Investigating Software Requirements Through Developed Questionnaires To Satisfy The Desired Quality Systems (Security Attribute Example)

Sherif M. Tawfik, Marwa M. Abd-Elghany
Arab Academy for Science and Technology and Maritime Transport, P.O. Box: 1029 Miami, Alexandria, Egypt
Sherif226@hotmail.com, marwam@aast.edu

***Abstract*- It is well recognized in software industry that requirements engineering is critical to the success of any major development project. Quality attributes could not be achieved without certain requirements specified by project managers that should be exhibited within the system. Thus, further research is needed to calculate the weighting factors of software quality attributes in an attempt to quantify, or in other words, to measure a software quality attribute from software project specified document. The aim of this paper is to propose a questionnaire that is designed to model one of the software quality attribute which is the security attribute and to illustrate the method used in determining the weighting factor for each question in the questionnaire. The proposed questionnaire is to elicit security requirements and its relative importance in the project under consideration for example: security could not be fulfilled without the presence of appropriate security mechanisms such as authentication, access control, and encryption, and to give a measurement for development efforts on security related feature.**

I. Introduction

In a previous work, the authors utilized the non-functional requirements i.e. quality attributes defined by the project managers, as the basis for software cost estimation (SCE) in order to provide enhanced and more realistic results when undertaking the cost estimation process [1]. The authors introduced a new cost estimation (CE) model as seen in "Fig 1"; using case-based reasoning, with "just enough" of each attribute to satisfy the requirements of the software system.

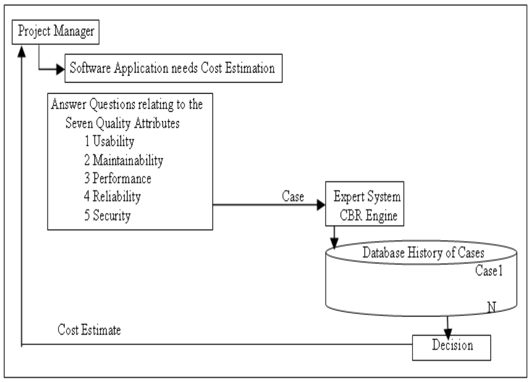

Fig. 1. The Proposed Model

As seen from "Fig 1" that the first step in the cost estimation process is to try to collect the relevant data through answering a group of questionnaires that reflect the quality characteristics that are required in the new software system, and try to quantify these attributes [1].

So that, The main objective of this paper is to give a clear illustration about one of the designed questionnaire that will be utilized to develop the previous work taking the security quality attribute as an example. As an example for the importance of the security in the desired software system, developing web based catalogue services which have on line payment options should be accompanied with secure electronic transaction payment requirements that would acquire policies like cryptographic controls for the protection of the information transmission. It is hoped that this work will act as a useful example to practitioners in the near future for other software quality attributes (like performance, reliability, usability and so on …), but with different questioning criteria.

In this paper, the second section is dedicated to discuss the software requirement specifications. Subsequently in the third section, focusing on security requirement and then in the fourth section the security questionnaire is presented with the explanation of the questionnaire contents and the justification for choosing these questions, also with the demonstration of the data gathering technique

II. Software Requirements Specifications

Software Requirement Specification (SRS) is the initial product development phase in which information is gathered about the requirements. This information-gathering stage can include onsite visits, questionnaires, surveys, interviews, and perhaps a Return-on-Investment analysis or Needs Analysis of the customer or client's current business environment. SRS is basically an organization's understanding of a customer's or client's system requirements and dependencies at a certain time prior to any actual design or development work. It's a two-way insurance policy that assures both the client and the organization understand the other's requirements [2].

The SRS document itself states in precisely and explicitly those functions and capabilities that the software system (i.e. a

978-1-4244-1736-0/08/$25.00 ©2008 IEEE

software application, an e-commerce web site, and so on) must provide, as well as any required constraints by which the software system must abide [3]. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, work statements, testing and validation plans, and documentation plans are related to it [2]. It is important to note that SRS contains only functional and non-functional requirements; it does not offer design suggestions, possible solutions to technology or business issues, or any other information other than the understanding of the development team of what the customer's system requirements meant to be [4].

III. Security requirements

When discussing security requirements, they often tend to be general mechanisms such as password protection, firewalls, virus detection tools, and the like. Studies show that attention to security can save the economy billions of dollars, yet security concerns are often treated as an afterthought to functional requirements.

A recent study found that the Return on Investment when security analysis and secure engineering practices are introduced early in the development cycle ranges from 12 to 21 percent [6]. As reported by [7], software with security faults and poor reliability costs the economy $59.5 billion annually in breakdowns and repairs. [8] defined security requirements as "restrictions or constraints" on system services. [9] described a security requirement as "a manifestation of a high-level organizational policy into the detailed requirements of a specific system" and they remarked that security requirements are a kind of non-functional requirement. [10] stated that a security policy is a document that expresses what protection mechanisms are to achieve and that the process of developing a security policy is the process of requirements engineering. [11] appears to take a similar view, stating that "security requirements mostly concern what must not happen". [12] affirmed that "security constraints define the system’s security requirements". [13] expressed a security requirement as "a quality requirement that specifies a required amount of security in terms of a system-specific criterion and a minimum level that is necessary to meet one or more security policies". These multiple definitions of security requirements are difficult to understand satisfaction criteria, and lack a clear track for deriving security requirements from business goals.

Generally, requirements should specify what data privileges should be granted to the various roles at various times in the life of the resource, and what mechanisms should be in place to enforce the policy [14]. The integrity of the data is determined if the data origin is validated i.e. to ensure that the data arrived unaltered (whether accidental or malicious) therefore, integrity is handled as data origin authentication where a failure in authentication can lead to a violation of access control policy [14]. Confidentiality mechanisms are used to enforce authorization i.e. when a resource is exposed to a user, what exactly is exposed, the actual resource, or some transformation or proxy? This involves encryption, algorithms and parameters for initialization [14]. Resources can be any piece of data or functionality that can be used by a program, including not only application data such as personal information of users, but also many kinds of resources that are often implicit or overlooked in specifying a software system such as: databases, cryptographic key stores, registry keys, web pages (static and dynamic), audit logs, network sockets, any other files and directories [14].

From all of the above it can be said that security requirements represent constraints on functional requirements that are needed to satisfy applicable system's security goals and they express these goals in operational terms, precise enough to be given to a designer.

IV. Security Questionnaire

The authors report here a survey to examine software security requirements at the early stage of the software development life cycle. Interviews handled with a sample size of 40 participants (including requirements' engineers, system designers, software developers, system operational support, software maintenance specialists, testers, etc.) whom currently and previously were engaged in managing software development projects. The participants are staff members from the Information and Documentation Center and also from the Faculty of Computing & Information Technology, within the collaborating institution (The Arab Academy for Science and Technology and Maritime Transport (AASTMT). The participants were selected based on their expertise in programming and analysis.

The questionnaire was passed through two steps. The aim of the first step was to get the potential responses to each interview question to ensure that each question sought sufficient and appropriate to the security field, taking into consideration that these security requirements does not involve hardware nor networking requirements, it does only involve the software security modules. The output from this step was resultant list of questions that was reviewed and modified to respond to comments.

Then, the second step was carried out and a questionnaire was distributed to the same sample. The checklist used in the distributed questionnaire is for eliciting and prioritizing security requirements when developing application projects, This is obtained by judging the degree of relevance of each statement to software security requirements from the participants’ real experience to determine the weighting factor of each question Wf(q_i) = (answer value to question i /summation of all answer values to all questions).

Then using this weighting factor of the question number i to calculate the measure of the software quality attribute in percentage as follows:

$$\sum_{j=1}^{n}[wf(q_i)*wf(ans)_{q_i}] \qquad (1)$$

$Wf(ans)_{qi}$ is the weighting factor of the answer to question number i that would vary according to the responding answers of the project managers depending on the project application type the manager desires to develop; its value would be 0 if he does not want this feature to be included in his required software so it won't cost him and would be equal to (nominal) = 1 if he seeks to apply this feature hence it would cost him.

The questionnaire (see table 1) was divided into four parts as illustrated consecutively below. The participant is being presented to a number of statements to indicate his appropriate response to the importance degree of each statement to security requirements specification by ticking ✓ the suitable number:

5 =Very High, 4 =High, 3 =Fair, 2 =Low, 1 =Very Low. In case of inapplicability of the statement he/she can tick ✓ the Not Applicable (NA) column. For briefing the required system is the new software application under development which:

At first, it should be noted that the Information Systems (IS) that are used to capture, create, store, process or distribute classified information must be properly managed to protect against unauthorized disclosure of information, loss of data integrity, and to ensure the availability of the data and system. Protection requires a balanced approach including IS security features to include administrative, operational, computer, communications, and personnel controls.

Technical protection measures depend on the required system whether it is a single user system or a multi user system. Systems that have one user at a time, but have a total of more than one user with no sanitization between users, are multi-user systems, in which it is allowed that each user of the system to have private files that the other users cannot tamper with or read. Extensive measures are usually inappropriate and inordinately expensive for single-user, stand-alone systems.

As the complexity of a specific IS and the associated residual risk for this system increase, the need for identification and authentication of users and process becomes more significant. Identification and authentication controls are required to ensure that users have the appropriate clearances and need-to-know for the information on a particular system and those controls are divided into two kinds: (a) Unique Identification: Each user shall be uniquely identified and that identity shall be associated with all auditable actions taken by that individual. (b)Authentication at Logon: Users shall be required to authenticate their identities at "logon" time by supplying their authenticator, such as a password, or smart card prior to the execution of any application or utility on the system.

Policies and procedures to detect and deter incidents caused by malicious code, such as viruses or unauthorized modification to software, shall be implemented or not. It is supposed that all files should be checked for viruses before being introduced on an information system and checked for other malicious code as feasible.

Session controls shall be required over identification and authentication or not, for controlling the establishment of a user's session. As for example if the software system provides the capability of tracking successive logon attempts, the following should be done: (a) access denial after multiple repeated unsuccessful attempts on the same user ID, (b) and limitation of the number of access attempts in a specified time period. Furthermore, the required system shall detect an interval of user inactivity then disable any future user activity until the user re-establishes the correct identity with a valid authenticator. All of the previously mentioned requirements are relating to the systems users.

Then coming to the second part, which is dealing with the integrity of the system information sensitivity that would be preserved, the following must be asked. The privileged users required to have access to IS controlling, monitoring or administrative functions. For example: users having "super user", "root", or equivalent access to a system like system administrator, i.e. with complete control of an IS, set up and administer users' accounts and authenticators, users who are given the authority to control and change other users' access to data or program files like database managers, and users who are given special access for troubleshooting or monitoring an IS' security functions like analysts.

After the determination of the system information sensitivity, next requirements should be inquired.

Control of changes to data may range from simply detecting a change attempt to the ability to ensure that only authorized changes are allowed in order to preserve data integrity. Procedures and features are to be implemented to ensure that changes to data are executed only by technically qualified authorized personnel then a transaction log shall be available to allow the immediate correction of all unauthorized data changes at all times. In other words, system recovery functions shall be addressed to respond to failures or interruptions in operation in order to ensure that the system is returned to a condition where all security-relevant functions are operational. If disaster recovery planning is contractually mandated, as in the facility's mission essential applications and information, procedures for the backup of all essential information and software should be identified and the testing procedures as well.

Following program related requirements, the data security requirements should be involved. Security auditing involves recognizing, recording, storing, and analyzing information related to security-relevant activities.

The audit records can be used to determine what activities had occurred and which user or process was responsible for them. Audit records shall be created for the required system to record the following: (a) Enough information to determine the date and time of action, the resources and the action involved; (b) Successful and unsuccessful logons and logoffs; (c) Successful and unsuccessful accesses to directories, including creation, opening, closing, modification, and deletion; (d) Changes in user authenticators; (e) The blocking or blacklisting of a user ID, terminal or access port and the reason for the action; (f) Denial of access resulting from an excessive number of unsuccessful logon attempts.

These contents of audit trails shall be protected against unauthorized access, modification, or detection. Audit analysis and reporting shall also be scheduled, and performed.

TABLE I

Produced Questionnaire from Conducted Interviews

Question Element		Importance Degree				
First: Questions relating to System Users	NA	1	2	3	4	5
1. Supports single user or multi users.						
2. Supports that all the authorized users are uniquely identified before granting access to the system or that all the authorized users are globally identified.						
3. Is capable of stating the number of invalid access attempts that may occur for a given user identifier or access location (terminal or port) and describing the actions taken when that limit has exceeded.						
4. Blocks an account if the password has not been changed within the time limit or the account has remained unused.						
5. Generates logs that contain information about security relevant events such as detection of malicious code, viruses, and intruders (hackers) for example.						
6. Generates logs that contain information about Users relevant events (i.e. identification and documentation of allowed access).						
7. Requires audit logs to be protected from unauthorized access or destruction by means of access controls based on the user.						
Second: Questions relating to Security Administrator by whom the integrity of the sensitivity of all information internal to the system would be preserved	NA	1	2	3	4	5
8. Supports that the security administrator has a choice of enabling or disabling of Users' Identifications.						
9. Supports that the security administrator has the authority of giving grant for accessing specific modules of the systems.						
10. Supports that the security administrator has the authority of giving grant for accessing specific tasks of the systems.						
11. Supports that the security administrator has the authority of giving grant for accessing specific functions of the systems.						
Third: Questions relating to Programs	NA	1	2	3	4	5
12. Requires access control to be established over the system modules.						
13. Requires access control to be established over the system tasks.						
14. Requires access control to be established over the system functions.						
15. Requires any routine program in order to ensure the consistency of the data and its synchronization with the audit logs data.						
16. Requires a policy in case of cryptographic controls for protection of information (i.e. algorithms to transform, validate, authenticate, encrypt or decrypt data).						
17. Requires digital signatures to protect the authenticity and integrity of electronic documents.						
Fourth: Questions relating to Data	NA	1	2	3	4	5
18. Requires its data to be stand alone or shared with other system.						
19. Requires an audit.						
20. Requires an audit for a few of the system tasks or some of the system tasks or a lot of the system tasks or most of the system tasks or all of the system tasks.						
21. Provides audit logs for the capability to investigate unauthorized activities after their occurrences so that proper corrective actions can be taken.						

V. Study result

The pilot study yielded the following results after the data analysis of the distributed questionnaires as shown in table 2. Also, the table contains a scenario for assuming that the project manager conduct a meeting with the user and collect the answers for the security requirements desired in the new software system.

The data in column 2 in table 2 shows the weight for each question which, is supposed to be constants and obtained from the distributed questionnaires as described earlier in the proposed formula in the previous section. Also, the score in the last column illustrates how a software quality attribute such as security would be represented in percentage (i.e. to be put into a quantity); taking into consideration that the given values of each question was calculated based on the answer of each question as shown in the answer column.

The last row in the table represents the final result (60%) after conducting the scenario of answering all the questions. Notes that, this percentage will be varying from system to other as a result of the answers of the questions.

The use of % unit is important since it shows the relative 'saturation' in the attributes with respect to current user requirements thus reflecting the 'maturing' user needs.

VI. Conclusion

This work portrays an unambiguous recognition of the importance of security; the software world within which the argument exist for validating whether or not the system can satisfy the security requirements. Yet, the main contribution of this paper remains in establishing a list of 21 questions that helps the software project manager to cover and acquire all the information needed for the security requirement matter of the required new software system. Moreover, a method for giving a weighing-style quantification of security requirements was presented with a clear example. In the future work, similar questionnaires would be developed to convey the related software features to other software quality attributes in order to quantify them. So that, the main goal for establishing a CBR system that use the quality attribute as a case features for the process of software cost estimation will be achieved.

References

[1] S. M. Tawfiq, M. M. Abd Elghany, S. Green, A Software Cost Estimation Model Based on Quality Characteristics. In MeReP: Workshop on Measuring Requirements for Project and Product Success, IWSM-Mensura, IWSM (International Workshop in Software Measurement) and MENSURA (International Conference on Software Process and Product Measurement), Palma de Mallorca, Spain. November 2007, 13-31. Available at: http://www-swe.informatik.uni-heidelberg.de/home/events/MeRePDocs/paper2.pdf

[2] Jr. Donn Le Vie, Writing Software Requirements Specifications. *Technical Communication Community* ,2007, Available online at: http://www.techwr-l.com/articles/writing/softwarerequirementspecs

[3] S. R. Faulk, Software Requirements: A Tutorial. In Software Requirements Engineering (2^{nd} ed.), *R. Thayer, M. Dorfman (eds.), IEEE Computer Society Press* , Los Alamitos, CA, 1997, 7-22. Available online at: http://www.cs.umd.edu/class/spring2004/cmsc838p/Requirements/Faulk_Req_Tut.pdf

[4] D. E. Jenz, Requirements Packages. *Jenz & Partner* July 2000. Available online at :http://www.bpiresearch.com/Resources/Techniques/requirements_packages.htm

[5] B. Li, Y. Wei, B. Huang, M. Li, M. Rodríguez, C. Smidts, Integrating Software into Probabilistic Risk Assessment. *Final NASA Report*, Center for Risk and Reliability Engineering, University of Maryland, 2006.

[6] K. S. Hoo, J. W. Sudbury, J. R. Jaquith, Tangible ROI through Secure Software Engineering, *Secure Business Quarterly, 1*(2),2001.

[7] National Institute of Standards and Technology: Software Errors Cost U.S. Economy $59.5 Billion Annually. (NIST 2002-10). Available online at: http://www.nist.gov/public_affairs/releases/n02-10.htm

[8] G. Kotonya, I. Sommerville, *Requirements engineering: processes and techniques*. (United Kingdom, John Wiley and Sons, 1998).

[9] P. Devanbu, S. Stubblebine, Software Engineering for Security: A Roadmap. Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, 2000, 227-239.

[10] R. Anderson, Security Engineering: A guide to building dependable distributed systems, (Wiley, 2001).

[11] J. Rushby, Security Requirements Specifications: How and What?. Invited paper, In Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS), Indianapolis, USA, 2001.

[12] H. Mouratidis, P. Giorgini, G. Manson, Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In Proceedings of the 15th Conference on Advanced Information Systems Engineering (CAISE'03). 2003, 63-78.

[13] D. Firesmith, Specifying Reusable Security Requirements. Journal of Object Technology, 3(1), 2004, 61-75.

[14] J. Viega, Building Security Requirements with CLASP, *Proceedings of the 2005 workshop on Software engineering for secure systems—building trustworthy applications* (SESS'05) , St. Louis, MO, USA, 2005, 1-7.

TABLE II

Pilot Study Results

Question Number Q_i	Question Weight W_if	Description		Score
		Selected Feature	Answer	
Q_1	W_1f = 15%	*a)* single user= (0.2) *b)* multi users= (0.8)	b)✓	(15*0.8) = 12%
Q_2	W_2f = 10%	*a)* uniquely identified= (0.8) *b)* globally identified= (0.2)	a)✓	(10*0.8) = 8%
Q_3	W_3f = 5%	a) capable of stating the number of invalid access attempts= (1) b) incapable of stating the number of invalid access attempts= (0)	a)✓	(5*1) = 5%
Q_4	W_4f = 2%	a) blocks an account if it has remained unused= (1) b) does not block= (0)	a)✓	(2*1) = 2%
Q_5	W_5f = 6%	a) generates logs that contain information about security relevant events= (1) b) does not generate= (0)	a)✓	(6*1) = 6%
Q_6	W_6f = 4%	a) generates logs that contain information about Users relevant events= (1) b) does not generate= (0)	a)✓	(4*1) = 4%
Q_7	W_{7f} = 5%	a) requires audit logs to be protected from unauthorized access= (1) b) does not require= (0)	a)✓	(5*1) = 5%
Q_8	W_8f = 2%	a) permits the security administrator to enable or disable Users' Identifications= (1) b) does not permit= (0)	a)✓	(2*1) = 2%
Q_9	W_9f = 3%	a) allows the security administrator to have the authority of giving grant for accessing specific modules of the systems= (1) b) does not allow= (0)	a)✓	(3*1) = 3%
Q_{10}	$W_{10}f$ = 3%	a) allows the security administrator to have the authority of giving grant for accessing specific tasks of the systems= (1) b) does not allow= (0)	a)✓	(3*1) = 3%
Q_{11}	$W_{11}f$ = 3%	a) supports that the security administrator has the authority of giving grant for accessing specific functions of the systems= (1) b) does not support=(0)	a)✓	(3*1) = 3%
Q_{12}	$W_{12}f$ = 4%	a) requires access control to be established over the system modules= (1) b) does not require= (0)	b)✓	(4*0) = 0%
Q_{13}	$W_{13}f$ = 5%	a) requires access control to be established over the system tasks= (1) b) does not require= (0)	b)✓	(5*0) = 0%
Q_{14}	$W_{14}f$ = 6%	a) requires access control to be established over the system functions =(1) b) does not require =(0)	b)✓	(6*0) = 0%
Q_{15}	$W_{15}f$ = 2%	a) requires any routine program in order to ensure the consistency of the data and its synchronization with the audit logs data= (1) b) does not require= (0)	b)✓	(2*0) = 0%
Q_{16}	$W_{16}f$ = 7%	a) requires a policy in use of cryptographic controls for protection of information =(1) b) does not require =(0)	b)✓	(7*0) = 0%
Q_{17}	$W_{17}f$ = 4%	a) requires digital signatures to protect the authenticity and integrity of electronic documents =(1) b) does not require= (0)	b)✓	(4*0) = 0%
Q_{18}	$W_{18}f$ = 4%	a) requires its data to be stand alone or shared with other system= (1) b) does not require= (0)	b)✓	(4*0) = 0%
Q_{19}	$W_{19}f$ = 3%	a) requires an audit= (1) b) does not require= (0)	a)✓	(3*1) = 3%
Q_{20}	$W_{20}f$ = 5%	Requires an audit: a) for a few of the system tasks or some of the system tasks= (0.1) b) or a lot of the system tasks= (0.2) c) or most of the system tasks= (0.3) d) or all of the system tasks= (0.4)	d)✓	(5*0.4) = 2%
Q_{21}	$W_{21}f$ = 2%	a) provides audit logs= (1) b) does not provide= (0)	a)✓	(2*1) = 2%
Total	100%	Software Security Quality Attribute Percentage		60%