How to Select a Security Requirements Method? A Comparative Study with Students and Practitioners

Fabio Massacci and Federica Paci

Department of Information Engineering and Computer Science, University of Trento
name.lastname@unitn.it

Abstract. Most Secure Development Software Life Cycles (SSDLCs) start from security requirements. Security Management standards do likewise. There are several methods from industry and academia to elicit and analyze security requirements, but there are few empirical evaluations to investigate whether these methods are effective in identifying security requirements. Most of the papers published in the requirements engineering community report on methods'evaluations that are conducted by the same researchers who have designed the methods.

The goal of this paper is to investigate how successfull academic security requirements methods are when applied by someone different than the method designer. The paper reports on a medium scale qualitative study where master students in computer science and professionals have applied academic security requirements engineering methods to analyze the security risks of a specific application scenario. The study has allowed the identification of methods' strenghts and limitations.

1 Introduction

The OWASP CLASP project [20], Microsoft SDL [15], and Cigital's Touchpoints [13] are examples of Secure Development Software Life Cycles (SSDLCs) whose target is the development of secure software. Those processes identify as preliminary step the collection of software's security requirements. Security management standards such as ISO-2700x, COBIT [9], or the NIST standard [17] propose very similar processes where the initial phase is the collection of requirements.

A number of academic methods [6,16,4,14,11,10] have been proposed to elicit and analyze security requirements, but there are few empirical and comparative evaluations that help to select a method rather than another. A number of papers in the academic literature usually present a single, not repeatable experiment according to the terminology from [3] where the designer to show the effectiveness of a proposed method, applies the method to a more or less complex scenario.

However, the only way to investigate the actual effectiveness of academic security requirements methods is to conduct empirical studies.

A. Jøsang and B. Carlsson (Eds.): NordSec 2012, LNCS 7617, pp. 89–104, 2012.

This paper presents a qualitative study that we have conducted to investigate whether academic methods are effective in identifying security requirements, and what and why makes these methods effective.

The study involved master students in computer science and professionals in IT Audit for Information Systems who have no previous knowledge in the methods. The empirical evaluation consists of an initial *Training* phase where the participants are instructed about a specific security requirements and risk analysis method, and an *Application* phase where the groups of participants apply the method to identify the security issues of real industrial application scenarios. Each group represents a team of security practitioners that are hired by a company to analyze the security risks of the company using one of the security requirements methods under evaluation. We have collected data on the methods' effectiveness using different data sources. We have video-audio recorded the participants during the application of the methods, we collected the artifacts generated by each group, and administered a number of questionnaires during the different phases of the study execution. We have also conducted focus group sessions with the groups at the end of the Application phase. The analysis of the collected data has allowed us to identified strenghts and limitations of the methods under evaluation.

In the next section (§2), we describe the design our study to compare the different security requirements and risk analysis methods. Then, we introduce the first run of the study that was conducted in 2011 (§3), and the participants and designers that we have recruited (§4). In Section 5, we describe the results of the data analysis. At the end we present related works (§6), discuss the threats to validity in (§7), and conclude the paper (§8).

2 Research Design

We have used qualitative research as main research method because it is suitable to answer research questions of the type *how*, *what*, *why*. In our study we want to investigate *how* well do academic security requirements methods actually work when applied by someone different than the designer, *what* aspects make these methods work, and *why*. Thus, we formulate our research questions as follows:

- **RQ1:** *How effective are academic methods to elicit risk and security requirements when applied by a person different than the designer?*
  - **RQ1.2:** *Can a novice to an academic security requirements engineering method easily apply it?*
- **RQ2:** *Which factors do make the methods effective? Which one don't?*
- **RQ3:** *Why these factors make the methods effective? Why they don't?*

2.1 Evaluation Protocol

Since we run the study with subjects novice to the methods, we have distilled an evaluation protocol that consists of the following phases:

Table 1. Main actors of the evaluation protocol

Role	Description
Customer	provides the application scenario for the analysis. It is responsible of providing to participants all the relevant information about the scenario.
Method designer	gives tutorials on the method to aprticipants and remains available for questions connected to the method.
Observer	has to take notes about the behavior of the groups during the Application phase and mediates the interaction between the participants and the method designers.
Participant	conducts the analysis of risk and security issues of the scenario provided by the customer, by using one specific method provided by one method designer. The participant should not have any prior knowledge about the assigned method.
Organizer	is in charge of the evaluation, keep the contacts among the actors, and organize the data collection and analysis.

- **Training.** Participants attend training sessions, in the form of tutorial lectures, about the method they are going to work with. After the training session, participants receive an information package containing the scenario and the instructions on the materials they are asked to produce during the analysis. Participants then are given some time to get familiar with the method.
- **Application.** Participants work in groups and apply the method that was assigned to them on a scenario provided by customers; the group collaboration can take place both face-to-face or remotely by using multiple communication channels (e.g. mail, chat, video conferencing facilities) for supporting the group-work. The Application phase ends with the delivery, by each group of participants, of a final executive report.
- **Analysis.** The organizing team takes care of the data analysis and of the comparative evaluation of the methods. A report of the results of the evaluation is shared with all designers.

The main actors involved in our protocol are illustrated in Table 1. During the Training phase, designers and participants are the only actors that really need to be involved. Collection of material can be done easily by video or audio recording, in particular if we use classical lecture style presentation. On-line, web-based tutorials allow for even richer data collections by means of logs and screen recordings.

The Application phase is the moment where customer and observers play a major role in the process. The customer is there to answer all possible questions that may arise out of the participants analysis (starting from "which legislation does apply?" to "do you already have a SSL server?"). observers are also important. Even if we audio-video record the groups at work, there are "social" events that can be better captured by human observers. A simple example is a change in the group internal organization that was agreed during a coffee break; when participants return to the experiment, two people work on the mitigation strategy while one works on the risk assessment. As a result, the audio recording shows 50 minutes of silence out of which a final executive reports is produced.

Table 2. Data sources

Data Source	Description
Questionnaires	Questions ranged from information about participants' knowledge of IT security and risk assessment methods and their evaluation of the different methods.
Audio/Video Recordings	Audio-video files of the Application phase and Focus Groups discussions. Video and audio recording are transcribed and annotated to identify common patterns of behavior.
Method's Artefacts	Graphs, drawings, diagrams, notes, produced by the groups during Application phase.
Post-it Cloud	Post-it where participants were required to list the five aspects they consider to be particularly positive or negative about the method and about the evaluation procedure.
Focus Group Discussions	Participants discuss with method designers a number of topics related to the method, its application on the given scenario and the process of evaluation.
Group Presentations	Presentations given by groups in front of the method designers, the members of the organizing team, and all the other participants.
Final Reports	A 10-page final recommendation including some information about the analysis

Different types of data (listed in Table 2) should be collected during the study by using the techniques and instruments typically used in the Social Sciences for observing and measuring participants' behavior, attitudes and opinions.

3 The Actual Protocol Run

The first study took place in May 2011 as shown in Figure 1, while a second study is taking place at the time of writing.

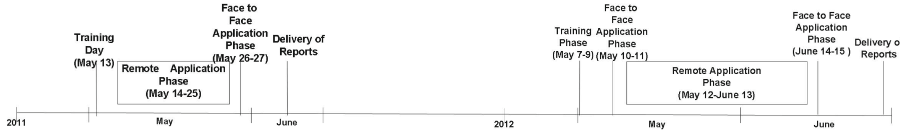
Training Day (May 13)
Remote Application Phase (May 14-25)
Face to Face Application Phase (May 26-27)
Delivery of Reports
2011
May
June
Training Phase (May 7-9)
Face to Face Application Phase (May 10-11)
Remote Application Phase (May 12-June 13)
Face to Face Application Phase (June 14-15)
Delivery of Reports
2012
May
June

Fig. 1. Chronology of the comparative case study

During the *Training* day, we have introduced the participants to the aims, procedure, the expected outcomes of the study, and to the application scenario. The chosen application scenario was about the Healthcare Collaborative Network (HCN) by IBM [1], which is a "health information infrastructure for interconnecting and coordinating the delivery of information to participants in the collaborative network electronically". HCN was applied to a fictional Healthcare system based in Cityville (France). In the fictional set up, the CEO of the Healthcare System (the customer) hired the teams of analysts (the participants) to analyze the security and risk issues of HCN when applied to the context of Cityville.

The participants, during the Training day, received from the customer two chapters of the HCN book (Ch.1 and Ch.6). Moreover the participants received a 1-hour seminar about HCN, which was given by one member of the organizing team. We have divided participants in groups and assigned them to a security engineering method. Each group was formed by three or four participants: three professionals and one master student. Once divided in groups, the participants were required to attend the tutorials given by method designers about the method they would have used for the analysis of the application scenario. Each tutorial had a duration of approximately 2,5 hours. The *Application* phase lasted from the 14th to the 25th of May: the members of the groups worked remotely using collaborative tools (mainly BSCW and Marratech). During this phase, the groups received additional material about the application scenario: the material was a 2-page long note from the CEO informing the participants (the CEO's fictional consultants) about additional requirements from their client, the Health Care Authority. The remote Application phase has been followed by a two day face-to-face Application phase which took place in Paris. The first day of the face-to-face Application phase was organized into five group work sessions. Each session had duration of 75 minutes. On the second day, groups were asked to give a brief presentation of their work to an audience including organizers, method designers and other students. Participants were also asked to provide their feedbacks via questionnaires and focus group discussion conducted by the observers. Feedbacks were related to the assigned security engineering methods and the organization of the study. At the end of the Application phase, each group has to deliver a 10-page final report that was evaluated by the method designers.

In the 2012 study we have made some changes such as more time for the training phase, and the explicit participation of two industry representatives as customers.

4 Recruiting Participants and Method Designers

We have invited a number of research groups to join the activity (travel partly at our expenses). The selection of the security requirements methods to be evaluated was driven by three main factors: the number of citations, the fact that research on the method is still ongoing, and availability of the method designers. Out of the various oral and email invitations, only four groups accepted to participate in 2011. The most frequent justification has been lack of human resources ("PhD student terminated his studies"), followed by "no longer active in the field". The four security methods that have been the object of study for the comparative evaluation in 2011 are Coras, Secure Tropos, Problem Frames and SI*. Coras is a model-driven method for risk analysis proposed by SINTEF [12]. Problem Frames [6] is a framework for security requirements elicitation and analysis developed at Open University. Secure Tropos [16] is a methodology designed at Univerity of East London; the methodology supports capturing,

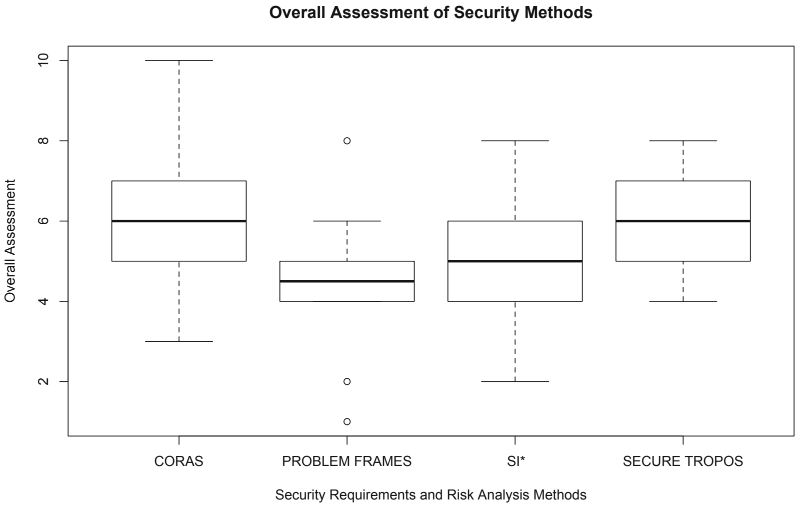

Fig. 2. Overall Security Methods Assessment (scale 1-10)

analysis and reasoning of security requirements from the early stages of the development process. SI* [4] is a formal framework developed at the University of Trento for modeling and analyzing security requirements of an organization. Forty-nine participants were involved in 2011: thirty-six participants were professionals with a minimum of five years of working experience in the field of Auditing in Information Systems. The professionals were attending the Master Course in Audit for Information System in Enterprises at Dauphine University. Thirteen participants were master students in Computer Science from the University of Trento with a background in Security Engineering and Information Systems. We have decided to have both junior and senior participants because involving only students in empirical research is known to be a major threat to external validity [19]. Therefore, by involving professionals, we wanted to avoid this issue.

5 Data Collection and Analysis

In this section we report some of the preliminary findings deriving from the analysis of the data collected by means of the questionnaires, the focus groups and the post-it notes fill out by the participant. Then, we compare the the coverage of security requirements derived from the analysis of groups' final reports with the feedbacks given by the participants about methods coverage.

5.1 Rating Tasks and Data Distribution

Participants were asked to give a final vote to the methods on a scale from 1 to 10 representing the overall level of appreciation of the methods. Figure 2 shows the distribution of the participants's responses to the overall assessment of each method. There is no statistically significant difference among the methods: the median evaluation is barely sufficient (solid line for each box). Each method also had both supporters and detractors (as one can seen from the whiskers), with the exception of the Problem Frames method that had a more concentrated distribution around the median.

A more refined analysis of the responses of the participants on the conceptual model, the analysis capabilities, and the tool support for each of the methods, provide a clearer separation among the methods. The flattened distribution obscures these important details: each method has strenghts and limitations which tend to balance each other.

For the conceptual model and the analysis, participants specified the level of appreciation in the scale 0 (Dislike), 1 (Like it the least), 2 (Like it a little), 3 (Like), 4 (Fairly Like), and 5 (Like it the most). Figure 3 shows that the least liked conceptual model belongs to the Problem Frames method while the one more appreciated by the participants is SI*'s conceptual model.

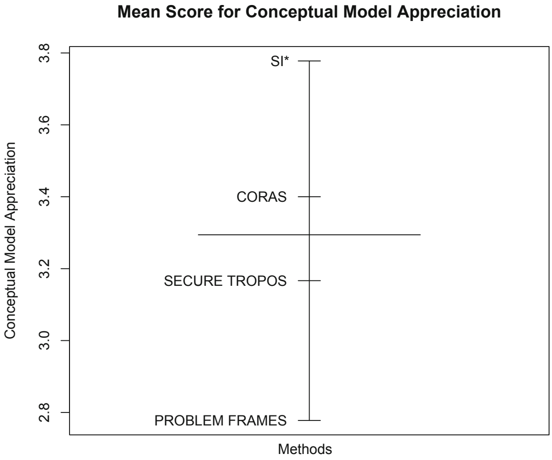

Fig. 3. Conceptual Model Appreciation

The participants helped to identify the key features of the SI* conceptual model: "*Considers dependencies between actors (social aspects). Effective to clarify responsibility of all the actors. Takes into account trust relations*", " *Study of relations between goals and agents*", and "*Easy to show the permission level*". Regarding Coras, the participants have not reported negative or positive aspects. On the contrary, the participants have spotted several weaknesses of Secure Tropos and Problem Frames's conceptual models. For the participants, Secure Tropos conceptual model "*does not have any mechanism to analyze alternative solutions to achieve a goal or to enforce treatments*", "*Difference between goals and objectives is not clearly defined.*", "*Ambiguity in assigning constraints between depender and dependee.*", and "*Hard goals Vs soft goals are ambiguous*".

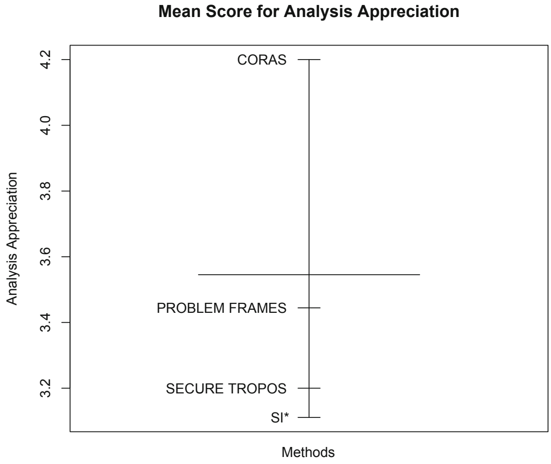

Fig. 4. Analysis Appreciation

Instead, the participants have appreciated less Problem Frames's conceptual because: "*Difficult/ confusing to understand few terms like warrants, formal arguments*", "*Resource nodes cannot be distinguished*", and that "*Unfortunately you don't see the actor connected to the task in the process. Based on my experience, it is very useful to have such information".*

Figure 4 shows that for Secure Tropos and SI* the security analysis is a critical aspect that impacts on the overall assessment of the methods.

Secure Tropos analysis' limitations are: "*It does not have any mechanism to analyze alternative solutions to achieve a goal or to enforce treatments*", "*Ambiguity in assigning constraints between depender and dependee*", "*No automatic analysis process*, "*Lack manual for guidance*". Instead, the weak aspects of SI*'s analysis are: "*Time consuming. Hard to make the links between the model and the risks maybe due to lack of time*", " *Risk analysis not really obvious. Method complicates the risk analysis with too many details. Does not cover all kinds of risks. Difficulty to find all the risks - thanks to the chart*"; "*Trust relations should support high level of detail*"; "*Starts with the actors and not the goal analysis. Should support goal prioritizations*"; " *Focuses only on some elements in the case, for example: no place for physical asset* ", and "*Too precise - going down to too many levels of details*". Coras and Problem Frames analysis have been appreciated by the participants because they provide a detailed step-by-step process. Participants said that Coras is "*Step by step procedure. Very detailed step with specific explanation*" while Problem Frames has" *Clear and organized steps to analyze problem domain. Structured.*"

About the tool, participants have been asked to evaluate the usability choosing a value in the scale from 0 (Unusable), 1 (Not easy at), 2 (Extra effort needed), 3 (Easy), 4 (Fairly Easy), and 5 (Very Easy). As shown in Figure 5, usability is a factor that could have influenced the overall assessment of the methods since the majority of the participants have negatively evaluated the usability of the tools.

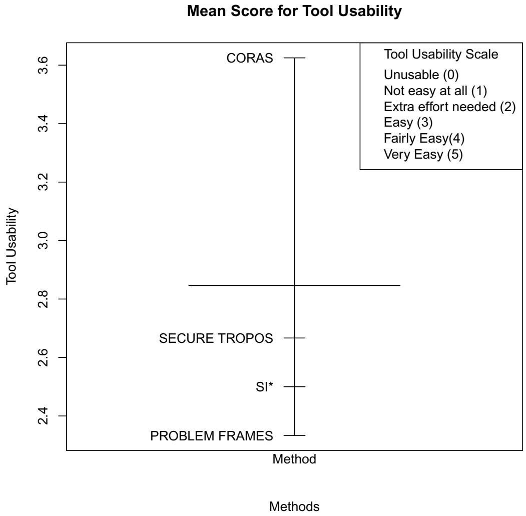

Fig. 5. Tool Usability Assessment

The concerns given by the participants during focus groups discussions and on post-it notes allowed us to understand which are the features of the tools that determine their poor usability. participants have reported several bugs of the Secure Tropos tool. To mention some:" *Problem in saving projects when the size of diagram increases*", " *Unreliable system - Impossible to open the project, once it is closed*", " *Cannot select multiple components to manipulate. For example: to delete a goal, you have to also delete the dependencies*", " *Its an incomplete system which does not have many features like select, cut, copy, duplicate.. etc*", and " *Difficult to modelise/draw a diagram with a complex scenario.* SI*'s tool has received mixed comments: "*Good tool and useful to modelize - Clear presentation of the diagram. Simple graphic diagrams. Friendly graphic formalism. Easy to draw diagrams*", and "*The tool is too abstract, it is possible that the analysts drop some part and don't go into detail.* Also the comments given by the participants on Problem Frames tool are mixed: some of them reported that is a" *nice tool for modeling basically the security issues*", while others mention missing features:"*in other tools we can have decompositions like 'OR' but in this, I can't find it*". The participants have expressed their concerns also on Coras tool, which has been rated the more usable tool among the four methods: they said that the tool " *not enough help to troubleshoot software problem*", " *difficult to understand diagrammatic representation.*", " *no reasoning services in the tool (automation computing of likelihood) support*", and that the " *tool is too complex with only fixed determined models (threat, treatment, asset diagrams)*".

5.2 Data on Methods' Application

During focus groups discussions, participants were asked to evaluate the coverage of the security requirements elicited because of the application of the methods.

participants who applied Coras mentioned that it provides"*..powerful and good coverage in case right people are involved"*. Other participants also said that"*the coverage may depend on the way you apply the method than on the method itself*", and that "*the concept of the asset we found before, if you are the right guy you will get good coverage otherwise it is very difficult"*.

Problems Frames provides good coverage according to participants's feedbacks: one participants said "*I think the method provides wide coverage on different aspects .... And also we can identify what are the assets that we need to protect, and why we need to protect it, not exactly how but what and why we need to answer. So it provides a good coverage to understand the problem"*; another Partcipant said that the"*Method is good in providing coverage but is not good at providing all kinds of treatments"*.

participants did not provide specific feedbacks on the coverage of Secure Tropos. Instead, on SI* participants asserted that leads to the identification of general security requirements: "*we found a set of general security requirements that every organization would have. Maybe they are 70% relevant but not sure*" and that "*our requirements are not ambiguous but probably they are too global, too large, and so we don't define precise recommendations to have a complete view*".

In order to see if the participants' feedbacks on security requirements coverage were well founded, we have also analyzed the final report delivered from the groups: for each method, we have retrieved the security requirements and the security recommendations identified by the group as result of the method application.

Figure 6 shows that each method lead to the identification of different requirements. The groups working with Coras and Secure Tropos have focused on Availability, Confidentiality, and Integrity. The groups who have applied Problem Frames instead have identified Integrity and Confidentiality as main security requirements. Finally, the groups working with SI* have focused mostly on Integrity and Privacy. Figure 7 shows that Access Control and Training are the most frequent proposed security solutions across all the methods. These results are quite obvious since Access Control and Training are two of the most common security solutions and they can be applied to any system.

However, what we noticed is that the security solutions identified were quite generic and sometimes not linked to the security requirements at all. This is true in particular for Secure Tropos and SI*. In the final recommendations deriving by the application of Secure Tropos, Confidentiality and Integrity have been recognized as important requirements but there no security recommendations on how they have to be preserved. The same is true for SI*, where Integrity is one of the important security requirements but no security solutions on how to preserve it have been found in the reports. Instead, for the groups working on Coras, it seems that the security solutions proposed in the final recommendations have been easily derived from the treatments identified during the risk analysis.

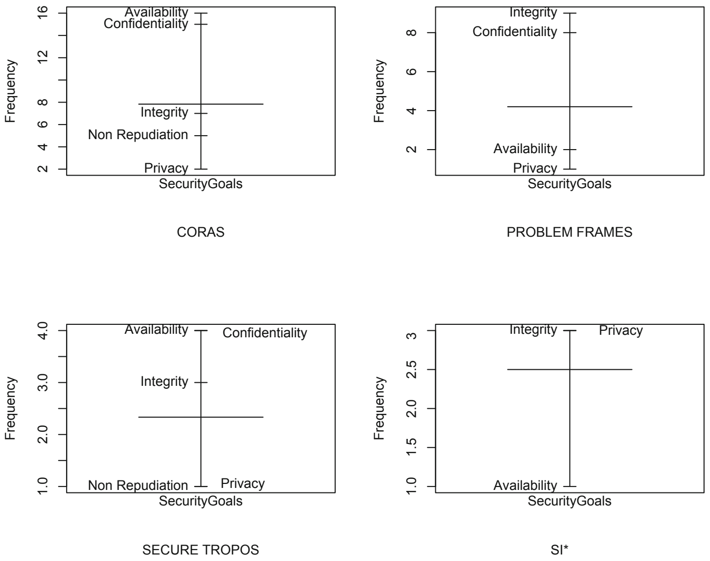

Fig. 6. Security goals frequency for each method

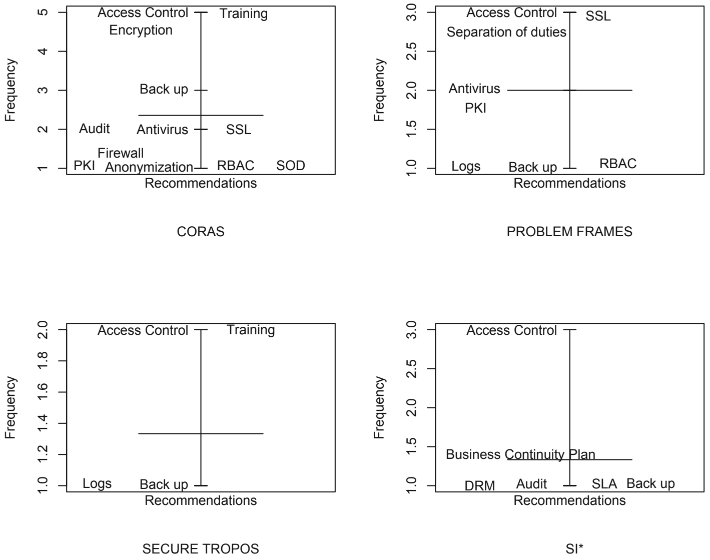

Fig. 7. Security recommendations frequency for each method

6 Related Works

In a mapping study on empirical evaluation conducted in 2009 [3], only the 13% of the papers reporting research in Requirements Engineering were based on a case study.

For instance, [2] is an example of an observational study, while [18] is a good illustration of controlled experiments in security. Two usability studies have been performed to assess how easily models used in risk analysis can be understood [5,8]. Yskout et al.[21] have proposed a methodology to preserve trust properties in a software design. They validated the methodology via an empirical study involving 12 subjects. The results show that their approach does provide an edge in terms of reduced effort required to evolve a software architecture.Heyman et al. [7] have performed an assessment of the quality of about 200 security patterns by means of panel judgment. The main findings are that, often, the same pattern is re-published under a different name, and that the average quality of the documentation for security patterns is low. The closest work to ours is the one by Opdahl et al. [18]. They have carried out two controlled experiments (with 28 and 35 participants, respectively) to compare two methods for early elicitation of security requirements, namely attack trees and misuse cases. They assessed the effectiveness and coverage of these methods. The main differences between the study we present in this paper and the experiments conducted by Opdahl et al. are: a) our study involved not only master students but also professionals to limit threats to external validity; b) the groups of participants have applied only one method among the one under evaluation and not all the methods as in the study by Opdahl et al.; c) our study was a qualitative study, and thus to collect data we have not only used questionnaires but we also interviewed the participants. Other initiatives have focused on gaining knowledge and understanding the secure implementation of software products with empirical means. These usually include tapping the "wisdom of the masses" by surveying development groups to determine what the typical practices are that one should follow to come up with secure software. The most comprehensive such survey as of today is the Building Security In Maturity Model (BISSIM3) [13] which observed and analysed real-world data from 42 leading software development companies, 9 of which were Europeans, the rest from the US. However the level of granularity is too coarse.

Compared to these proposals, the comparative empirical study we describe in this paper does not aim to evaluate a single security method or to survey the best practice in building secure software. Our initiative aims at comparing different methods when they are used by subjects different than inventor.

7 Threats to Validity

- **Construct Validity.** Threats to construct validity are related to decide to which extent what was to be measured was actually measured. The main threat to construct validity in our case studies regards the design of the research instruments: our main measurement instruments are interviews and questionnaires. Three researchers have checked that only questions of relevance to the research questions were included in the interview guide and in the questionnaires; therefore we believe that our research instruments measure what we want to measure. Moreover, to reduce this threat we have gathered data using other data sources like audio-video files, post-it notes, and participants' reports on methods'application.

- **Internal Validity.** A threat to internal validity is that the time spent in training participants was not enough for them to apply the method and understand the application scenarios. To mitigate this threat, method designers and customers should be available to answer questions that participants may arise during the application of the methods. Another threat is represented by participants' previous knowledge of other methods. For example, in one group it was decided to use COBIT to identify the security requirements for the application scenario, rather than the method assigned to the group. In this case, the feedbacks provided by the group have no value because the method was not applied.

  One additional threat is that the time participants spent on the methods'application was too short to let them provide insigthful feedbacks on methods' effectiveness. We are aware that the opinions of the participants may have been different if they would have applied the method over a longer period of time.
- **External Validity.** We have evaluated the effectiveness of the security requirements and risk analysis methods with both master students and professional from different countries, and we have applied the methods in different contexts - Smart Grid and Healthcare. This give use some confidence that our hypothesis and conclusions on methods' effectiveness have a medium degree of generalizability.
- **Conclusion Validity.** An important threat to the conclusion validity of our studies is that our sample is relatively small in statistical terms. In fact, for each method the sample consists of twelve participants. In order to increase the statistical significance of the emerging hypothesis and insights on methods' effectiveness, we are conducting another case study to have a bigger data sample.

8 Lessons Learned and Conclusions

The first study we have conducted allowed us to collect a set of insightful comments about the Coras, Secure Tropos, Problem Frames and SI* and to identify their strenghts and limitations.

Coras overall has been the most appreciated method because it provides a step-by-step process and the conceptual model comprises concepts that are easy to understand. However, the usability of the tool needs to be improved with automatic reasoning support for example automatic likelihood and consequence computation or treatments selection. An important aspect that came out is that it leads to a complete identification of security risks only when experts are involved in the risk analysis.

Secure Tropos requires improvements in all its aspects: the definition of some of the concepts in the conceptual model needs to be revised to avoid ambiguity on when to use a concept or another; the steps of the process to elicit functional and security requirements needs to be clearly defined; the tool requires to improve the functionalities to create, delete and copy diagrams. SI* has to improve the process to elicit requirements under two aspects: the steps of the process are not clear and need to be detailed and the risk analysis has to be simplified because it is overcomplicated and it does allow to identify all the possible risks. Also the SI* tool needs to improve the visual notion of the diagrams. Last and least, Problem Frames strenghts are the process because it is very detailed and it leads to identify which assets need to be protected, and the tool because it allows to model basic security issues. However, the conceptual model for security argumentation needs to be better defined.

The study also helped us to understand a number of aspects to improve the evaluation protocol for the next edition of the challenge, which is currently taking place in May and June 2012. We list the main lesson learned below.

- **Don't (try to) collect too much data.** When we initially designed the protocol, we followed the design principles recommended in most statistics and action research texbooks and tried to collect everything that could be collected (audio, video, photos of artifacts, computer diagrams, questionnaires for each possible phases of the experiment etc.) in order to eliminate all possible confounding factors. We found out this was a mistake. At first the sampling disturbs the natural flow of the study. Second, participants developed an uneasy feeling of being stalked or got simply tired of filling questionnaires. This study requires participants to be intellectually motivated, concentrated and challenged[1].
- **Take time to explain orally everything.** Another title could be "don't assume people read the consent information sheet". This is particularly important for audio and video recording. Does the participant understand whether they are being monitored and for which purposes? If they have not read the information sheet carefully, after some time they might feel monitored and withdraw from the study.
- **Write simple scenarios and ask a customer to join.** Participants will always ask unexpected questions about the scenario. Following the experience of our first small-scale trials, we have tried to offset in advance the questions raised by participants by choosing a 100+ pages with details and a long description of business and high-level security requirements by the CEO. It turned out to be a mistake. Most people didn't read them carefully. The description of the application scenario should be kept short and provide only key information. Rather a customer should be present during the Application phase to answer participants' questions.

[1] Humans or mice do not usually withdraw from medical or biology experiments because they find it boring or feel stalked (they either die or their therapy is somehow modified).

- **Define collectively the rules of engagement:** Do observers and method designers know which questions by the participants they can answer during the Application phase? Since method designers represent themselves and are not under the control of the experimenters they might answer the question of the participants by doing the particular fragment of the model for them. It is therefore important that this issue is discussed and understood by everybody.
- **Beware temptations to use background knowledge.** In one group the participants decided that COBIT would have yielded to better result and silently switched to it. It should be made clear that the evaluation is not about the method results itself but on the application of the method.

In summary we think that this was a very challenging and interesting study. It has been the first time that more than 40+ CS students and practitioner consultants tried to apply security requirements engineering methods in a *comparative* settings. The results of the analysis are still preliminary but this could be the first step towards the development of a scientific protocol for the empirical evaluation of security requirements engineering methods.

Acknowledgments. This work is partly supported by the projects EU-FP7-IST-NoE-NESSOS, EU-EIT-ICTLabs, EU-FP7-SEC-SECONOMICS, PAT-TRISE.

References

1. Healthcare Collaborative Network Solution Planning and Implementation. Vervante (2006)
2. Asnar, Y., Giorgini, P., Massacci, F., Saidane, A., Bonato, R., Meduri, V., Ricucci, V.: Secure and dependable patterns in organizations: An empirical approach. In: Proc. of RE 2007, pp. 287–292 (2007)
3. Condori-Fernandez, N., Daneva, M., Sikkel, K., Wieringa, R., Dieste, O., Pastor, O.: A systematic mapping study on empirical evaluation of software requirements specifications techniques. In: Proc. of ESEM 2009, pp. 502–505 (2009)
4. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling security requirements through ownership, permission and delegation. In: Proc. of RE 2005, pp. 167–176 (2005)
5. Grondahl, I.H., Lund, M.S., Stolen, K.: Reducing the effort to comprehend risk models: Text labels are often preferred over graphical means. Risk Analysis 31(11), 1813–1831 (2011)
6. Haley, C., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering 34(1), 133–153 (2008)
7. Heyman, T., Yskout, K., Scandariato, R., Joosen, W.: An analysis of the security patterns landscape. In: Proc. of the 3rd Int. Workshop on Soft. Eng. for Secure Systems, SESS 2007, p. 3. IEEE Computer Society (2007)
8. Hogganvik, I., Stølen, K.: A Graphical Approach to Risk Identification, Motivated by Empirical Investigations. In: Wang, J., Whittle, J., Harel, D., Reggio, G. (eds.) MoDELS 2006. LNCS, vol. 4199, pp. 574–588. Springer, Heidelberg (2006)

9. ITGI. CoBIT - Framework Control Objectives Management Guidelines Maturity Models, 4.1 ed. The IT Governance Institute (2007)
10. Jürjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
11. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg (2002)
12. Lund, M.S., Solhaug, B., Stolen, K.: A guided tour of the coras method. In: Model-Driven Risk Analysis, pp. 23–43. Springer (2011)
13. McGraw, G., Chess, B., Migues, S.: Building Security In Maturity Model (BSIMM3), 3rd edn. Cigital Inc. (2011)
14. Mead, N.R., Stehney, T.: Security quality requirements engineering (square) methodology. SIGSOFT Softw. Eng. Notes 30(4), 1–7 (2005)
15. Microsoft Security Development Life Cycle. Microsft sdl website (2011), `http://www.microsoft.com/security/sdl/default.aspx`
16. Mouratidis, H., Giorgini, P., Manson, G.: Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 1031–1031. Springer, Heidelberg (2003)
17. NIST Comp. Security Division. Recommended security controls for federal information systems and organizations. Tech. Rep. 800-53, U.S. Nat. Inst. of Standards and Technology, Rev. 3 (2009)
18. Opdahl, A.L., Sindre, G.: Experimental comparison of attack trees and misuse cases for security threat identification. Inf. Softw. Technol. 51(5), 916–932 (2009)
19. Potts, C.: Software-engineering research revisited. IEEE Softw. 10(5), 19–28 (1993)
20. The Open Web Application Security Project. Owasp website (2011), `http://www.owasp.org`
21. Yskout, K., Scandariato, R., Joosen, W.: Change patterns: Co-evolving requirements and architecture. Soft. and Sys. Modeling J. (2012)