Towards the Selection of Security Tactics based on No-Functional Requirements: Security Tactic Planning Poker

Felipe Osses, Gastón Márquez, Cristian Orellana and Hernán Astudillo
Toeska research group
Universidad Técnica Federico Santa María
Valparaíso, Chile
Email: felipe.osses@sansano.usm.cl, gaston.marquez@sansano.usm.cl, cristian.orellana.13@sansano.usm.cl, hernan@inf.utfsm.cl

***Abstract*—To achieve security requirements in software design, software architects often adopt security tactics which provide mechanisms to detect, resist, react, and recover from attacks. Nevertheless, there are situations in which the selection of security tactics must be performed in a group manner involving practitioners with different profiles for a more accurate achievement of the security requirements. In this article we propose Security Tactic Planning Poker (SToPPER), a technique that allows stakeholders to interact with each other in order to select security tactics in group form. To validate our proposal, an empirical study was performed in a group of 9 subjects who were presented with three specific non-functional requirements (NFRs) of a particular project. The results revealed that the use of our technique allowed establishing a common basis for the use of security tactics, generating the interaction of all stakeholders involving quick familiarization; a good process of interaction and integration; and the possibility to quickly learn security tactics. At the same time, it was observed that: (1) the subjects performed involuntarily trade-offs between security tactics, and (2) subjects with greater experience selected security tactics with more foundation than the novice subjects.**

***Index Terms*—Planning poker, tactic planning poker, security tactics, gamification**

I. Introduction

Organizations often complain that their applications are hard to maintain, understand, and manage [1]. Software systems are often more complex than thought because the problems it tries to solve are complicated, and the methods involved in addressing the issues are also complicated. This complexity can be described in terms of quality attributes (QAs), which describe the overall performance of a system [2] and represent areas of concern that have wide impact across system layers, where security is a high impact QA for organizations. Is possible characterize QAs from non-functional requirements (NFRs) [3], serving as selection criteria for choosing among alternatives designs and final implementation [4].

Software architects face a constantly growing pressure to build secure software by design. To achieve this goal, security architects work along with stakeholders to identify security requirements and adopt appropriate architectural solutions to address these requirements. These architectural solutions are often based on security architectural tactics. An security architectural tactic (from now, security tactic) is a design decision that influences the achievement of a security response, allowing the evolution of the software architecture or removing obsolete decisions to satisfy changing requirements [2]. Security tactics satisfy security quality attributes regarding resisting attacks (e.g., tactic "Authenticate Actors"), detecting attacks (e.g., tactic "Detect Intrusion"), reacting to attacks (e.g., tactic "Revoke Access"), and recovering from attacks (e.g., tactic "Audit") [5].

Although security tactics help achieve security requirements, decisions that must be made in security concerns should be supported by stakeholders who are embedded in the security issues. Since each stakeholder has a different vision for security requirements, the software architect can analyze other fields that he/she had not previously considered. Therefore, in this article we propose Security Tactic Planning Poker (SToPPER), a card game based on security tactics to involve the stakeholders (who may or may not know the security tactics) in a collaborative and integrated manner in decision-making regarding security requirements.

The structure of the article is as follows: in Section II we describe the motivation to conduct this study. In Section III we illustrate the related work regarding *gamification* in security. In Section IV we describe SToPPER giving the pass to the Section V, where we describe the evaluation performed for SToPPER. In Section VI we detail the results and analysis of the evaluation and in Section VII, we summary the corresponding conclusions.

II. Motivation

Design decisions are relevant statements in software systems that impact the determination of the rationale of the software architecture, allowing capturing its functional and quality requirements [6]. Moreover, architectural design decisions play a significant role in the design, development, integration, evolution, and reuse of software architectures [7]. Capturing design decisions and rationale has several advantages [8], and the acquisition and record of this information is an interesting research topic.

In our previous research [9], we have comprehended that to use architectural tactics, is necessary to create a condition

Presented at: 2017 36th International Conference of the Chilean Computer Science Society (SCCC)

who need to be evaluated when arrives to the system, and these conditions are obtained by the business analysis done by the stakeholders. So, we have realized that architectural tactics can be used to offer design decisions alternatives to the software architect, but depend on NFRs that stakeholders provide. These alternatives allow to software architects to compose more informed decisions to satisfy stakeholder's needs.

Nevertheless, incorrect designs related to security can provoke architecture weakness [10]. Software architecture design is the first and the fundamental step to address quality goals surrounding attributes such as security, privacy, safety, reliability, dependability, and performance. According to [11], estimations indicate that roughly 50% of security problems are the result of software design flaws such as miss-understanding architecturally important requirements, poor architectural implementation, violation of design principles in the source code and degradation of the security architecture. Bad decisions in software architecture systems can have a wide impact on various security concerns in the system and, as a result, giving more space and flexibility for malicious users.

For the reasons described in this section, one of the main challenges that a software architect must face is to consider the most appropriate secure design decisions. Therefore, we believe that a collaborative technique that can involve stakeholders (with different profiles and with or without knowledge of security tactics) in decision-making, can help to better satisfy security requirements. In this way, the software architect will have additional information provided by stakeholders, to select the final decision regarding security concerns.

III. Related work

The use of *gamification* to address security aspects has been extensively addressed. Williams et al. [12] propose a technique called Protection Poker, which is software security game. Its output is a list of each requirement's relative security risk. The team can use this relative risk to determine the type and intensity of design and validation and verification (V&V) effort the development team must include in the iteration for each requirement. The team can then use this list to help prioritize security engineering resources toward software areas with the highest risk of attack based on factors such as how easy the new functionality is to attack and the value of the data accessed through the functionality. Consequently, the team properly estimates the necessary effort to implement the requirement securely, so it can proactively plan which resources are needed for secure implementation. This prioritization and increased knowledge should lead a team toward developing more secure software.

But, other researchers use interesting unconventional tools to satisfy security requirements. Denning et al. [13] create *"Control-Alt-Hack: White Hat Hacking for Fun and Profit"*: a recreational, tabletop card game about computer security. The goal of the authors is generate awareness of security issues and improve the accuracy of people's perception of computer security as a discipline and career choice. They traded some technical complexity in the topics discussed in exchange for increased engagement: put another way, they set out to create a game that players could find inherently fun, from which they might learn incidentally in the course of enjoying the gameplay. Following the same idea, Gondree et al. [14] describe some opportunities for exposing young audiences to cyber security via informal lessons, leveraging play for education and outreach. The authors expose the experience in using the proposal of [13] where they conclude that games inspire players to challenge the limits of play by exploring the meaning and interpretation of rules. Similarly, rule testing, rule interpretation, and rule breaking are prerequisite Red Team skills. They argue that such adversarial thinking is foundational to both strategic games and security engineering. Put another way, many cyber security concepts are game concepts. So, games are natural vectors for teaching the subject matter.

On the other side, Beckers et al. [15] propose to use a card game to elicit security requirements, which all employees of a company can play to understand the threat and document security requirements. The game considers the individual context of a company and presents underlying principles of human behaviour that social engineers exploit, as well as concrete attack patterns. The authors evaluate their approach with several groups of researchers, IT administrators, and professionals from industry. Another alternative is described by Baslyman et al. [16]. The authors propose *Smells Phishy?*, a board game that contributes to raising users' awareness of online phishing scams. They design and develop the board game and conducted user testing with 21 participants. The results showed that after playing the game, participants had better understanding of phishing scams and learnt how to better protect themselves. Participants enjoyed playing the game and said that it was a fun and exciting experience. According to the authors, the game increased knowledge and awareness, and encouraged discussion.

In our revision of related works, we found evidence of using cards games to teaches security issues. Thompson et al. [17] evaluate effectiveness of OWASP Cornucopia, a card game which is designed to assist software development teams, identify security requirements in agile, conventional and formal development processes. They performed an experiment where sections of graduate students and undergraduate students in a security related course at University of North Texas were split into two groups, one of which played the Cornucopia card game, and one of which did not. Quizzes were administered both before and after the activity, and a survey was taken to measure student attitudes toward the exercise. The results show that while students found the activity useful and would like to see this activity and more similar exercises integrated into the classroom, the game was not easy to understand.

Finally, Zad et al. [18] illustrate a complete investigation about collaboration through gaming. They propose a framework that includes both dimensions, initially for investigating existing work and then for implementing a collaborative game. The aim is to demonstrate that such collaborative games are both resolute and entertaining.

Most of the proposals use different card game approaches to assess security aspects. The articles cited in this section stand out for their originality when addressing such a complex

issue as security. However, the evidence shows that there is insufficient information of proposals that use security tactics to address security requirements and support decision-making. This motivates us to propose SToPPER to address security requirements from the point of view of design decisions in software architectures.

IV. Tactic Planning Poker

In this section, we will describe SToPPER, a card game technique which is based on the philosophy of Planning Poker. According to [19] [20], Planning Poker is an agile estimating and planning technique that is consensus based. The rules of Planning Poker are described below [20]:

1) To start a poker planning session, the product owner or customer reads an agile user story or describes a feature to the estimators.
2) Each estimator is holding a deck of Planning Poker cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100, which is the sequence recommended.
3) The estimators discuss the feature, asking questions of the product owner as needed. When the feature has been fully discussed, each estimator privately selects one card to represent his or her estimate. All cards are then revealed at the same time.
4) If all estimators selected the same value, that becomes the estimate. If not, the estimators discuss their estimates. The high and low estimators should especially share their reasons. After further discussion, each estimator re-selects an estimate card, and all cards are again revealed at the same time.
5) The poker planning process is repeated until consensus is achieved or until the estimators decide that agile estimating and planning of a particular item needs to be deferred until additional information can be acquired.

The steps described in the above list illustrate the procedure for using Planning Poker to estimate based on consensus. Having said this, we believe that the philosophy that Planning Poker uses can be extrapolated to estimate alternative design decisions. SToPPER what it does is replace some elements of Planning Poker, but it uses the same philosophy, estimation based on the consensus. In order to adapt Planning Poker to SToPPER, we propose the following cards (See Figure 1):

Each SToPPER card contains the following fields:

- Number: Describes the number of the security tactic
- Name: Describes the name of the security tactic
- Stimulus: Illustrates the incentive to use the corresponding security tactic
- Response: Describes the response associated with the corresponding security tactic
- Priority: Estimates the priority, according to the stakeholder, of the importance of the tactic to satisfy the NFR. The priority range corresponds to 1: Very low, 2: Low, 3: medium, 4: High and 5: Very high.

In Tables I and II we will describe the 17 SToPPER cards. SToPPER uses the security tactics catalog described in Bass et al [2] (see Figure 2).

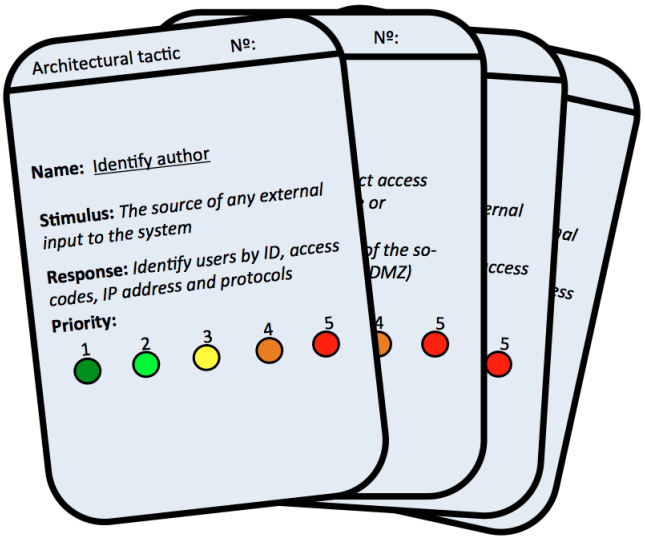

Fig. 1. SToPPER cards

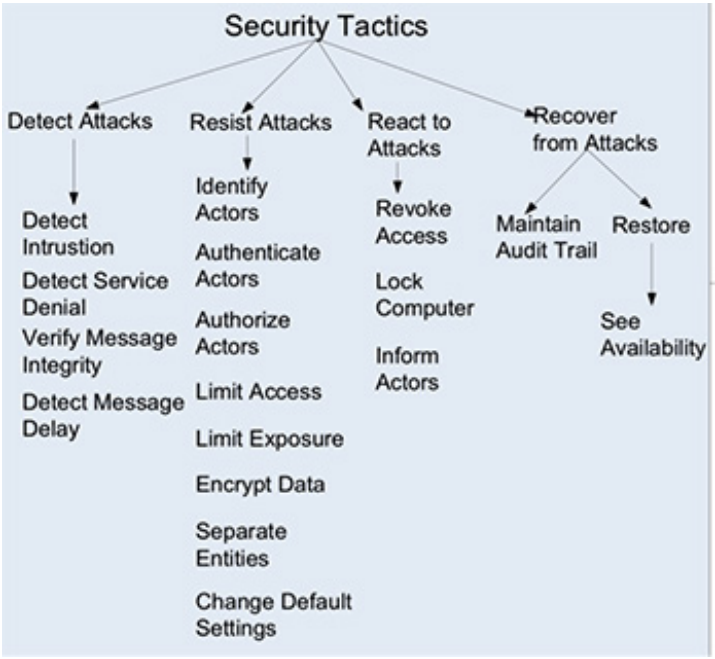

Fig. 2. Security tactics catalog of [2]

For each security tactic, we have summarized them using the fields described in Tables I and II for easy understanding of stakeholders. Finally, we will describe the procedure using SToPPER, which was adapted from Planning Poker:

1) To start a SToPPER session, the moderator reads an NFR.
2) Each participants is holding a deck of SToPPER cards with 17 cards.
3) The participants discuss the NFR, asking questions to the moderator if is necessary. When the NFR has been fully discussed, each participant privately selects one or more cards to represent his or her option. All cards are then revealed at the same time.

TABLE I
STOPPER CARDS LIST (PART I)

C1	**Name**	*Identifying author*
	Stimulus	The source of any external input to the system
	Response	Identify users by ID, access codes, IP address and protocols
C2	**Name**	*Detect intrusion*
	Stimulus	Identify malicious behavior stored in systems
	Response	Detect Malicious Behavior in Protocols, Applications, and others
C3	**Name**	*Detect Denial of Service*
	Stimulus	Denial of service
	Response	Check the configuration of routers and Firewalls to stop invalid IPs
C4	**Name**	*Verify Message Integrity*
	Stimulus	Procedures whose purpose is to alter the integrity of the data of a message
	Response	Procedure to ensure verification of data integrity
C5	**Name**	*Detect delay in message*
	Stimulus	Detect a potential attack by man-in-the-middle
	Response	Detect suspicious behavior
C6	**Name**	*Authenticate actors*
	Stimulus	Ensure that an authorized user has the rights to access and modify data or services
	Response	Define user groups, roles, or individual listings
C7	**Name**	*Authorize actors*
	Stimulus	Ensure that an authorized user has the rights to access/modify data or services
	Response	Define individual user groups, roles, or lists
C8	**Name**	*Limit access*
	Stimulus	Firewalls that restrict access based on source messages or destination ports
	Response	DMZ configuration (demilitarized zone). Providing Internet services but not a private network

TABLE II
STOPPER CARDS LIST (PART II)

C9	**Name**	*Limit Exposure*
	Stimulus	Exploit a single weakness to attack all data and services
	Response	Host services mapping design for limited services to be available on each host
C10	**Name**	*Encrypt data*
	Stimulus	Data must be protected from unauthorized access
	Response	Confidentiality, data protection, Virtual Private Network (VPN), Secure Sockets Layers (SSL)
C11	**Name**	*Separate identities*
	Stimulus	Separation of different servers
	Response	Reduces the chances of attack of those who have access to non-sensitive data
C12	**Name**	*Change default settings*
	Stimulus	The default settings that are on a system
	Response	Prevents attackers from accessing the system through settings that are publicly available
C13	**Name**	*Revoke access*
	Stimulus	Access must be severely limited to sensitive resources
	Response	Protection of sensitive resources
C14	**Name**	*Lock computer*
	Stimulus	Many failed login attempts
	Response	Mechanisms to prevent potential attacks from non-legitimate users
C15	**Name**	*Inform actors*
	Stimulus	Notify a certain actor
	Response	Report when the system has detected an attack
C16	**Name**	*Maintain audit*
	Stimulus	Collect, group and evaluate evidence of attacks
	Response	Trace actions of an attacker
C17	**Name**	*Restoration*
	Stimulus	Restoration of services
	Response	Recovery of an attack

4) If all participants selected the same card, that becomes the most appropriate security tactic. If not, the participants discuss their cards selected. After further discussion, each participants re-selects an card, and all cards are again revealed at the same time.
5) The SToPPER process is repeated until consensus is achieved or until the participants decide that NFR needs to be deferred until additional information can be acquired.

V. EVALUATION OF SECURITY TACTIC PLANNING POKER

In this section, we will describe the pilot test prepared to evaluate the SToPPER process. To do this, it is necessary to establish and plan detailing each step to be performed. First, we establish the objectives to be achieved in this empirical evaluation with the SToPPER process. Then, we present a case study and finally, we design an activity separate in 3 phases (PRE - DURING - POST).

For the design of the activity we are based on a case study of an academic innovation project (described later) in which we have analyzed the most relevant security requirements. This study includes addressing the most important NFRs for stakeholders. This pilot evaluation was presented to a group of professionals dedicated to IT security, who are not familiar with architectural tactics.

A. Objectives and Research questions

The objectives proposed for this evaluation are related to knowing the operation of the SToPPER process and obtaining the first impressions. This will allow us to know the existing strengths and weaknesses of the process and to continue its development. The expected objectives for SToPPER are:

- Rapid familiarization of subjects with the SToPPER process.
- Good interaction and integration of the different participants.
- Quick understanding of architectural tactics.
- Correct selection of tactics by different subjects.
- Collection of valuable data to be used by software architects.

B. Case study: Innovation Management Project

At the Federico Santa María Technical University (UTFSM from now on), the life cycle of a project has critical stages involving various actors, whether internal or external. These stages are: (1) the creation of the initiative or challenge, whether of the company or academic, (2) the development of new ideas or the use of a portfolio of initiatives that could solve this problem, (3) the preliminary draft or feasibility study, (4) the investigation and execution of the chosen solution and, finally, the possible patents, licenses and spin-offs, which could result in finalizing this project.

However, the current situation does not allow agile communication to coordinate support for potential or developing projects in different aspects. In the UTFSM there are systems of financial assistance for projects, management, application of patents, and various consultancies. But, these systems are limited to the use of specific units, and they use own systems that hinder effective coordination. The above also restricts the access to the information to the stakeholders on some of the potential projects, in development or finalized stages. The same situation is manifested when somebody wants to access records of entrepreneurship initiatives that emerged as the product of some result or project.

Given the above description, we are currently working on the proposal of architecture with the aim of showing all the potential work of the UTFSM to the community. Because the objectives of the project require that there is a communication between the UTFSM and external systems, there are security requirements that must be satisfied as a requirement of the project. For the above and this comparative study, we select the three most important security NFRs to know what security tactics can help us to make better design decisions in the architecture in which we are currently working. Next, the three security NFRs descriptions are detailed:

- NFR1: The platform will use a publish/subscribe architecture, where the messages to be posted must be transformed into the desired format and addressed to one or more subscribers to communicate the initiatives of the UTFSM. For this, this communication should be based on the following aspects: confidentiality of information, integrity, authentication and access management under UTFSM standards.
- NFR2: The platform will have an SOA architecture that will communicate with the services offered by UTFSM systems through web services. This communication is required to use security mechanisms, such as WS-Security.
- NFR3: The platform must guarantee mechanisms to promote confidentiality and integrity in communications with internal systems of the UTFSM and with external platforms.

C. Evaluation Design

The design includes three phases. The prior phase to the exercise, the phase in which the exercise is performed and the final phase of collecting data called PRE - DURING - POST respectively.

The characteristics of the evaluation were:

- Activity guided by two people, a moderator and an assistant/record.
- The moderator is in charge of guiding each phase of the activity
- The assistant must support the moderator at all times and ensure the greatest possible number of registrations
- 9 subjects participated in the pilot SToPPER process
- Only security tactics were used
- 17 cards were used
- Three security NFRs were used based on an academic project

D. Exercise procedure

1) PRE phase: In 1 hour we introduce SToPPER technique. A presentation was made to the 9 subjects of the exercise, which we spent 20 minutes explaining the security tactics, 20 minutes in the SToPPER process, 10 minutes in the operation of the cards and 10 minutes in all the details regarding the evaluation. Subsequently, there was a process to answer questions.

It was appreciated from the subjects a great motivation of participating in an academic experience uncommon to their daily activities and thus be part of the empirical development of a new technique. In addition, the possibility of continuing the process of knowledge exchange in the future was opened.

2) DURING phase: In this phase we performed the exercise, which we spent 1.5 hr and it was the next week from the previous phase. In DURING phase, the case study was presented together with the three selected NFRs. Each NFR was read by moderator and then, were given 10 minutes to perform their selection of cards with the corresponding security tactics. After 10 minutes, each person had to argue the reason for each card selected, a process that spent 20 minutes and was guided by the moderator.

3) POST phase: In this phase we collected data and we performed an analysis of the results. The collection, storage and analysis of the obtained information was carried out. This phase was the longest and we carefully reviewed each card selected by each subject according to the NFR. In addition, the audio of the DURING phase was analyzed in detail and a subsequent meeting was held with the subjects who participated in the evaluation to obtain their feedback.

VI. Results and Analysis

A. Selection of cards according to percentage

Tables III, IV, V and VI show the results obtained using SToPPER in the empirical evaluation.

- Table III illustrates those security tactics that were selected by the 9 subjects for each of the NFRs.
- Table IV describes those security tactics that, on average between a range of 50% and 99%, were selected by for each of the NFRs.
- Table V summarize Table III and IV.
- Table VI shows those security tactics that, on average between a range of 1% and 49%, were selected for each of the NFRs.

TABLE III
SECURITY TACTICS SELECTED BY ALL SUBJECTS - 100%

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
NFR 1						x	x										
NFR 2		x								x							
NFR 3		x				x	x	x		x						x	
Selected tactics = 6																	

TABLE IV
SECURITY TACTICS SELECTED BY 50% OR MORE SUBJECTS AND LESS THAN 99%

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
NFR 1	x	x						x		x	x	x	x		x	x	x
NFR 2			x					x								x	x
NFR 3	x		x	x							x	x	x		x		x
Selected tactics = 12																	

TABLE V
SUMMARY OF SELECTED SECURITY TACTICS BY MORE THAN 50\% OF THE SUBJECTS

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
Select tactics= 14																	

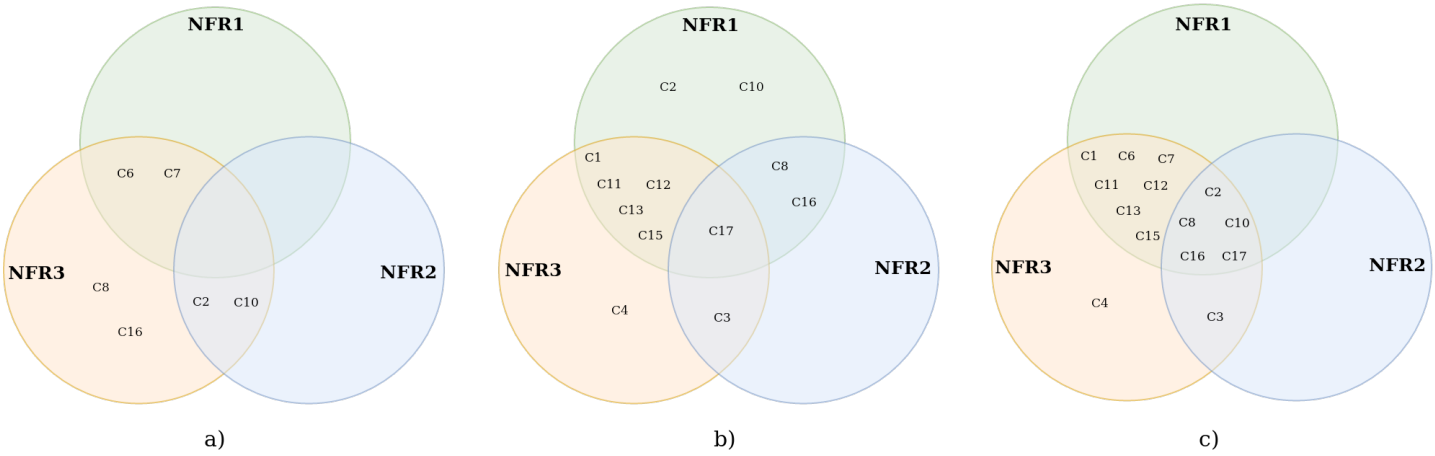

Fig. 3. Visualization of a) security tactics selected by all subjects - 100%, b) security tactics selected by more than 50% of the subjects and less than 99% and c) summary of selected security tactics by more than 50% of the subjects

1) Security tactics selected by all subjects – 100%: In Figure 3-a), is possible to appreciate that the most observed category of tactics was *resistance to attacks* (reflected in tactics C6, C7, C8 and C10). In this phase the subjects located their analysis based on the textual description of the NFR. At this point, the moderator intervened rarely in the discussion of the subjects. It should be noted that there is no intersection between the three NFRs.

2) Security tactics selected by more than 50% of the subjects and less than 99%: Figure 3-b) represent the most recurrent decisions made by the subjects. It can be observed that the category *recover from attacks* represents a greater relevance for the subjects, where tactics C17 (*restore services*) predominates in all three NFRs, as well as tactic C16 (*maintain audit*) selected for NFR1 and NFR2.

In addition, it is important to note that the subjects recognize that for the region that intersects NFR1 and NFR3, tactics C13 and C15 are chords to react to attacks.

3) Summary of selected security tactics by more than 50% of the subjects: Figure 3-c), illustrate the consolidation of Figure 3-a) and 3-b). The subjects revealed that for the proposed NFRs, it is necessary to consider all categories of security tactics. For example, we highlight the tactics C2, C8, C10, C16 and C17, which are transversal to all NFRs. Also, it can be seen that the subjects considered a great similarity between NFR1 and NFR3, where they share 12 security tactics.

B. Selection of cards according to NFRs priorities

SToPPER allow subjects to select not only security tactics, but also a priority associated with each tactic. In this sense, it was observed that for each tactics selected, priorities used were between 3 and 5, excluding priorities 1 and 2. Based on this evidence, the subjects were consulted about their criteria. They responded that they felt that priorities 1 and 2 were similar to discarding the card, which is why they prefer not to use them.

With respect to the prioritization of tactics, it was possible to see that there was a trend towards a higher prioritization (4.79) (see Table VII) in the segment of security tactics selected by all subjects, whereas in security tactics selected by more than 50%

TABLE VI
SECURITY TACTICS SELECTED BY 1% OR MORE SUBJECTS AND LESS THAN 50%

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
NFR 1			x	x	x				x					x			
NFR 2	x			x	x	x	x	x	x		x	x	x	x	x		
NFR 3					x				x					x			
Selected tactics=13																	

TABLE VII
PRIORITY OF SECURITY TACTICS SELECTED BY ALL SUBJECTS - 100%

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	$\overline{X}$
NFR 1						4.88	4.66											4.78
NFR 2		4.77								4.89								4.83
NFR 3		4.67				4.78	4.89	4.78		5						4.22		4.76
																		$\overline{X}$ priority= 4.79

TABLE VIII
PRIORITY OF SECURITY TACTICS SELECTED BY 50% OR MORE SUBJECTS AND LESS THAN 99%

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	$\overline{X}$
NFR 1	4.75	4.4						4.67		4.3	3.8	4.8	4.2		4.5	4	4.6	4.4
NFR 2			4					4.7								4.3	4.2	4.3
NFR 3	5		4.3	3.7							4.5	4.25	4.5		4.75		4.6	4.44
																		$\overline{X}$ priority= 4.38

of the subjects and less than 99 %, the average priorities were 4.38 (see Table VIII). From the above reason, it is possible to mention that when all the subjects agreed with a particular security tactic, they also did so with the selected prioritization level.

To obtain the average of the priorities in this segment, a priority ratio was used. This ratio considers the priority established by tactics versus the number of subjects that selected that tactic.

C. Interventions made by subjects

The Table IX shows the number of interventions per subject. Interventions are those suggestions performed by the subjects at the moment of give an opinion for each NFR.

TABLE IX
NUMBER OF INTERVENTIONS PER SUBJECT

Interventions	
Subject	N° interventions
1	12
2	6
3	8
4	8
5	6
6	4
7	5
8	7
9	6
Total	62
Average	6.888889

The interventions were counted manually after the exercise. The average number of interventions observed was 6.9 interventions per subject. By observing each intervention in detail, it is possible to determine that the SToPPER process allows the interaction and intervention of all participants. It also allowed us to verify that there was a quick familiarization for most of the issues related to NFRs, which allowed them to participate in a group form, giving different points of view.

***Key finding (1)**: SToPPER allows to the subjects to establish trade-offs between security tactics*

An interesting finding that we observed at the moment of SToPPER was executed is that the process allowed the subjects to discuss which security tactic is most appropriate for each NFR. For example, subjects with more experience corrected novice subjects because they did not know the value of the meaning of security tactics. Experts could see that certain security tactics *shut down* (they used that term) to other security tactics when analyzing NFRs. This type of analysis allowed the experts to select a low amount of security tactics and the novices, a high amount.

***Key finding (2)**: Subjects with work experience select security tactics more accurately than subjects with no experience*

Clearly, experience in selecting security tactics plays a fundamental role. The expert subjects selected security tactics not only thinking about the design decision regarding security, but also in the context of the business. They appreciated that the selection of security tactics has a close relationship with the objectives of the business that wants to achieve the project under study.

D. General view of project's software architects

The experience with SToPPER for this case study revealed that the technique was a contribution to the project security decision-making. Once the results of SToPPER were presented to the software architects, they gave their feedback about it. For NFR1, the most attractive security tactics were C6 and

C7. From the point of view of data protection, the architects had a notion of authenticating and authorizing users. However, seeing that the subjects of the studies also coincided with their vision, they are now more certain that the final design of the project must involve these resistance mechanisms of attacks. For NFR2, the same situation occurs. The security tactics selected by the subjects also coincided with the decisions taken by the architects. From the beginning, they had decided that implementing audit-related decisions could help them recover faster from potential attacks on the project. However, there was much discrepancy for NFR3. The security tactics selected by the subjects did not satisfy the decisions that the architects thought. Because SToPPER was applied in a pilot test, the moderators were unaware of aspects of the business that the architects already knew. Therefore, the architects agreed that SToPPER should consider business guidelines within the SToPPER process.

VII. Conclusions

In this work, we propose the Security Tactic Planning Poker (SToPPER) technique based on Planning Poker philosophy, whose objective is the selection of security tactics in a collaborative and integrated manner based on consensus.

SToPPER presents 17 card related to the 17 security tactics described by Bass et al [2]. Each of the cards contains 5 fields: number, name, stimulus, response and priority. The procedure considers the approach of the NFR to the stakeholders. Each stakeholder will have the possibility of privately select the tactics that they consider appropriate to satisfy the security requirement. Subsequently, all stakeholders will reveal the selected cards at the same time and then each subject will have the possibility to argue the reason for the selection of that card. Finally, when they have completed their participation, they can modify their decisions based on the discussion and then record the final results. This procedure is performed for each existing NFR.

In order to verify our proposal, we prepared an empirical evaluation that allowed us to verify the following expected objectives: rapid familiarization of subjects with the SToPPER process; good interaction and integration of the different participants; quick understanding of architectural tactics; proper selection of tactics by different subjects and collection of valuable data to be used by software architects.

To do the empirical evaluation, we present a case of study with three security NFRs too a group of professionals dedicated to IT security, who are not familiar with security tactics. The activity include three phases: the prior phase to the exercise, the phase in which the exercise is performed and the final one of collecting data called PRE - DURING – POST respectively.

The results obtained revealed that SToPPER allows to fulfill the imposed objectives, showing the subjects' quick familiarization, good process of interaction and integration as well as the possibility to quickly learn the security tactics, the ones with which they were able to select the ones that support an actual architect. Besides that, SToPPER allows the subjects to establish trade-offs between security tactics, and was observed that the subjects with work experience showed a more accurate selection of security tactics than subjects with less experience.

Our future work considers the validation of SToPPER through techniques associated with experimental software engineering.

Acknowledgments

This work has been financially supported by (1) Comisión Nacional de Investigación Científica (CONICYT) CONICYT-PCHA/Doctorado Nacional/2016-21161005 and (2) FONDECYT regular 1140408. We also appreciate the support of the Chilean Navy and Stephanie Sherman Leinenweber to help in this investigation.

References

[1] E. Ogheneovo, "On the relationship between software complexity and maintenance costs," *Journal of Computer and Communications*, vol. 2, no. 14, 2005.

[2] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice (3rd Edition)*. SEI Series in Software Engineering, 2013.

[3] M. K. Bruce R. Maxim, "An introduction to modern software quality assurance," *Software quality assurance: in large scale and complex software-intensive systems. Morgan Kaufmann.*, pp. 19 – 46, 2015.

[4] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*, vol. 5. Springer Science and Business Media., 2012.

[5] M. Mirakhorli, "Tactical vulnerabilities in chromium, php and thunderbird," *IEEE Software Blog*, 2017.

[6] D. Tofan, M. Galster, P. Avgeriou, and W. Schuitema, "Past and future of software architectural decisions – a systematic mapping study," *Information and Software Technology*, vol. 56, no. 8, pp. 850 – 872, 2014.

[7] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions," *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, pp. 109–120, 2005.

[8] J. Tyree and A. Akerman, "Architecture decisions: Demystifying architecture," *IEEE Software*, vol. 22, no. 2, pp. 19 – 27, 2005.

[9] G. Márquez and H. Astudillo, "Selecting components assemblies from non-functional requirements through tactics and scenarios," *35th International Conference of the Chilean Computer Science Society, SCCC*, 2016.

[10] J. C. da Silva Santos, "Toward establishing a catalog of security architecture weaknesses," *Thesis. Rochester Institute of Technology. Accessed from http://scholarworks.rit.edu/theses/9004*, 2016.

[11] M. Mirakhorli, "Common architecture weakness enumeration (CAWE)," *IEEE Software Blog*, 2016.

[12] L. Williams, A. Meneely, and G. Shipley, "Protection poker: The new software security "game"," *IEEE Security Privacy*, vol. 8, no. 3, pp. 14–20, 2010.

[13] T. Denning, A. Lerner, A. Shostack, and T. Kohno, "Control-alt-hack: the design and evaluation of a card game for computer security awareness and education," *Proceedings of the 2013 ACM SIGSAC conference on Computer communications security*, pp. 915–928, 2013.

[14] M. Gondree, Z. N. Peterson, and T. Denning, "Security through play," *IEEE Security Privacy*, vol. 11, no. 3, pp. 64–67, 2013.

[15] K. Beckers and S. Pape, "A serious game for eliciting social engineering security requirements," *IEEE 24th International Requirements Engineering Conference (RE)*, pp. 16–25, 2016.

[16] M. Baslyman and S. Chiasson, ""Smells phishy?": An educational game about online phishing scams," *APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–11, 2016.

[17] M. Thompson and H. Takabi, "Effectiveness of using card games to teach threat modeling for secure web application developments," *Issues in Information Systems*, vol. 17, no. 3, 2016.

[18] D. D. Zad, M. C. Angelides, and H. Agius, "Collaboration through gaming," *Handbook of Digital Games*, no. 784, 2014.

[19] K. Moløkken-Østvold, N. C. Haugen, and H. C. Benestad, "Using planning poker for combining expert estimates in software projects," *Journal of Systems and Software*, vol. 12, no. 81, pp. 2106–2117, 2008.

[20] Mountain Goat Software, "Planning poker," *https://www.mountaingoatsoftware.com/agile/planning-poker*, 2017.