CHAPTER VII

DISCUSSION AND FINDINGS

7.1 Introduction

An evaluation of the real subjects is necessary to understand the level of learning and the assistance provided by proposed techniques. The chapter in hand, reports two of the different evaluation methods. These were validated by pilot study and experts' review. Which again were evaluated by building two prototypes, the first one uses the Secure Appreciative Inquiry Fuzzy Quantification Technique (SAIFQT) and the second one uses the normal SDLC. It also contains the details about the penetration test conducted over the period of research, and at the end the comparison of the results of penetration test was done. The participants, both from academia and industry, were asked to complete a series of tasks as required by each of the evaluation methods, SAIFQT and E-Business Online Prototypes, respectively. Each of these evaluation methods was aimed at probing the various features of the support offered by proposed technique. These features range from the support for eliciting and quantifying security requirements in requirements phase, to the support for designing the case study for fuzzy quantification algorithm, penetration testing artifacts and security experts' reports, software experts reports and practitioners reports. The details of each evaluation and validation methods are described below.

7.2 Evaluation and Validation Measures

Since the research is aimed to elicit and quantify security requirements, the main outcome measurements are quantified security requirements. They are measured and

evaluated by using two methods; make a penetration test on two prototypes, one of theme built under the proposed technique SAIFQT steps and the other built on using normal SDLC and compare the results of covered security requirements in both of them. The validation method comprises of quantifying security requirements by pilot study using "IslamTag" website. In addition to what's above it aims to study the variations of attributes in the proposed technique by software experts, security experts and practitioners to get expert's feedback to enhance technique quality and to check the validity. A detailed result of all measures collected during the study is below.

7.2.1. Pilot Study: IslamTag Social Network

The social network "www.islamtag.com" was designated for the application of the study to quantify security requirements worked as follows: Firstly, both of vulnerabilities and errors indexes will be calculated resulting in an input used to find or calculate security index. Secondly, a step of joining the security requirements with security index required to obtain the quantification of security requirements to be called later the security requirements index (See Appendix B).

Following findings were obtained by applying fuzzy quantification algorithm on the real system to validate the algorithm:

a. To quantify security requirements index, need to calculate each of vulnerabilities and errors indexes first, then use this two indexes to calculate security index. The algorithm goes through the previously mentioned steps respectively in a successful way.

b. The results (quantification of security requirements) are identical to NIST and SANS lists. For example, the most dangerous vulnerability in algorithm's results was "SQL Injection" and the most dangerous vulnerability in NIST and SANS lists is "SQL Injection", which is proved algorithm validity.

c. The quantified security requirements were listed as:

1. Command Injection Flaws.
2. Data and Input Validation.
3. OS Command Injection Flaws.
4. Cross Site Scripting.
5. Buffer Overflows.
6. Authentication.

On the other hand, the website developer covers four of security requirements which are:

1. Command Injection Flaws.
2. OS Command Injection Flaws.
3. Cross Site Scripting.
4. Authentication.

According to the arguments mentioned above, the algorithm proves its validity and ability in quantifying new and unique security requirements.

7.2.2. Experts Review

The expert panel is one of the most fundamental components of a Delphi study. The strength of Delphi is the belief that 'n +1' participants are better than one (Crisp et.al., 1997; Verran, 1981). This is particularly useful when there are only a limited number of experts in a field of interest. Linestone and Turoff (2002) suggest the following mix of experts:

- Stakeholders or those who are or will be directly affected.
- Those that have an applicable specialty or relevant experience.
- Those that have skills in organizing, synthesizing and stimulating.
- Interdisciplinary members.

This study selects seven of experts who have deep and thorough knowledge and experience in software engineering and security and they work as doctors in different universities in Malaysia, Jordan and UK. A report was sent to the experts, which

contains details about the proposed technique such as the purpose of the technique, objectives, and the problems on which the study is aimed to solve. This section particularly discusses the proposed technique validation.

According to Scheele (2002), the advantage of using Delphi method is that the study could benefit from subjective judgments based on collective wisdom. The individuals needed to contribute to the examination of a complex problem represent diverse backgrounds and are from wide apart geographical areas. Therefore, their face-to-face meetings were arranged via Skype and e-mail. The validation was conducted by five experts in the area of this study and two developers (see appendix D). Their collective decision suggested adjustments and modifications for the technique.

Moreover, face-to-face or via Skype interviews were conducted to discuss the technique with the experts individually to clarify any ambiguity or misinformation in the report. Later on and at different times, the seven experts sent their reports the validation best to their judgments, and feedbacks about the technique such as characteristics, weaknesses, and a number of notes to update the technique (see appendix D). The thorough study of the reports of the experts on the proposed technique, SAIFQT, reveals a positive assessment of the proposed technique for the requirements of the PhD research. Table 21 below summarizes the expert's notes.

TABLE 21: Summery of experts' feedback

Category	Knowledge Experts					Practitioners	
Name of Expert / Experts' Notes	1	2	3	4	5	1	2
Elicit Software and Security Requirements	√	√	√	√	√	√	√
Quantify Security Requirements	√	√		√	√	√	√
Cover Research Objectives	√	√	√	√	√		
Quantifying Security Requirements Before Building the System	√	√		√	√	√	√
Is the Proposed Technique Help Developers to Capture Security Requirements?	√		√	√		√	√
Succeed Integration Between Software and Security Approaches	√	√	√	√	√	√	√

Most of the experts support a vital feedback and informative reports. It is evident from the table 21 that the proposed technique SAIFQT covers the research objectives. Moreover, all the experts agree on the capacity of the proposed technique to elicit software and security requirements. Moreover, they are one-voiced over the success of the integration between software and security approaches and say that it was done in the right way. Besides, they found that the proposed technique would help developers to capture security requirements that improve secure software industry ultimately. All, except one, experts strongly agree that the most important feature in the proposed technique is the ability to quantify security requirements before building the system, unlike the rest of the techniques; this feature obtained a clear acceptance by the experts.

7.2.3. Develop the Systems

The purpose of the usage of SAIFQT as a security requirements elicitation technique is to elicit a set of security and users` requirements and develop a system. It should be accurately reflecting the security needs and security expectations to the system to be developed. A variety of security requirements are needed to develop a holistic solution that meets the customer requirements and is technically feasible also. Requirements should specifically define functionality, describe quality and non-functional expectations, and consider future goals. This section describes the analysis of the testing and evaluation that has been conducted during this study.

7.2.4. Penetration Test

Penetration testing was implemented in this study, which generally involved in evaluating the developed E-business website which built two times in chapter VI (using normal SDLC and using SAIFQT) against a set of vulnerabilities. The results of the penetration test can be taken as a baseline or a controlling measurement. The penetration test will be done two times, the first, on the prototype that is related to the proposed technique SAIFQT, and second, on the prototype, which is related to the normal SDLC to indicate improvement and register study contributions.

The effectiveness of any security requirements elicitation technique is influenced by the degree that a system is initially defined and developed. Systems that were more defined by their developers using SAIFQT derived more benefit than other additional security requirements techniques. OWASP ZAP tool was used to make penetration test with two different versions of prototypes to execute the penetration testing process (See Appendix E).

The penetration testing was conducted using OWASP ZAP version 2.2.2 Tool; the penetration testing tool requires system (E-business website) to examine how the tasks were carried when attempting to penetrate the website and record any vulnerability that they might encounter and rate the security issues with the developed website.

Penetration test results from the two prototypes that are illustrated in figure 28 and figure 29.

FIGURE 28: Vulnerabilities Alerts for Normal SDLC Prototype.

FIGURE 29: Vulnerabilities Alerts for SAIFQT Prototype.

As figure 28 shows, the vulnerabilities alerts for normal SDLC prototype are four high priority alerts which means that the system is insecure enough or has a high risk i.e. the system is **vulnerable**. In addition, the system has one alert for medium, low, and informational risk.

The figure 29 shows that how much the second prototype (SAIFQT Prototype) is **secure**. This prototype does not contain any high, medium priority alerts which means that the system is secured enough, or does not have a high risk. Besides, it does not have any informational alert, but contains low priority alerts, which pose no danger at

all. The above picture of the fact depicts that the proposed technique SAIFQT has a great potential to contribute in the promotion of secure software industry by developing software that is much more reliable.

7.3 Summary

In this chapter, the results show that the proposed technique, Secure Appreciative Inquiry Fuzzy Quantification Technique (SAIFQT) was proved successfully in eliciting a new and unique software and security requirements, besides quantifying all security requirements for any system through developing the system before its delivery not after. Moreover, the proposed SAIFQT was also used to make the decisions for all elicited security requirements to prioritize and categorize them. The decision for the developers to deliver multi levels of security for any system is established by requirements prioritization and categorization features that were provided by the proposed SAIFQT. The results revealed that the elicited and quantified security requirements would help systems developers to build a strong immune system based on the results of the penetration test for the prototypes.