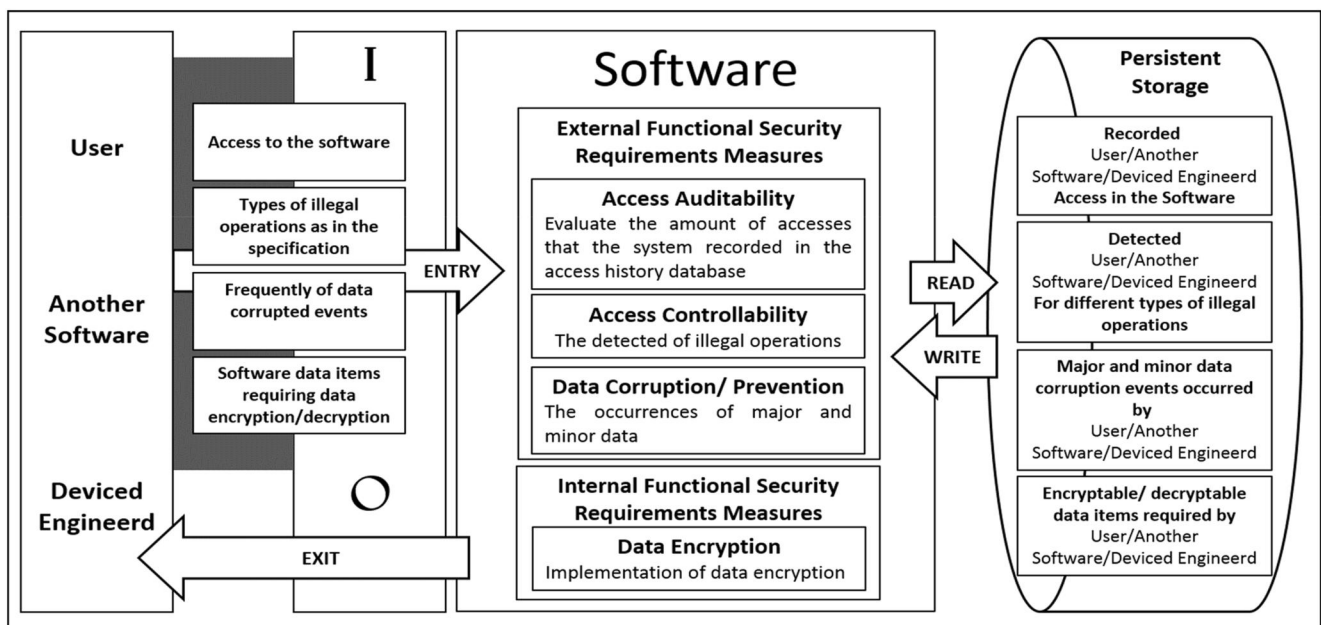


FIGURE 2 A standards-based functional security meta-model

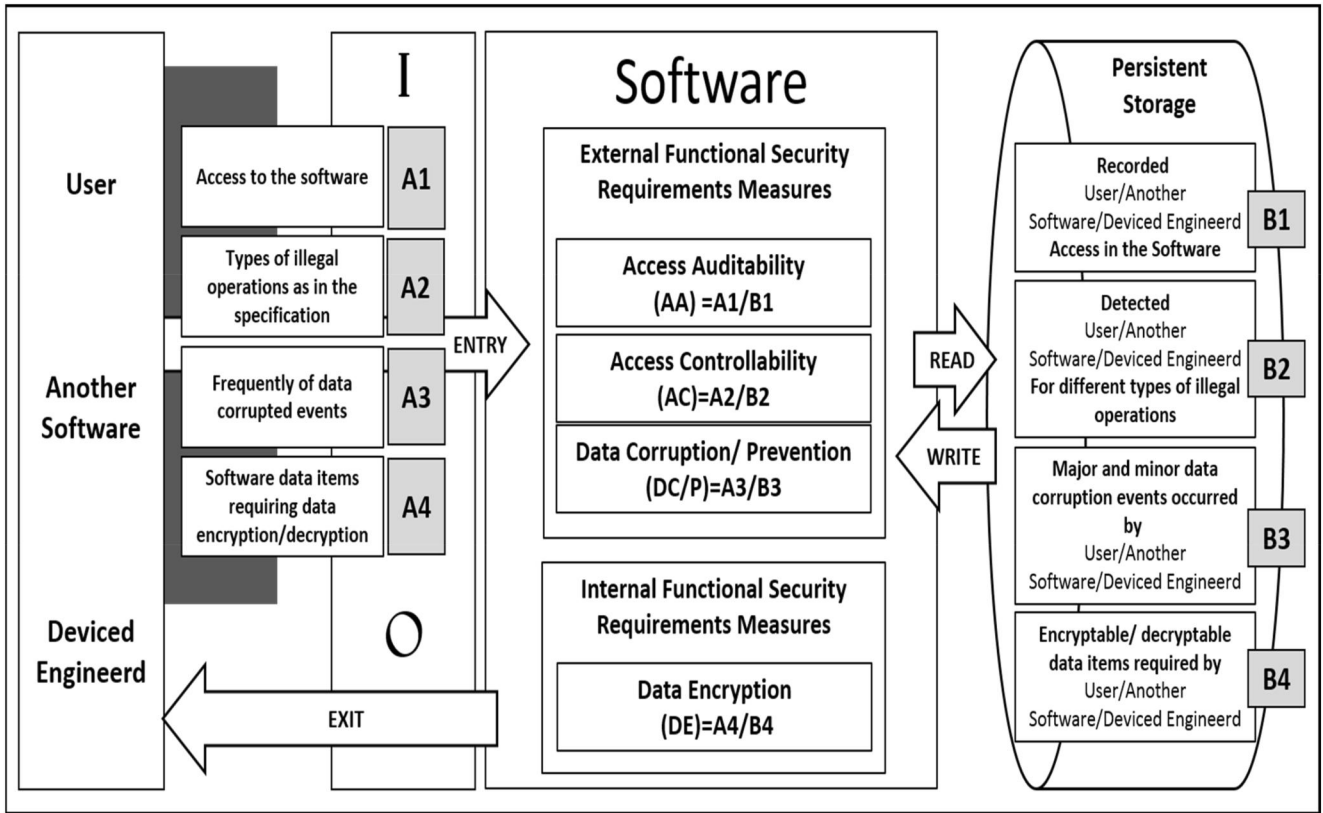


FIGURE 3 A standards-based non-functional security meta-model

ISO/IEC 19761 offers four different types of these data movement as follows: Entry (E), Exit (X), Read (R) or Write (W). Each movement from the four types can be measured by a standard unit called COSMIC Function Point (CFP). One CFP means there is a one data movement in which represent a functional size measurement of each counted data movement types.

4.2 | Identification of data groups

The identification of data groups in this section is used to list the relationship between the data source and destination. Among these relations, there is a set of the object of interest for the type of movements used for the validity when they build the numerical assignment rules using these types of interest, for more details, See-Table 1.

Table 1 illustrates these data groups to be used later by sizing the functional and non-functional security as candidate input for estimation and benchmarking in the future purposes.

4.3 | Functional size measurement of the functional security requirements

This section presents an instantiation case of the functional size measure (FSM) for security. The functional

security perspective according to ISO/IEC 19761 is presented in Figure 2. Using the identification of the functional processes and data movements for functional security requirements for one-process data movements between the source and destination for more detail see Table 2.

Concerning the reference framework, the security functional size (internally and externally) equals 20 CFP for one functional process and considered the data and accesses observed during the security process based on Figure 2 and Table 3.

4.3.1 | The functional size measurement for external functional security for one process is

The functional access auditability (AA) can be measured using Equation (1):

$$\begin{aligned}
 F(AA) &= \sum(\text{Data Movement of Data Groups for AA}) \\
 &= 1E + 1R + 1W + 1X \\
 &= 4 \text{ CFP if the access are recorded by the system}
 \end{aligned} \tag{1}$$

The functional access controllability (AC) can be measured using Equation (2):

TABLE 1 Security data sources and data destinations

Data sources	Data destinations	Objects of interest
User access to the software	Access auditability	Access user
Recorded user access in the software		Access user
Types of illegal operations as in the specifications	Confidentiality	Operation
Detected different types of illegal operations		Operation
Frequently of data corrupted events	Data integrity	Data
The occurred major and minor data corrupted		Data
Software data items requiring data encryption/decryption	Data encryption/decryption	Data
Encryptable/decryptable data items required by the user/another software and devised engineered		Data

$$\begin{aligned}
 F(AC) &= \sum(\text{Data Movement of Data Groups for } AC) \\
 &= 1E + 2R + 2W + 1X \\
 &= 6 \text{ CFP if the access are detected of illegal operations}
 \end{aligned}
 \tag{2}$$

The functional integrity (DC/P) can be measured using Equation (3):

$$\begin{aligned}
 F(DC/P) &= \sum(\text{Data Movement of Data Groups for } (DC/P)) \\
 &= 1E + 2R + 2W + 1X \\
 &= 6 \text{ CFP if the access for occurrences of} \\
 &\quad \text{major/minor data corruption events}
 \end{aligned}
 \tag{3}$$

The total FSM measure for external security can be measured using Equation (4):

TABLE 2 The functional size measurement for functional security for one instantiation case

Functional processes	Data movement description	Data movement type	COSMIC Function Point
Access auditability	Entry when the user access to the software	E	1
	Read and write when the user access to the software by evaluating the number of accesses that the system recorded in the history database	R & W	2
	Exit by recorded user access in the software	X	1
Confidentiality	Entry by the user all types of illegal operations as in the specification.	E	1
	Read and write by the user all types of illegal operations as in the specification.	R&W	2
	Read and write by the user the detected illegal operations	R&W	2
	Exit by the user when the detected different types of illegal operations	X	1
Integrity	Entry by the user the frequently of data corrupted/prevention events	E	1
	Read and write by the user the frequently of data corrupted/prevention events	R & W	2
	Read and write by the user the occurrence major and minor data corruption events	R & W	2
	Exit user from major and minor data corruption events	X	1
Data encryption	Entry when the user requiring data encryption/decryption	E	1
	Read and write by the user the data encryption/decryption	R & W	2
	Exit by the user from the encryptable/decryptable data items	X	1
Total COSMIC 20 CFP			

TABLE 3 Analysis of FSM results for functional security

NO	Functional Processes Description	Number of Data Movements				CFP
		E	X	R	W	
1	Access auditability	1	1	1	1	4
2	Access controllability	1	1	2	2	6
3	Data corruption/prevention	1	1	2	2	6
4	Data encryption	1	1	1	1	4
4 Functional processes		5	5	8	8	20

Abbreviations: CFP, COSMIC Function Point; FSM, functional size measure.

$$F(Ext.Sec) = n * \sum \left(AA + AC + \frac{DC}{P} \right) \quad (4)$$

$$= 4 + 6 + 6 \rightarrow 16 \text{ CFP}$$

4.3.2 | The functional size measurement for internal functional security for one process

The functional data encryption (DE) can be measured using Equation (5):

$$F(DE)$$

$$= \sum \left(\text{Data Movement of Data Groups for } (DE) \right)$$

$$= 1 E + 1 R + 1 W + 1 X$$

$$= 4 \text{ CFP if the DE are recorded by the system} \quad (5)$$

4.3.3 | Functional size measurement for (internal & external) functional security for one process

The functional size measurement for external and internal security for one functional process, can be measured using Equation (6):

$$F(Tot.Sec) = \sum((Ext.Sec) + DE)$$

$$= 16 + 4 \quad (6)$$

$$\rightarrow 20 \text{ CFP, For one functional process}$$

4.3.4 | The total functional size of the security [for all-functional processes]

The functional size measurement for external and internal security for multi-functional processes, can be measured using Equation (7):

$$F(Tot.Sec) = n * \sum((Ext.Sec) + n * DE) \quad (7)$$

n : number of functional processes for the security

4.3.5 | The analysis of measurement results of security

The analysis of measurement results for external and internal functional security is depicted in Table 3.

4.4 | Functional size measurement for the non-functional security for an instantiation case

This section presents an instantiation case of the non-functional measurement size (FSM) for the security as a functional requirements perspective. According to ISO/IEC 19761 in Figure 3 and based on the same instantiation case in the previous section 3.

The non-functional security can be computed based on findings in Figure 3 that is built in Figure 2 and the proposed measures in ISO/IEC 25010 (SQuARE) as follows:

4.4.1 | External non-functional security requirements measures

The functional size of non-functional access auditability (AA) can be measured using Equation (8):

$$NF(AA) = \sum \left(\frac{A1}{B1} \right) \quad (8)$$

A1: User Access to the Software.

B1: Recorded User Access in the Software.

AA: It is the results of the number of user access divided by the number of users recorded in the software.

The functional size of non-functional confidentiality/access controllability (AC) can be measured using Equation (9):

$$NF(AC) = \sum \left(\frac{A2}{B2} \right) \quad (9)$$

A2: User Types of illegal operations as in the specification.

B2: Detected different types of illegal operations.

AC: is the results of the number Types of illegal operations as in the specification divided by the number of Detected different types of illegal operations.

The functional size of non-functional data corruption/prevention (DC/P) can be measured using Equation (10):

$$NF(DC/P) = \sum \left(\frac{A3}{B3} \right) \quad (10)$$

A3: Frequently of Data Corrupted Events by USER

B3: Major and Minor data corruption event occurred.

DC/P is the results of the Frequently of Data Corrupted Events by USER divided by the Major and Minor data corruption event occurred.

4.4.2 | Internal non-functional security requirements measures

The functional size of non-functional data encryption (DE) can be measured using Equation (11):

$$NF(DE) = \sum \left(\frac{A4}{B4} \right) \quad (11)$$

A4: number of Items requiring data encryption/decryption by USER

B4: Actual Number of Encryptable/decryptable data items in the system.

DE: is the results of the number of Items requiring data encryption/decryption by USER divided by the Actual Number of Encryptable/decryptable data items in the system.

4.4.3 | External and internal non-functional security requirements measures

This section illustrates the total non-functional security for external non-functional security and internal non-functional security for Equations (8)–(11), and can be measured using Equation (12) for one functional process and Equation (13) for multi-functional processes.

$$NF(Tot.Sec) = \sum \left(\left(NF(AA) + NF(AC) \right) + NF(DC/P) + NF(DE) \right) \quad (12)$$

for one functional process for nonfunctional security

$$NF(Tot.Sec) = \sum (n * NF(AA) + n * NF(AC) + n * NF(DC/P) + n * NF(DE)) \quad (13)$$

n : number of functional processes for the non – functional security

These values are mapping with ISO/IEC 19761 strategies and procedure of the functional size measurements using the elementary data movements in this standard and sizing by ISO/IEC 19761 unit, which is CFP as defined in Figure 3 for the validity of these results.

5 | EVALUATION OF THE PROPOSED MEASURES FOR SECURITY REQUIREMENTS

The proposed measurements for software functional and non-functional security in this paper are identified separately based on all functionality allocated to software security as a piece of the application in the requirements, whether it has yet to be built or it has already been delivered.

The proposed measurement method is based on the concepts of a functional size, which are the core of the ISO/IEC 14143-1 and ISO/IEC 19761 standards. The security measurement method proposed in this document avoids a number of the weaknesses of different security measures found in the literature by using, as a foundation, a standard method for size measurement, that is, ISO/IEC 19761.

The measurement unit is defined in the proposed measurement of the security with CFP. The generic measurement frameworks for security is built based on the analysis of the content of the ISO/IEC 25021 and ECSS series of standards.

6 | THE PROPOSED REFERENCE FRAMEWORK AMONG SOFTWARE ETALON

In software engineering, concepts of units and etalons have seldom been used, and this is a symptom of the immaturity of the software measures themselves. With regards of the method of designing an etalon aligned with ISO/IEC 14143, the proposed frameworks in this paper are considered as a reference framework for measuring the functional size of the security requirements for the following reasons:

- The definitions and the interpretation of the security requirements are taken from the definitions of security requirements in the European ECSS and international ISO standards. This could be considered as primary material that measures the proposed security requirements as a generic security measures
- A design measurement method used in this paper, including four steps, according to [34] these steps ensure that measurements are performed consistently, a baseline is established as a primary reference
- Using ISO/IEC 19761 standard method to identify the functional size measurement of the security requirements and provide measurement units
- The calibration between steps 1, 2 and 3; the requirements of the proposed security measurement framework, this is equivalent to a measurement instrument or the reference material concerning software etalon

The proposed ‘reference measurement framework’ of security requirements concerning standards etalon offers:

- Security measures, both internally and externally, based on the number of functional processes

- A reference framework that provides a measurement method for each type of security requirements, for example, the measurement of the access auditability, confidentiality, integrity and data encryption/decryption
- Defined interrelations between the internal and external measurements, for example, each process between the internal and external security measurements
- Defined functional size measurements for the software security requirements for all-functional processes (internally and externally)
- Clear traceability of security functional size measures (both internally and externally measures)
- Controlled and repeatable measurement results
- Defined measurement unit

7 | ATM CASE STUDY

An ATM is an automatic banking device that allows clients to do their simple transactions deprived of the assistance of a branch representative or bank clerk. Any clients within a credit card or debit card can access the utmost ATMs. ATMs are useful, permitting clients to complete quick, self-serve transactions from ordinary banking such as 'deposits and withdrawals to more complex transactions such as bill payments and transfers'.

This ATM case study illustrates a process for verification of the proposed framework of the functional and non-functional security using the software specifications of the ATM machine. The ATM is improved to serve bank customers for financial services. Several suggested requirements specifications are chosen in this case study.

7.1 | ATM specifications

This section presents the security requirements for an ATM using a proposed framework. The ATM is a system that helps customers to access their accounts and make numerous transactions such as get cash from the account, add deposits through the ATM, funds transfers or account information inquiries.

In this case study, the subsequent assumptions are made: The ATM assists one bank customer at a time. Each customer is requested to insert his/her bank card into the machine as identification. Next, the ATM asks the customer to enter his/her personal identification number (PIN) on the keypad.

The ATM confirms that the PIN entered matches the one encrypted on the card. Once verified, the customer can access the account to perform the desired transaction. Otherwise, the system displays the appropriate message to explain the denial of access. Figure 4 presents the instantiation block diagram for the ATM system in this example.

7.2 | ATM security specifications

The ATM security specifications in [35–37] define three types of planes: (1) user, (2) control and (3) management planes.

Each plane has many protocol layers: (1) physical, (2) ATM and (3) ATM adaptation layers.

The following security service areas for the above planes are defined: data confidentiality, integrity, and authentication, as well as data, access control. These services are supported in point-to-point and point-to-multipoint connections for the virtual connection and path virtual connections or both of them.

More specifically, the methodology steps used by Ref. [35] are as follows:

- 'User plane authentication' controls the connection at the beginning
- 'User plane confidentiality' affords cryptographic mechanisms to protect the user data on a virtual channel from unauthorized disclosure
- 'User plane data integrity' provides a technique that permits for detection of 'a modification data values or sequences of data values, even in the presence of malicious modification threats'.
- 'Access control data' needs mechanisms to carry out the access control data during connection establishments and some security techniques within ATM components
- 'Control plane Authentication and Integrity' is the ATM security services to binds the ATM message to its source.

7.3 | ATM security specifications within a reference security framework

In this section, the ATM Security ought to postulate mechanisms for user access control, data authentication for the user, data integrity and confidentiality for the user plane.

For more details, see Table 4. Table 4 illustrates the mapping between ATM security specifications and a proposed block diagram for the ATM system [35–37].

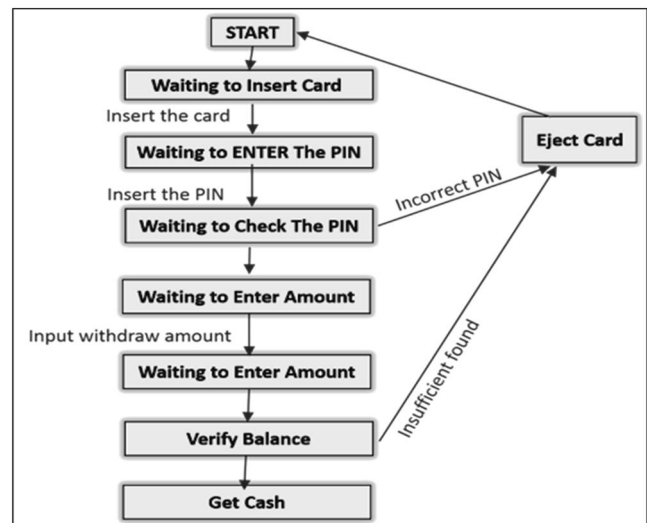


FIGURE 4 Block diagram for ATM machine system

TABLE 4 The mapping between ATM security specification and ATM Block diagram

Security activity for ATM	ATM security specifications in the real-world			A reference measurement framework for security requirements	
	ATM security plane	ATM security protocol layers	ATM security process	Functional security measures using Figure 2	Non-functional security measures using Figure 3
Waiting to insert to card					
Insert the card	User plane authentication	ATM security layer	Establish virtual channel connections	Access auditability	(AA) = A1/B1
Waiting to enter the PIN	User plane authentication	ATM security layer	Establish virtual path connections	Access controllability	(AC) = A2/B2
Enter the PIN	User plane confidentiality	ATM physical layer	Use cryptographic mechanisms	Data encryption	(DE) = A4/B4
Waiting to check PIN	User plane data integrity	ATM physical layer	Use a mechanism that allows for the detection of data values	Data corruption and/or prevention	(DC/P) = A3/B3
Incorrect PIN: Eject the card	User plane data integrity	ATM adaptation layer	Use a mechanism that allows for modification of data values	Data corruption and/or prevention	(DC/P) = A3/B3
Correct PIN	User plane access control	ATM adaptation layer	Use a mechanism for transport access control for information via channels	Access controllability	(AC) = A2/B2
Waiting to enter amount	User plane access control	ATM adaptation layer	Use another mechanism within ATM components to determine whether access to the connection should be granted.	Access controllability	(AC) = A2/B2
Verify balance	Control plane authentication and integrity	ATM adaptation layer	ATM signalling message to its source	Data corruption and/or prevention	(DC/P) = A3/B3
Insufficient balance found: Eject the card	Control plane authentication and integrity	ATM adaptation layer	ATM signalling message to its source	Data corruption and/or prevention	(DC/P) = A3/B3
Sufficient balance found in the card	Control plane authentication and integrity	ATM adaptation layer	ATM signalling message to its source	Data corruption and/or prevention	(DC/P) = A3/B3
Get cash	Control plane authentication and integrity	ATM adaptation layer	ATM signalling message to its source	Data corruption and/or prevention	(DC/P) = A3/B3

Table 5 presents the measurement of system security requirements of the ATM using the reference framework; there are 11 security activity for ATM. These are identified, and they are presented in column #1, The functional and non-functional security in column #2 and the measurements of the functional and non-functional security in column #3. For each identified functional process. The description of the measured resource represents a data security group, which is moved by a one data movement type. Each data movement type that moves one data security group is measured as one CFP.

For instance, this section illustrates how ATM system requirements can be allocated to the security software functions in the proposed framework. The measurement of the data movements using COSMIC in the eight functions presented for the ATM system. And the mapping with the ATM user and Control planes services. According to COSMIC ISO/IEC

19761, the total functional size measurements for the data movements within the functional processes group is 41 CFP.

7.4 | Summary of results

The proposed reference framework of software security requirements is experimented only using the requirements specifications of the withdrawal process for an ATM system. The methodology used in this experiment is built based on the identified FURs (user perspectives) to identify the functional and non-functional security processes. As well as to measure the functional size measurement of the identified security functional and non-functional processes independently of the languages used to develop such a product. The following measures have a unified measurement unit (CFP).

TABLE 5 COSMIC size measurement of security requirements allocated to software in the ATM example for one data movement

Security activity for ATM (#1)	A reference measurement framework for security requirements (# 2)		Data movements identified (#3)								Total size in CFP
	Functional security process-based Figure 2	Non-functional security process-based Figure 3	Functional security measures					Non-functional security measures			
			E	X	R	W	Size in CFP	E	X	R	
Insert the card	Access auditability	(AA) = A1/B1	1	1	1	1	4	A1	B1	AA = 1	5
Waiting to enter the PIN	Access controllability	(AC) = A2/B2	1	1	1	1	4	A2	B2	AC = 1	5
Enter the PIN	Data encryption	(DE) = A4/B4	1	1	1	1	4	A4	B4	DE = 1	5
Waiting to check PIN	Data corruption and/or prevention	(DC/P) = A3/B3	1	1	1	1	4	A3	B3	DC/p = 1	5
Incorrect PIN: Eject the card	Data corruption and/or prevention	(DC/P) = A3/B3		1		1	2	A3	B3	DC/p = 1	3
Correct PIN	Access controllability	(AC) = A2/B2		1		1	2	A2	B2	AC = 1	3
Waiting to enter amount	Access controllability	(AC) = A2/B2		1		1	2	A2	B2	AC = 1	3
Verify balance	Data corruption and/or prevention	(DC/P) = A3/B3		1		1	2	A3	B3	DC/p = 1	3
Insufficient balance Found: Eject the card	Data corruption and/or prevention	(DC/P) = A3/B3		1		1	2	A3	B3	DC/p = 1	3
Sufficient balance found in the card	Data corruption and/or prevention	(DC/P) = A3/B3		1		1	2	A3	B3	DC/p = 1	3
Get cash	Data corruption and/or prevention	(DC/P) = A3/B3		1		1	2	A3	B3	DC/p = 1	3
							30			11	41 CFP

Abbreviation: CFP, COSMIC Function Point.

The total functional size measurement for this instantiation is equal to 41 CFP. More specifically, 30 out of 41 is the measurement results of the functional processes and 11 out of 41 is the measurement results of NFRs.

In this instantiation example for the suggested ATM block diagram, the measurement results for the functional and NFRs appear that almost half of the measurement results are for the data corruption/prevention requirements (DC/P). These means DC/P requirements consume more effort from developers, and they will take more cost than other security requirements. This followed by Access controllability requirements with 11 out of 41, and finally, the access auditability (AA) and data encryption (DC) each have 5 out of 41. In all cases, this size number can be used in the future with cost estimation models or for software benchmarking.

7.5 | Practical implications in very small entities

Software quality becomes of the matter of concerns; The ISO 29110 series [38,39] is developed with explicit purposes to improve ‘product, service quality, and process performance of the software product quality for the very small entities (VSEs) within the system and software life cycle’ [40–41].

In ISO 29110-3 [38] describes in their contexts; the assessment process guiding principle and ‘compliance requirements’ desired for the defined VSEs profiles as well as it is concluded information for developers about assessment methods and tools.

The proposed reference framework of software security can be simply used by developers or even skilled persons in the small companies. The proposed reference model can influence the ISO 29110-3 [38] for those who have direct relations with the assessment process on the VSE profile and need guidance on certifying that the security requirements for performing an assessment have been met. However, the proposed and reference security model can be adopting by ISO 29110-5 [39] at management delivery guideline for the product profile (i.e. provide set of security services delivered to customers). Applying this proposed approach in VSE profiles need additional research work and efficiency studies is required.

8 | CONCLUSION AND FUTURE WORK

This paper introduced a new design security measure for both functional and NFRs as well as internally and externally. A generic functional size framework for security requirements using COSMIC ISO/IEC 19761 standard has also been defined independently of the software type or languages.

Moreover, the design of the measurement method specifies the strategy of the measurement rules to perform the mapping between the concepts of COSMIC and the generic security meta-models and rules to identify the data movements and to perform the measurements.

It is important to remark that the proposed measurement procedure for software security requirements has been developed to apply the COSMIC measurement method, to obtain the functional size of the security as a separate piece of software in the early stages of the software development process.

The proposed generic framework is described as a method of design Etalon standards. Or as the security generic framework for measuring the functional size for security requirements with the Etalon contents and methodology.

The ATM case study is illustrative of how our proposed approach is applicable in a relatively simple context: this corresponds more or less to a 'proof of concept'.

Future research is indeed needed to investigate its scalability to VSE profiles contexts. Of course, for such VSE profiles, organizations dedicate much more resources as well and have more resources to use this proposed approach.

The advantages and the limitations of the generic frameworks also stated at the sections of the paper as future work to enhance the proposed generic frameworks and applicable to use it in the industry.

ACKNOWLEDGEMENT

Hashemite University supported this research. We thank Professor Alain Abran from the École de technologie supérieure (ÉTS) of the Université du Québec for his support that greatly assisted the research.

ORCID

Khalid T. Al-Sarayreh  <https://orcid.org/0000-0002-9373-4577>

Mohammed Zarour  <https://orcid.org/0000-0002-1169-9502>

REFERENCES

- Meridji, K., et al.: System security requirements: a framework for early identification, specification and measurement of related software requirements. *Comput. Stand. Interfac.* 66, 103346 (2019)
- Meridji, K., et al.: Towards A requirements model of system security using international standards. *Int. J. Software Eng. Applicat.* 9(4), 139–164 (2015)
- Abran, A., Al-Sarayreh, K.T., Cuadrado-Gallego, J.J.: A standards-based reference framework for system portability requirements. *Comput. Stand. Interfac.* 35(4), 380–395 (2013)
- Al-Sarayreh, K.T.: Dependability model for decomposition and allocation of system safety integrity levels of software quality. *Int. Rev. Comput. Software.* 10(11), 1110–1119 (2015)
- Al-Sarayreh, K.T., Meridji, K.: Towards a development of an operational process for software requirements: case study application for renewable energy software. *Int. J. Software Eng. Applicat.* 9(7), 11–26 (2015)
- Maqousi, A., et al.: A reference model of security requirements for early identification and measurement of security awareness program. *J. Theor. Appl. Inf. Technol.* 63(1), 74–84 (2014)
- Al-Sarayreh, K.T., Hassan, L., Almakadmeh, K.: A trade-off model of software requirements of balancing between security and usability issues. *Int. Rev. Comput. Software.* 10(12), 1157–1168 (2015)
- Al-Sarayreh, K.T., Abran, A., Cuadrado, J.J.: A standards-based model of system maintainability requirements. *J. Software Evol. Proc.* 25(5), 459–505 (2013)
- ISO/IEC 25010:2011: Systems and software engineering—systems and software quality requirements and evaluation (SQuaRE)—system and software quality models. International Organization for Standardization.
- ISO/IEC 25012:2008: Software engineering—software product quality requirements and evaluation (SQuaRE)—data quality model. International Organization for Standardization.
- ISO/IEC 25021:2012: Systems and software engineering—systems and software quality requirements and evaluation (SQuaRE)—quality measure elements. International Organization for Standardization.
- ISO/IEC/IEEE 29148:2018: International Standard—systems and software engineering—life cycle processes—requirements engineering. International Organization for Standardization.
- ISO/IEC/IEEE 12207:2017: Systems and software engineering—software life cycle processes. International Organization for Standardization.
- ECSS-E-ST-10C:2017: Space engineering: system engineering general requirements & standards division noordwijk, The Netherlands.
- ECSS-, Q-ST-80C:2016: Space product assurance: software product assurance, requirements & standards division noordwijk, The Netherlands.
- ECSS-ESA:2005: Tailoring of ECSS, software engineering standards for ground segments, Part C: document templates. ESA Board of Standardization and Control (BSSC).
- ISO/IEC 14143-6:2012: Information technology—software measurement—functional size measurement—Part 6: guide for the use of ISO/IEC 14143 series and related International Standards. International Organization for Standardization.
- ISO/IEC 19761:2011: Software engineering—COSMIC: a functional size measurement method. International Organization for Standardization.
- ISO/IEC 24570:2018: Software engineering—NESMA functional size measurement method—definitions and counting guidelines for the application of function point analysis. International Organization for Standardization.
- ISO/IEC 20926:2009: Software and systems engineering—software measurement—IFPUG functional size measurement method. International Organization for Standardization.
- Wang, W.: Towards a security requirements management framework for open-source software. In: *proc. of IEEE 26th International Requirements Engineering Conference (RE)*, Banff, AB, pp. 478–483 (2018)
- Yang, Q., et al.: AnFRA: anonymous and fast roaming authentication for space information network. *IEEE Trans. Inf. Forensics Secur.* 14(2), 486–497 (2019)
- Wu, L., et al.: Secure key agreement and key protection for mobile devices user authentication. *IEEE Trans. Inf. Forensics Secur.* 14(2), 319–330 (2019)
- Vistbakka, I., Troubitsyna, E.: Towards a formal approach to analysing security of safety-critical systems. In: *Prod. of 14th European Dependable Computing Conference (EDCC)*, Iași, Romania, pp. 182–189 (2018)
- Zhang, Y., et al.: User security authentication scheme under SaaS platform of enterprises. In: *International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 147–151. Changsha (2018). <https://doi.org/10.1109/ICVRIS.2018.00043>
- Setyoko, Y.A., Yasirandi, R.: Security protection profile on smart card system using ISO/IEC 15408 case study: Indonesia Health Insurance Agency. In: *Proc. Of 6th International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, pp. 425–428 (2018). https://www.tuv.com/indonesia/en/iso-27001-certification.htmlPwt_mc=SEA.Ads.Google.ID20_S01_ISMS.ID20_S01_ISMS_GA.textad.ISMSBMM&cpid=ID20_S01_ISMS_GA
- ISO/IEC 15408-1:2009: Information technology—security techniques—evaluation criteria for IT security—Part 1: introduction and general model. International Organization for Standardization.
- Hovorushchenko, T., Pavlova, O.: Evaluating the software requirements specifications using ontology-based intelligent agent. In: *Proc. Of IEEE 13th International Scientific and Technical Conference on*

- Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, pp. 215–218 (2018)
29. Maskani, J.B., El Ghazi, E.S.: Modeling telemedicine security requirements using a SysML security extension. In: Proc. of 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, pp. 1–6 (2018). <https://doi.org/10.1109/ICMCS.2018.8525939>
 30. Kunakov, E.P.: Improvement of the technological process of pipe bending with the introduction of digital technologies and information security requirements. In: Proc. of IEEE International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS), Saint Petersburg, Russia, pp. 225–229 (2018). <https://doi.org/10.1109/ITMQIS.2018.8525005>
 31. Ahanger, T.A., Aljumah, A.: Internet of things: a comprehensive study of security issues and defense mechanisms. *IEEE Access*. 7, 11020–11028 (2019)
 32. Subburaj, V.H., Urban, J.E.: Applying formal methods to specify security requirements in multi-agent systems. In: Proc. of Federated Conference on Computer Science and Information Systems (FedCSIS), Poznan, Poland, pp. 707–714 (2018). <https://annals-csis.org/proceedings/2018/drp/262.html>
 33. ISO/IEC 19515:2019: Information technology—object management group automated function points (AFP), 1.0. International Organization for Standardization.
 34. Abran, A.: Software metrics and software metrology, 1 edn. 1–328. John Wiley & Sons Interscience and IEEE-CS Press, New Jersey, USA (2010). <https://www.wiley.com/en-us/Software+Metrics+and+Software+Metrology-p-9780470597200>
 35. ATM Forum Technical Committee: ATM security specification. Version 1.1, 1–222. ATM Forum Worldwide, St. Louis, MO (2011). <https://www.broadband-forum.org/technical/download/af-sec-0100.002.pdf>
 36. ISO/IEC 27034-1:2011: Information technology—security techniques—application security—Part 1: overview and concepts. International Organization for Standardization.
 37. ISO/IEC 27034-3:2018: Information technology—application security—Part 3: application security management process. International Organization for Standardization.
 38. ISO/IEC TR 29110-3-1:2020: Systems and software engineering—lifecycle profiles for very small entities (VSEs)—Part 3-1: process assessment guidelines. International Organization for Standardization.
 39. ISO/IEC TR 29110-5-6-3:2019: Systems and software engineering—lifecycle profiles for Very Small Entities (VSEs)—Part 5-6-3: systems engineering: management and engineering guide: generic profile group: intermediate profile. International Organization for Standardization.
 40. Laporte, C.Y., Munoz, M., Mejia Miranda, J., O'Connor, R.V.: Applying software engineering standards in very small entities—from startups to grownups. *IEEE Software*. 35(1), 99–103 (2018)
 41. Laporte, C.Y., O'Connor, R.V.: Systems and software engineering standards for very small entities: accomplishments and overview. *IEEE Comput.* 49(8), 84–87 (2016)

How to cite this article: Al-Sarayreh KT, Alenezi M, Zarour M, Meridji K. A reference measurement framework of software security product quality (SPQ^{NFSR}). *IET Inf. Secur.* 2020;1–15. <https://doi.org/10.1049/ise2.12002>