A Preliminary Study: Challenges in Capturing Security Requirements and Consistency Checking by Requirement Engineers

Massila Kamalrudin[1], Nuridawati Mustafa[2], Safiah Sidek[1]
[1]*Innovative Software System and Service Group (IS[3]), Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.*
[2]*Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.*
massila @utem.edu.my

***Abstract*— There has been a growing concern on the importance of security with the rise of phenomena, such as e-commerce and nomadic and geographically distributed work. Realizing the security early, especially in the requirement analysis phase, is important so that security problems can be tackled early enough before going further in the development process and avoid re-work. Ensuring the consistency of elicited functional security requirement of requirements specification is also crucial as the requirements should be well understood and agreed upon by all the stakeholders and end-users. Therefore, the aim of this paper is to further discuss on the challenges faced by Requirement Engineers (REs) in: (1) capturing Security Requirement and (2) Consistency Checking in Requirement Engineering. Motivated from the need to ensure consistency in functional security requirement for developing secure software and the gaps found in the existing works, a survey has been conducted involving 38 experts in software engineering in the industry. The survey aims to identify the current problems faced by them during the elicitation process, security standards used as the reference, elicitation and validation method, and the important properties considered while developing secure software. Results of the survey show that REs face difficulties to understand the security needs and the existing standards are difficult to understand. Therefore, it is proposed that an automated tool to elicit security requirements should be developed.**

***Index Terms*— Consistency Management; Secure Software; Security Requirements; Security Requirements Validation.**

I. INTRODUCTION

The success of secure software development depends on quality security requirements. However, the process of eliciting security requirements is tedious and complex. It also requires REs to have security experience when eliciting consistent security requirements from the clients-stakeholders. Most of the REs also face problems in eliciting consistent security compliance requirements from the clients-stakeholders as they tend to misunderstand the real needs and the use of security terms. ‘‘If you don’t know what you want, it’s hard to do it right.’’ Unless the clients-stakeholders know what to secure, against whom, and to what extent, it is obviously very difficult to construct a secure system or to make a substantial statement about its security [1]. All of these issues may contribute to eliciting inconsistent security requirements.

Inconsistency in elicitation may lead to the development of incorrect and insecure software systems as well as the disruptions of schedule and the increase in a project's expenditure. In relation to this, finding errors at the later stage of system development are costly because correcting such errors may need changes in the whole specification and implementation. There are efforts to solve security leakages during the system development process for the purpose of increasing quality, but this may lead to extra cost and wastage of time to organizations [2]. In this case, it is vital to address the security issues as early as possible in software development

Whenever inconsistencies are discovered during the runtime of a system, it may not be possible to fix the problems since the causes of the problem cannot be identified or located, and a common practice to address this issue is to redevelop the system from the beginning. In this case, it is crucial to ensure the consistency of functional security requirements during the early stage of requirements specification to avoid a waste of time and high cost. However, as stated in [3], one of the challenges is to ensure that the set of identified security requirements is consistent and complete so that no necessary security requirements are left undiscovered, and that the set of security requirements jointly indeed enforces the security needs.

This paper is organized as follows. In Section 2, we discussed a survey of the literature by outlining the challenges faced by REs in capturing security requirements. Together with that, we discuss the existing consistency management works in handling inconsistencies in security requirements. Next, in Section 3, we explained the results from the survey conducted among the experts to see the current problems faced by them during the elicitation process, the elicitation and validation method used and also the important properties being considered in developing secure software in industries. Then, in Section 4, we discuss the thread of validity of this study. Finally, this paper ends with a conclusion in Section 5 that proposes an agenda for future work.

II. LITERATURE SURVEY

Several works related to capturing and consistency checking in security requirements have been discovered and these works are presented in this section.

Sindre et al. in [4], [5], [6] and [7] elicited the security requirements based on use cases, with the emphasis on description and method guidelines. The approach is an extension of the traditional use cases. The authors claimed

two new concepts, which are the misuse cases and misusers, along with suitable relationships using a diagram notation, templates for textual descriptions, and method guidelines. However, the method guidelines are still too general and imprecise as the number of potentially critical assets and associated threats that must be considered are large. Further, the misuse case approach itself is not equally suitable for all kinds of threats, specifically because the misuse does not always involve or exploit neither an identifiable sequence of actions nor an identifiable misuser.

As being explained by Lin et al. in [8] and [9], they derived the security requirements based on problem frames. It was implemented using Jackson's Problem Frames by analyzing security problems in order to determine security threats and vulnerabilities.

According to Houmb et al. [3], they proposed a security requirements engineering methodology called SecReq, which is an extension of security requirements engineering by seamlessly integrating elicitation, traceability and analysis activities. This methodology combines three techniques: the Common Criteria (CC), the heuristic requirements editor HeRA, and the UMLsec. The integrated SecReq method supports early detection of security-related issues (HeRA). Their systematic refinement is guided by the CC, and it has the ability to trace security requirements into UML design models. A feedback loop helps reusing the experience within SecReq and turns the approach into an iterative process for the secure system life-cycle. It is also in the presence of system evolution. However, it has several limitations: The consistency of the elicited security requirements during Step 1 is not being considered; and there is still no guarantee that these requirements will be correct and consistently represented in the solution design and then the implementation.

Similar to the previous work, El-Hadary and El-Kassas [10] also proposed a methodology for security requirement elicitation based on problem frames, which is to assist developers to elicit adequate security requirements during the requirement engineering process with the aid of previous security knowledge. This methodology adopted a security catalog based on the problem frames. It was constructed to help identify security requirements with the aid of previous security knowledge. Abuse frames were used to model threats, while security problem frames were used to model security requirements. They claimed their methodology can extract more complete security requirements compared to other relevant methodologies. However, the results are still immature since the comparison was made with two security requirement elicitation methodologies only. Perhaps, the consistency level has not been proven in their paper since more empirical studies on large-scale software systems are needed in order to evaluate the methodology.

Kamalrudin et al. [11] has introduced an automated tool support called MaramaAIC using semi-formal models: EUCs and EUI for managing business requirements consistency and validation. This tool provides an end-to-end rapid prototyping approach together with a patterns library that helps to capture requirements and check the consistency of requirements expressed in textual natural language requirements and then extracted to semi-formal abstract interactions, essential use cases (EUCs) and user interface prototype models. However, this tool does not consider the consistency for functional security requirement.

Security requirements engineering process with a generic system model core has been proposed as in [12]. Decke explains the system model core and demonstrates its extensibility using the example of vehicular systems. They explained two methods for formal inspection of the system model, which are how security engineer can be assisted by consistency checking of the system model, and how to verify the sum of generated security requirements to ascertain the correctness of the security concept. Even though the consistency checking is included in this model, the implementation is still tedious because no automated tool is provided. The implementation suggestion requires the REs to choose the checking on their own, depending on the type of the implementation of the methodology. However, their recommendation to use lambda functions in C++ 11 or Java 8 does not provide a guarantee that the result of consistency checking is achieved.

In summary, there are a number of works done in checking the consistency of requirements. However, only a few were found in security requirements, especially the functional security requirements. In addition, the existing consistency management approaches are still immature and have tedious implementation.

III. Preliminary Study: Software Practitioners

Based on the challenges found in the previous section, we have conducted a quantitative survey with few software industries such as IBM Malaysia, Cyber Security Malaysia and other software companies in the area of Lembah Klang, Malaysia. This survey was participated with 38 software experts with various position inclusive of software developers, system engineer and software tester. Further, this surveys was conducted through paper-based and online-based method, where all participants are and treated anonymously.

The aims of this survey are threefold: First, to analyze the current problems faced by them during the elicitation process; second, to identify the security standards used as reference, elicitation and validation method; and third, to identify the important properties being considered while developing secure software.

The followings were carried out during the survey: Firstly, nine questions were designed to address the aims of the survey. Next, the questionnaires were distributed to software engineering experts and a total of 38 software engineering experts from established companies took part in this survey. The results were analyzed using ATLAS.ti to identify the percentages of similar variables (responses). In order to do this, the variables were prepared based on the possible answers that correspond to the respective questions. To avoid biasness, all of the variables were validated by an expert.

The subsequent part of this section presents the findings of this survey. It begins with the background of the sample. The background of the sample was characterized by work experience. As shown in Figure 1, 68% of the respondents had less than 5 years work experiences as Software practitioners, while 32% of the respondents had more than 5 years work experience as software practitioners. This indicates that more than three quarter of the respondents having at least 2 years as software practitioners.

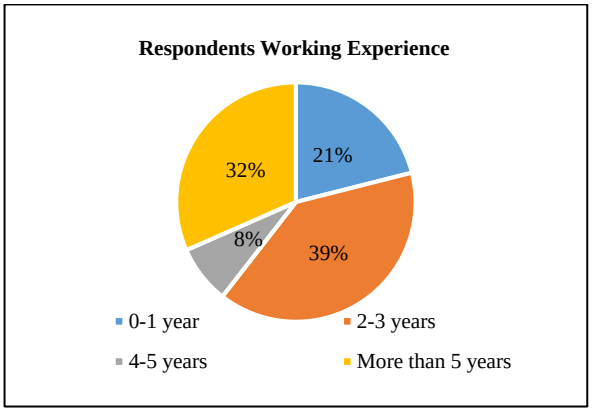

Figure 1: Respondents Working Experiences Distribution

Subsequently, as shown in Figure 2, most of the software practitioners involved in this survey are the software developer with 32% and follow by 18% of them are system engineer and 13% are software tester. This indicates that all of the respondents have experience working with software requirements.

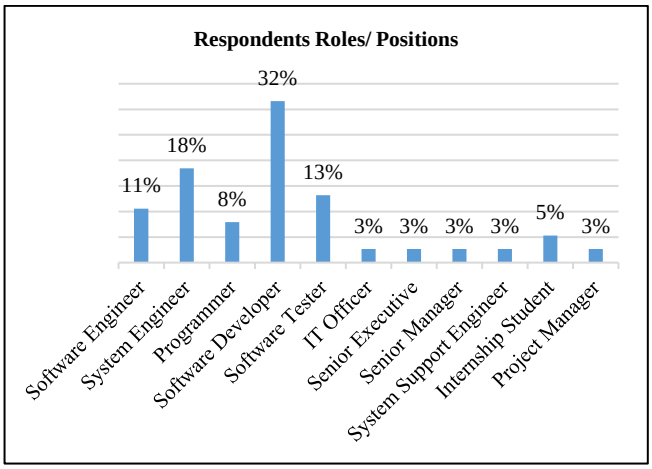

Figure 2: Respondents Roles and Positions

The respondents were also requested to indicate the problems they faced working as Requirements engineers. As shown in Figure 3, most of the RE's reported that their clients do not clearly know and understands the security needed by their systems. The second highest problems was the difficulty of RE's to understand the security terms in security standards documents. This is followed by the difficulty to write the security requirements. A possible reason why they are having this difficulty is that none of the template/best practice template is easily used and available to be used as template.

Furthermore, the majority of the respondents, which is 79% of the respondents, do not refer to any security requirement template when preparing the documents, as shown in Figure 4. This is also proven in Figure 3 as most of the respondents have difficulty in understanding the security terms in the existing templates. Only a small number of them refer to the companies' template and other existing standards, such as FIPS, SSL, NIST and US Security Layer/Standards. This clearly shows that the existing standards is difficult to understand and not user-friendly.

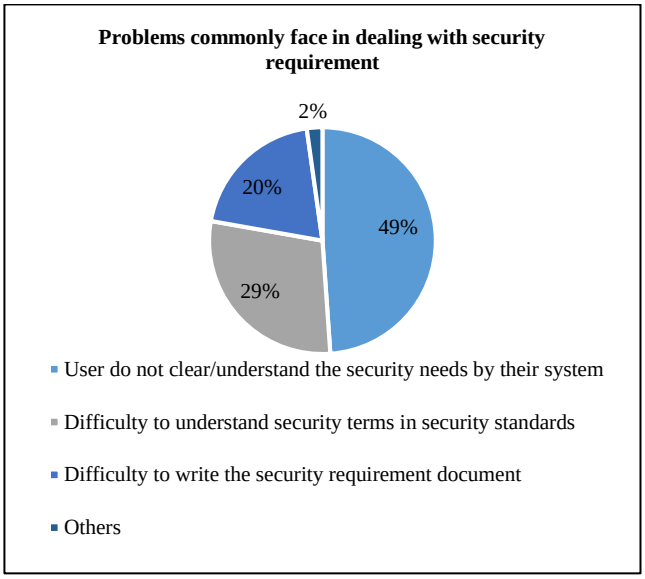

Figure 3: Problem in Security Requirements

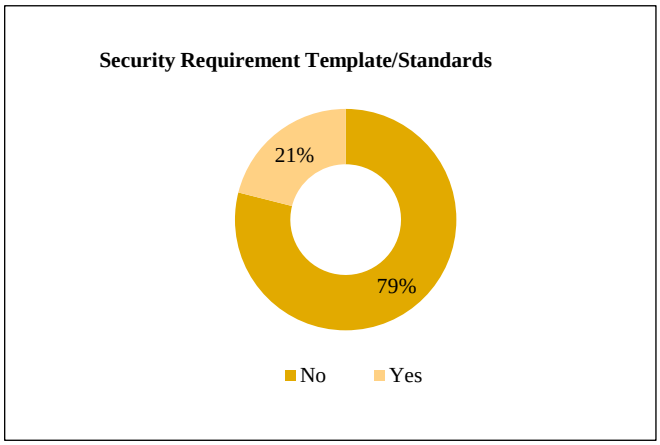

Figure 4: Security Requirement Template/Standards

In terms of security requirement consideration phase, as shown in Figure 5, the majority (66% of the respondents) mentioned that they considered the security requirements during Requirement Analysis phase. While others, (34% of the respondents) only considered the security requirement during the design, implementation and testing phase. It can be concluded that although the security requirements have been considered at the early stage, it is not taken into consideration seriously or fully explored.

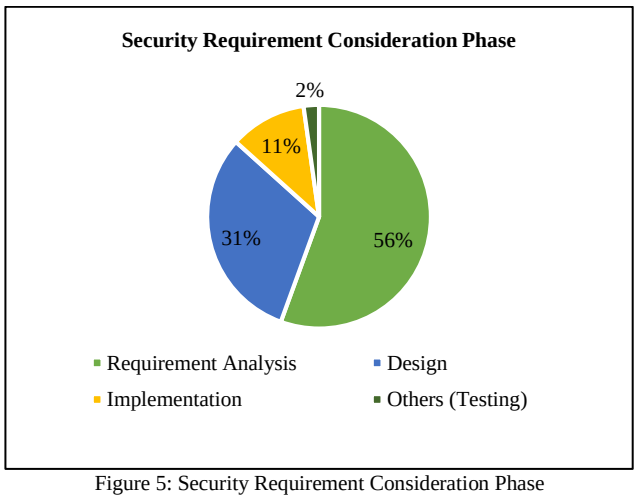

Figure 5: Security Requirement Consideration Phase

According to the respondents, there are a few methods that have been used to elicit security requirements. Figure 6 shows that 25% of the IT practitioners which is not even half of them collect the security requirements based on the feedback of the users and stakeholders. 20% of them still elicit the security requirement after the product has been developed, that is during product testing, which is already late in the software development cycle. Security is often thought as an after development issue and it can only proven to be very harmful once the software has been developed and exist in the market. It is difficult to remove any kind of virus or vulnerabilities that might have been introduced in the software during its development process. However, it would be highly beneficial if the security problems are understood in the early phases of SDLC process, especially during the requirement and analysis phases so that the software developed incorporates the security issues [13]. Further, others define the security requirements based on experience, discussion, analysis on situation, security analysis, depending on data sensitivity and SOP/ Network Security sharing.

Validation of requirements usually takes place after the elicitation of security requirements. Figure 7 illustrates the methods used to validate the elicited security requirements. As shown in Figure 7, 49% of the respondents used a tool to validate the requirements, while 27% of them used a model and 24% of the respondents validate the requirements manually. This implies that almost half of the respondents used a tool for validation. In this respect, tool is the most popular validation method used in validating security requirements. This result also indicates that there are quite a number of Software Engineering practitioners depend on the manual methods in doing the validation process, in which the existing validation tools are not their preference.

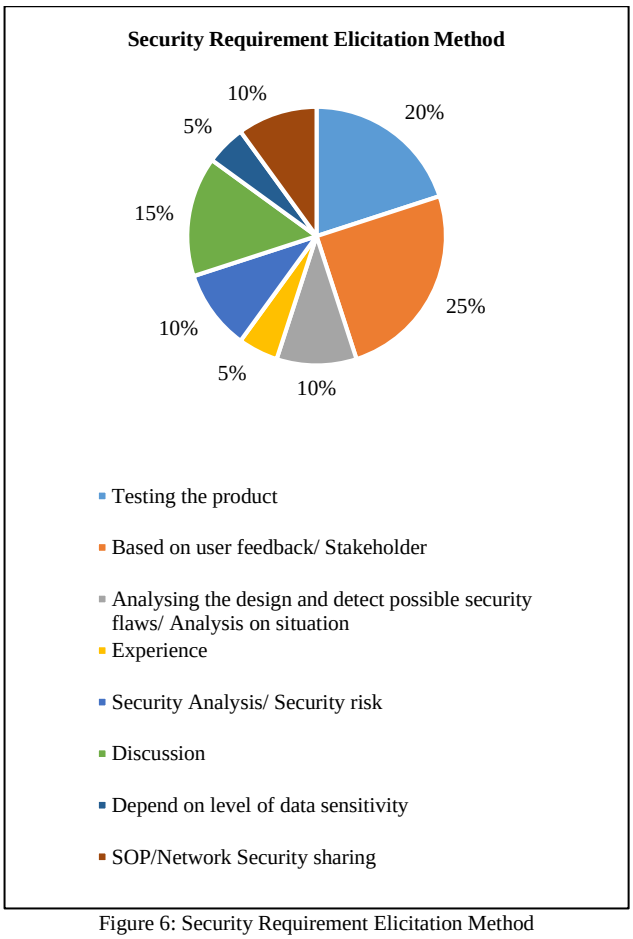

Figure 6: Security Requirement Elicitation Method

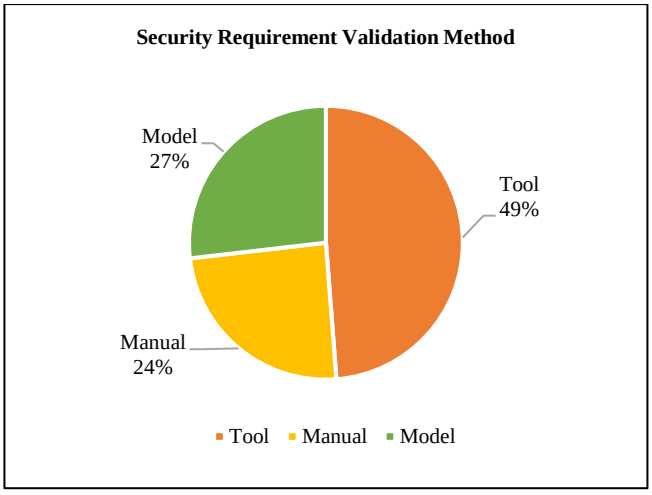

Figure 7: Security Requirement Validation Method

While conducting the security requirement validation, the two main properties that are highly considered are the correctness and consistency with 32% of the respondents as shown in Figure 8. These validation are not easily done as mentioned in [14]. One of the critical tasks of requirements engineers in this process is to ensure that requirements specification at each step remains correct, or at least that errors are found as early as possible. While others 20% of the respondents and 16% of them respectively are the completeness and ambiguity. The overall results are shown in Figure 8.

Considering that the security standards documents are difficult to understand and the template provided is not user friendly, it can be concluded that RE's have difficulties to write security requirements.

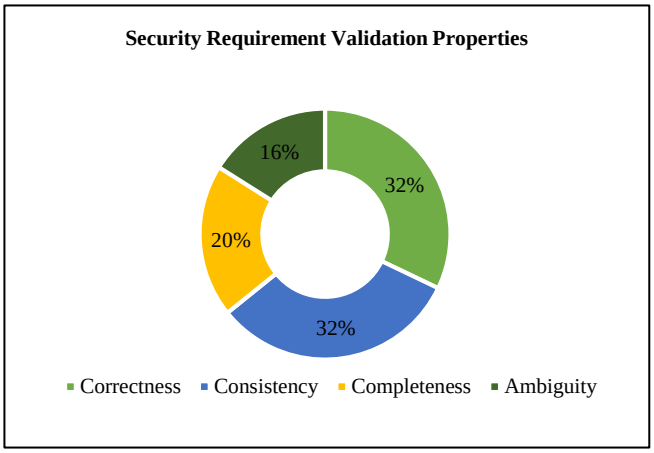

Figure 8: Security Requirement Properties

IV. LIMITATION

In summary, there is few limitations that we need to overcome in the future study. Firstly is the poll of respondents are the software practitioners from the medium to large size company only and mostly located at Klang Valley. Moreover, in this study, we do not focus on the small company because we believed they might not use specific tool and method due to cost and size of the projects. Both constraints are believed could affect the results as different demographic and size of company and project could contribute to different findings of the survey. Therefore, we plan for a replication of this study in the future to strengthen the results.

V. Conclusion and Future Works

Based on our literature survey and the online survey, we conclude that there is no automated tool and template that cater for elicitation of functional security requirements. Although there are few solutions has been proposed, the implementation is still tedious because no automated tool is provided. Furthermore, the result of the proposed model in terms of consistency checking is still immature. A more effective approach for security requirement engineering is needed to provide a more systematic way for eliciting adequate security requirements. We believed, it is timely to have an automated tool on eliciting security requirements. This is based on the results of the survey that found that the IT practitioners face difficulties in understanding the security terms when they elicit requirement requirements. As for future work, we are motivated to develop or propose a new best-practice template for guidance in writing consistent functional security requirement with consistency management checking. We strongly believe that this approach will improve the quality of elicited security requirement for secure software development.

VI. Acknowledgment

I would like to thank UTeM and MoE for the funding research: FRGS/1/2015/ICT01/FTMK/02/ F00291.

References

[1] B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt, "A comparison of security requirements engineering methods," Requir. Eng., vol. 15, no. 1, pp. 7–40, Mar. 2010.

[2] [2] T. R. Farkhani and M. R. Razzazi, "Examination and Classification of Security Requirements of Software Systems," in 2006 2nd International Conference on Information & Communication Technologies, 2006, vol. 2, pp. 2–7.

[3] [3] S. H. Houmb, S. Islam, E. Knauss, J. Jürjens, and K. Schneider, "Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec," Requir. Eng., vol. 15, no. 1, pp. 63–93, Mar. 2010.

[4] [4] G. Sindre and A. L. Opdahl, "Templates for Misuse Case Description," 7th Int. Work. Requir. Eng. Found. Softw. Qual. REFSQ 2001, vol. 6, pp. 125–136, 2001.

[5] [5] G. Sindre and A. Opdahl, "Capturing security requirements through misuse cases," NIK 2001, Nor. Inform. 2001, http// …, 2001.

[6] [6] G. Sindre, D. G. Firesmith, and A. L. Opdahl, "A Reuse Based Approach to Determining Security Requirements," 9th Int. Work. Requir. Eng. Found. Softw. Qual. REFSQ 2003, vol. 8, pp. 127–136, 2003.

[7] [7] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," Requir. Eng., vol. 10, no. 1, pp. 34–44, Jan. 2005.

[8] [8] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett, "Introducing abuse frames for analysing security requirements," in Proceedings of the IEEE International Conference on Requirements Engineering, 2003, vol. 2003–Janua, pp. 371–372.

[9] [9] L. Lin, B. a Nuseibeh, D. C. Ince, M. a Jackson, J. D. Moffett, and F. O. M. a. C. D. of Computing, "Analysing Security Threats and Vulnerabilities Using Abuse Frames," 2003.

[10] [10] H. El-Hadary and S. El-Kassas, "Capturing security requirements for software systems," J. Adv. Res., vol. 5, no. 4, pp. 463–472, Jul. 2014.

[11] [11] M. Kamalrudin, J. Hosking, and J. Grundy, "MaramaAIC: tool support for consistency management and validation of requirements," Autom. Softw. Eng., Feb. 2016.

[12] [12] H. Decke, "Checking and Verifying Security Requirements With the Security Engineering System Model Core," no. c, pp. 26–35, 2015.

[13] [13] R. Jindal, R. Malhotra, and A. Jain, "Automated classification of security requirements," in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 2027–2033.

[14] [14] D. Zowghi and V. Gervasi, "The Three Cs of Requirements: Consistency, Completeness, and Correctness," Int. Work. Requir. Eng. Found. Softw. Qual. Essen, Ger. Essener Inform. Beitiage, 2002.