Formality of the Security Specification Process: Benefits Beyond Requirements

Jose Romero-Mariona
University of California, Irvine
Donald Bren School of Information and Computer Sciences
jromerom@uci.edu

Hadar Ziv
University of California, Irvine
Donald Bren School of Information and Computer Sciences
ziv@ics.uci.edu

Debra J. Richardson
University of California, Irvine
Donald Bren School of Information and Computer Sciences
djr@ics.uci.edu

Abstract
An important difference among approaches to software requirements specification is the degree of formality of the specification process itself. In this paper we explore 12 approaches to security requirements specification. We divide the 12 approaches into two distinct groups, those that follow a formal specification process and those that follow an informal one. We evaluate and compare the benefits that each group of approaches offers in six key areas, including resulting system's security, scalability, security requirements integration, constraint consideration, testing benefits, and integration of other requirements. Our analysis shows that security requirements specified using a formal process are highly correlated with benefits that go beyond the requirements stage, compared to requirements specified using an informal process.

1. Introduction

This paper presents results from our ongoing research in the area of security requirements [1]. As part of this research, we evaluated and compared two types of approaches for specifying security requirements, namely those that follow a formal versus informal requirements specification process. We identified six areas of interest for comparing the two alternatives; analysis of the results provides insights as to how those six areas are affected by following a formal versus an informal specification process. Formal specification processes are known to be superior with respect to “discovering specification errors and presenting the system specification in an unambiguous way” [2]. We present six areas of security requirements found to be highly correlated with following a more formal process; we do not, however, claim causality, i.e., that formality would cause or otherwise guarantee success in those areas.

This paper makes several observations and contributions:

• Conducting a detailed and comparative survey of 12 approaches to security requirements, including, for each approach, specific inquiries about the formality of its specification process

• Though a literature survey is inherently subjective, our inquiries are designed to be as objective as possible and amenable to quantitative comparison

• Our analysis indicates that formal processes rank higher along six dimensions of comparison, thus confirming the benefits of formal processes for security requirements specification

• Our analysis confirms additional benefits beyond requirements, i.e., into the design, implementation and testing of the original security requirements.

The rest of this paper is organized as follows; section two provides background as well as limitations of our work. Section three explains relationship found between the six areas identified as highly correlated with the specification process’ formality. Section four summarizes the paper by presenting conclusions and lessons learned.

2. Background

The ideas presented in this paper stem from ongoing research in the area of security requirements engineering (SRE) [3]. We consider software security to have as its primary goals: the preservation of confidentiality, integrity, and availability of the information assets and resources that the software creates, stores, processes, or transmits including the executing programs themselves [4]. We consider a software process to be a coherent and structured set of activities for specifying, designing, implementing and testing software systems. A software specification process is the process of establishing what services are required and the constraints on the system’s operation and development [2].

Our larger research effort included surveying two different types of formality; namely the formality of the process for specifying security requirements and the level of formality of the security requirements themselves. In this paper we report only on the first type, namely formality of the process. We explore two kinds of formality of the process for specifying security requirements, formal and informal processes. A formal process tends to be defined, structured, and repeatable so that it produces specifications that are unambiguous, more complete, more precise, and so on [5]. Some of the drawbacks of a formal
specification process include limited scope of method and limited scalability [2]. An informal process is traditionally unstructured and produces specifications that are prone to more ambiguity and incompleteness. The advantage of an informal specification process is that it tends to be faster and easier than a formal process [6].

Originally, we surveyed 34 approaches that are used either directly or indirectly for the specification of security requirements. Of these, 12 were chosen based on their maturity, relevance to both industry and academia, and the amount of information and documentation. We consider these 12 approaches to be major representatives of current options in SRE. The surveyed approaches are:

- USER Method [17]       - CLASP [18]

2.1 Survey Method

Our framework for evaluation and comparison of each surveyed approach consists of 30 questions that were asked of each approach and answered by the authors of this paper. The questions were formed based on aspects found in our literature review that point to important characteristics security requirements and their approaches should have. Our framework has been carefully constructed to compare the approaches along aspects of interest and relevance for SRE. In addition to a text answer, each method is rated using a “star rating” ranging from 0-4 stars, based on a Likert scale [19]. The scale rates the amount of support that each approach provides for a specific question. The questions were constructed such that the star rating per approach can be determined as objectively as possible. The criteria for the star rating is determined as follows,

-“None”/Ø There is no information regarding any aspects/concepts about the question at hand in any of the sources describing the approach.

-“Low”/★ Aspects/concepts related to the question at hand are suggested (but not in detail) OR there is enough information to suggest that any support would be possible by the approach.

-“Moderate”/★★ Aspects/concepts related to the question at hand are explicitly mentioned in some of the sources describing the approach, but it is not a critical component. Support is described, but no specific measures to operationalize that support are given.

-“Moderate-High”/★★★ Aspects/concepts related to the question at hand are explicitly mentioned in a majority of the sources describing the approach; they are important aspects to the approach. Support is described and some measures to operationalize that support are described.

-“High”/★★★★ Aspects/concepts related to the question at hand are critical to the approach. Support for the specific question is described across the majority of sources describing the approach. There is evident and extensive support for the question at hand, and measures for achieving this support are described.

For example, question 15 from our survey explores the level of support provided by each approach for considering alternative/additional security requirements that could have been missed during elicitation. A rating of “None” suggests that there is no support. A rating of “Low” suggests that alternative/additional security requirements could be considered, but would require significant effort. A rating of “Moderate” is granted if the approach explicitly considers alternative/additional security requirements as part of the process. A “Moderate-high” rating indicates that the approach has a specific step in its process for specifying security requirements where various stakeholders consider alternative/additional security requirements. A “High” rating provides in addition to the support by the “Moderate-high” rating, tools for making this consideration more productive.

2.2 Survey Limitations

It is important to discuss some possible limitations associated with our survey method. The first limitation refers to the approaches chosen in the survey. As mentioned above, we started with over 30 approaches, but reduced this number to focus on 12 approaches that we consider to be most relevant and most qualified representatives of the field.

The survey is also limited by the amount of information available regarding each approach. As mentioned above, one of the factors in selecting which approaches to survey was the information available for each approach. Another limitation is, inevitably, the authors’ limited understanding and ability to dissect and analyze each approach, given the information available. Another possible limitation is the choice of questions to be included in our survey evaluation framework. We followed a structured process for constructing each of the 30 questions; we believe that the questions asked in our survey provide comprehensive coverage of the field of SRE.

Lastly, there is inherent subjectivity involved in the assignment of each star rating for each approach. We constructed each question as objectively as possible, so that the star rating for each one could also be determined as objectively as possible.

3. Benefits Beyond Requirements

The literature survey offers several interesting observations about as well as characteristics of the 12 approaches. One such characteristic is the effect that the formality of the process for specifying security requirements has on other aspects of SRE.

3.1 Formal vs. Informal Specification Process

We consider two different levels of process formality for specification, i.e., informal and formal. Rather than using a definition of formality, we looked for certain criteria in each
approach. The following criteria were used when determining the level of formality of each process,

Informal (I)- There is a lack of structured steps necessary to develop the specifications. It is/would be difficult to repeat the process. There are no specific roles defined for the stakeholders involved in the specification process.

Formal (F)- There are standard steps that must be followed in order to develop security requirements specifications; in addition there exists a defined order in which the steps should be followed. Developers with a variety of experience with the standard steps can successfully repeat the process of specifying security requirements. Roles are defined explicitly for each of the steps needed for the specification.

Based on these descriptions, we identify 7 approaches that follow a formal specification process and 5 approaches that follow an informal process. Thus we have,

- Misuse Cases (I) - Abuser Stories (I)
- Secure TROPOS (F) - Security Problem Frames (I)
- Anti-Models (F) - i* Security Requirements (F)
- Common Criteria (F) - SQUARE (I)
- OCTAVE (F) - Attack Trees (I)
- USER Method (F) - CLASP (F)

Though this categorization can be argued as biased, we followed the same process for each approach trying to be as impartial as possible. Formality of the process was not one of the original aspects considered in the survey, but it became evident that exploring this aspect could help explain some of the survey results.

3.2 Effects of the Formality Process

Once the 12 approaches were grouped based on the level of process formality to specifications, we started looking at the 30 areas in our survey for possible correlations that this grouping could have. We found that there are six major areas where the results obtained by both groups are reasonably different. These six areas were identified after the end of the larger survey effort; thus, the survey was not geared or biased in any way toward the observations reported here. Furthermore, these six areas correspond to characteristics that are useful, necessary, or present beyond the requirements stage of software development. We anticipate that the support available in these six areas could facilitate the development of the system.

Table 1 shows the six areas along with the star rating for each approach.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Resulting System's Security</th>
<th>Scalability</th>
<th>Security Requirements Integration</th>
<th>Constraint Consideration</th>
<th>Testing Benefits</th>
<th>Other Requirements Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Misuse Cases</td>
<td>★★</td>
<td>★</td>
<td>☐</td>
<td>☐</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td>Abuser Stories</td>
<td>★</td>
<td>★</td>
<td>☐</td>
<td>★★</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td>Secure TROPOS</td>
<td>★★</td>
<td>★★★★</td>
<td>★</td>
<td>★★</td>
<td>★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>Sec. Prob. Frames</td>
<td>★★</td>
<td>★★★</td>
<td>☐</td>
<td>☐</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Anti-Models</td>
<td>★★</td>
<td>★★★★</td>
<td>☐</td>
<td>☐</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>i*</td>
<td>★★★★</td>
<td>★★</td>
<td>★</td>
<td>★</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td>Common Criteria</td>
<td>★★★★</td>
<td>★★★★</td>
<td>★</td>
<td>★</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td>SQUARE</td>
<td>★★★</td>
<td>★★</td>
<td>☐</td>
<td>A★</td>
<td>★</td>
<td>☐</td>
</tr>
<tr>
<td>OCTAVE</td>
<td>★★</td>
<td>★★★</td>
<td>☐</td>
<td>☐</td>
<td>★</td>
<td>☐</td>
</tr>
<tr>
<td>Attack Trees</td>
<td>★</td>
<td>★</td>
<td>☐</td>
<td>D★</td>
<td>★</td>
<td>☐</td>
</tr>
<tr>
<td>USER</td>
<td>★★</td>
<td>★</td>
<td>★</td>
<td>D★</td>
<td>★</td>
<td>☐</td>
</tr>
<tr>
<td>CLASP</td>
<td>★★★★</td>
<td>★★★</td>
<td>★</td>
<td>A★</td>
<td>★★</td>
<td>★</td>
</tr>
</tbody>
</table>
3.2.1 Resulting System’s Security. The first area observed is the level of system security resulting from the specifications. While we cannot accurately assess the level of security of the final system, this area is simply a prediction about how the level of security of the specified requirements will have on “how secure” the resulting system will be. In making this prediction, a variety of aspects were taken into consideration, including, but not limited to, the types of security mechanisms that the approach helps specify, the involvement by security experts during specification, and support offered by the approach itself to check for security flaws. Figure 1 shows the predicted system security level based on the specifications created by each of the 12 surveyed approaches. In Figures 1-6 the Y-axis refers to the star rating that each approach on the X-axis received.

![Figure 1. Resulting system’s security due to the specifications](Image)

To our surprise, the majority of the approaches surveyed were highly scalable. Half of all the approaches surveyed were rated highly scalable; a quarter received a score of moderate, and the last quarter received a score of low. When we look at the approaches that rated high, five out of those six approaches follow a formal process to security requirements specification. We can then see that one of the potential benefits of a formal specification process is high scalability.

3.2.2 Scalability. This area refers to the ability of both the approach as well as the specifications to serve for a variety of project sizes and constraints. For the approach part, we looked for specific support that each approach provides for adjusting to projects of different sizes. For the specifications part, we considered important that the specifications produced by the approach need to be easily modifiable in the case that the size of the project changes. Figure 2 shows the star distribution for each of the 12 approaches when it comes to their scalability.

![Figure 2. Scalability of surveyed approaches](Image)

This is one of the surveyed areas with the least amount of support. We are concerned about the fact that the highest level of support in this area is only a moderate star rating (by both Secure TROPOS and CLASP). Furthermore, we can see that the only approaches that explicitly offer any support for integrating security requirements at later stages of development are those approaches that also follow a formal process to
security requirements specification. In fact, five out of the seven approaches that follow a formal process offer support. We suggest, based on these results, that security requirements specified following a formal process could be more easily integrated later on.

3.2.4 Constraints Consideration. Our survey explored constraints helpful during later stages of development, considered by the specifications created. These constraints vary from Architectural (A), Design (D), Implementation (I), and Maintenance (M). Each approach is rated in the level of support it provides for each of these types of constraints; the support for each type also ranges from 0-4, therefore the total amount of stars possible for each approach in this area is 16 stars. We refer to constraints, as factors that the security requirements specifications account for which can benefit other stages of development. Figure 4 below shows the total combined support of all types of constraints for each surveyed approach.

![Figure 4. Constraints consideration of specified security requirements](image)

In this area, both groups of approaches provide very low support. This might be due to the fact that specifications are not often created with future constraints in mind. What we can see from Figure 4 though is that specifications created following a formal process provide more constraint consideration than those created using an informal process.

3.2.5 Testing Benefits. The benefits that the specifications provide for testing, is an important areas of our survey. For this area we looked for specific instances where approaches either explicitly consider testing decisions as part of their specification process, or the specifications themselves record testing-related information. The level of testing benefits in the specifications created by each of the 12 approaches is shown in Figure 5.

![Figure 5. Testing benefits of specified security requirements](image)

We are glad to report that the majority of approaches offered a good amount of support for testing benefits. This is an area where we believe that specifications created following an informal process have a slight lead over formal process ones. Nonetheless, it is refreshing to see that almost all of the approaches in each group provide testing benefits.

3.2.6 Integration of Other Requirements. The last area refers to the support available for integrating requirements other than security ones at later stages of development. We consider this an important area because it evaluates the support that each approach provides not only for security, but also for other important aspects of development. In this area we looked for explicit support for other types of requirements (either functional or non-functional) including, but not limited to, usability, performance, and reliability. Figure 6 shows the level of support provided by each approach for this area.

![Figure 6. Other requirements integration](image)

Support for integration of other types of requirements is another area with not much support by either group. While most of the approaches that follow a formal specification process provide some support for integrating other types of requirements at later stages of development, the support is very limited. In the group that follows an informal process, only one approach (Misuse Cases) provides any support (low) for helping integrate other types of requirements. We therefore consider it an additional benefit associated with following a formal specification process over an informal one.
3.3 Support Summary

We argue that the formality of the security specification process is related to and highly correlated with other areas of development. Our evaluation and analysis described in this paper suggest that a formal process to specifying security requirements is preferred, as it provides the specification with added benefits for later stages of development. We cannot, however, claim causality, i.e., that these benefits are a direct result of a formal specification process.

For five out of the six areas surveyed, our study indicates there are moderate levels of support for all five by both types of approaches; only security requirements using a formal process provide support for one area, namely integrating them at later stages of development. We suspect this can be attributed to the kinds of artifacts created — when following a formal process - as well as their predetermined syntax. Both the kinds of artifacts and syntax could facilitate their integration later on. Somewhat different from all others, testing benefits was the only area where approaches following an informal process performed slightly better than those following a formal process. This is interesting and we suspect could be due to the fact that most informal approaches utilize some form of story-telling [20] to model security requirements. Storytelling is likely to convey information that is lost in a more formal process.

4. Conclusions and Lessons Learned

Security has become a system property that is not only sought after more than before, but also truly needed. In this paper we explored 12 approaches to SRE. We divided these approaches into two groups based on the level of formality they employ in their requirements process. We analyzed the level of support that these two groups provided in six different areas that were determined based on a survey previously conducted. The six areas explored were resulting system’s security, scalability, security requirements integration, constraint consideration, testing benefits, and integration of other types of requirements. The commonality among these six areas is that they represent characteristics important at later stages of development. Examining the level of support offered by each of the two groups across the six areas revealed that approaches that follow a formal process to security requirements specification offered more support in most areas than those following an informal specification process. The research also showed that when it comes to the integration of security requirements either at later stages or with other types of requirements, there is a definite lack of support across the board. We believe that results presented in this paper help reaffirm that while formality might require more resources than informality, it offers distinct benefits when used during security requirements specification. These benefits do not end at the requirements level, but rather prove vital in later stages of development as well. Further, results from our evaluation and analysis could be used to determine or select an approach to security requirements best-suited for a specific project [21].

5. References