Agile Practitioners' Understanding of Security Requirements: Insights from a Grounded Theory Analysis

Evenynke Terpstra, Maya Daneva
School of Computer Science
University of Twente
Enschede, The Netherlands
m.daneva@utwente.nl

Chong Wang
State Key Lab of Software Engineering, Computer School
Wuhan University
Wuhan, China
cwang@whu.edu.cn

***Abstract*—A 2017 systematic review on engineering non-functional requirements in agile projects revealed a number of published proposals for approaching security requirements in agile settings. While these proposals acknowledge the urgent need for methods to systematically engineer security requirements in agile projects, they were designed mostly in academic settings. Very little empirical evaluation of these proposals happened in real-life contexts. In turn, little is known about how agile developers understand security requirements and how they devise their coping strategies regarding these requirements. This position paper presents a qualitative analysis that sought to discover how agile practitioners reason about security requirements, what contextual factors they consider important for shaping the process of coping with security requirements in agile projects, and what these strategies are. We conclude with some implications for practice and research.**

***Index Terms*—Security requirements engineering, Agile project management, agile project development, qualitative study, empirical research method.**

I. Introduction

Agile project management (APM) and agile software development (ASD) approaches have provided organizations with the mindset, the processes and the tools to proactively deal with requirements changes and incorporate new requirements dynamically and continuously in every stage of the development lifecycle [1]. Despite the convincing empirical evidence indicating that agile methods work for the engineering of functional requirements, there are growing concerns about these methods' lack of respect for non-functional requirements (such as security, safety, usability). In turn, the researchers and practitioners in ASD, APM and in security requirements engineering (RE) came up with a variety of ideas for enhancing agile methods by embedding security practices [2].

This paper focuses on the interplay of agile project delivery and one type of non-functional requirements – namely, security. A 2017 review [2] on engineering non-functional requirements in agile projects revealed ten published proposals for approaching security requirements in agile. However, most of these proposals were designed in academic settings without any practitioner's involvement. Plus, very little empirical evaluation research has been done in real-life contexts regarding these proposals. How practitioners in the field think about security requirements and how they devise their processes of coping with the issues these requirements pose, is hardly known.

This position paper presents a qualitative study that sought to discover (1) those concepts that agile practitioners use when reasoning about security requirements issues and searching for coping strategies in their projects, and (2) the concepts describing the solution strategies themselves. For this purpose, our study attempts to answer two research questions: (1) *What contextual factors in agile projects do practitioners perceive as challenging and take into account when searching for coping strategies regarding security requirements?* And (2) *What coping strategies do they use?* We carried out a grounded theory (GT) analysis that used as input practitioners' postings on a prominent social media site, LinkedIn. The postings are in the professional LinkedIn group 'Agile and Lean Development' and include practitioners' comments and evaluations on aspects of security RE in agile projects. Because the data is in text format – and hence, qualitative in nature, its analysis easily lends itself to the coding and data comparison techniques of the GT method [3]. The rest of our position paper describes related work and presents our research process, our results and our discussion concerning implications for practice and research.

II. Related Work

Beznosov and Kruchten examined how the security assurance practices fit or do not fit in agile methods [4]. These authors found that approximately half of the conventional assurance methods mismatched the principles and practices of ASD. To relieve this problem from the perspective of agile methods, Gustav et al. extended one of the popular agile methods - eXtreme Programming (XP), by specifying additional steps in the XP Planning Game to engineer security requirements [5]. The steps include the identification of security sensitive assets, the formulation of Abuser stories, and the definitions of security-related user stories and coding standards. Moreover, Howard et al. proposed an iterative security architecture within an agile project to summarize which components contribute to security and consider security in each iteration [6]. From the perspective of roles involved in agile projects to cope with security requirements, Baca et al. proposed four new roles to every agile project team to deal with security issues [7]. These roles in-

978-1-5386-3488-2/17 $31.00 © 2017 IEEE
DOI 10.1109/REW.2017.54

clude security manager, security architect, security master and penetration tester [7]. Considering the characteristics of security requirements, Vähä-Sipilä states that not each of them is worth protecting [8]. For example, it depends on how high the costs for protecting would be and what it would cost to mop things up. Vähä-Sipilä explored how to merge security requirements to agile principles in individual cases. However, seldom research tried to get response from practitioners in this field and give a comprehensive understanding on challenges of implementing security requirements in agile projects and the solutions that worked.

III. Research Process

Our research process draws on the methodological work of Verner et al. [9] and Hookway [10] suggesting the use of publically available data for the purpose of exploratory qualitative research. Following these authors, our data collection includes the use of posts in a professional discussion group on LinkedIn. As Verner et al. and Hookway indicate, this data collection strategy fits well in situations when a researcher would like to balance the cost for executing the study against breadth and depth of the study and when publically available qualitative data is easily available for analysis. In particular, we focused on this LinkedIn group, because of its professional ethics. Plus, the second author has been a member of this group for six years and observed the group's conversations to be consistently professional. The group is the world's largest and engages members in in-depth discussions on Agile, Lean, eXtreme Programming, Scrum, Kanban, Organizational Transformation, Product Discovery, Agile Adoption, Continuous Delivery, Continuous Integration, and Code Quality. We expected that the analysis of the information on this discussion group would provide a deep understanding about what's in the practice of agile security RE, from the perspective of those in the field. The LinkedIn group's conversations are data in textual form, allowing for analyzing qualitative data immediately without the resource intensiveness of voice recording and transcription [10].

For our research purpose of, we took two online conversations dated April'15 and June'16, respectively. We chose these two because they match our research goal (i.e. they were on the topic of coping with security RE in agile). The first was initiated by a California-based agile trainer and the second – by an Australia-based agile consultant. The first included the shared opinions and evaluations of 19 practitioners that generated 42 posts. These practitioners were in roles of Scrum Consultants, Senior Agile Consultant, Technical Project Manager, Agile Systems Engineer, Software Engineers, working for organizations in America, Europe and Australia. The second conversation included evaluations of seven practitioners that contributed 19 posts. These practitioners were in roles such as Chief Technology Officer of an agile company, Agile Developers, Scrum Trainers and Agile Consultants, and worked in America, Australia, and the United Kingdom. We analyzed the posts' information manually by using GT coding techniques, which yielded the themes presented in Sect. IV. For interested readers, our qualitative data extracted from LinkedIn, is available at this site: https://www.evenynke.nl/security-requirements/.

IV. Results

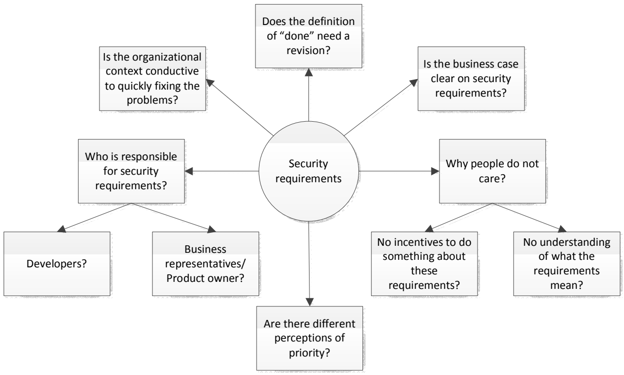
A. Understanding on the challenges

Table I presents the concepts that we found in the posts, indicating the problematic aspects of security RE in agile context, according to our participants.

TABLE I. Concepts Indicating Problems

ID	Concepts
C1	Security is hard to "sell" as a business value to the business
C2	Security requirements cost money to elaborate due to experts' involvement.
C3	Agile techniques are business-value-driven.
C4	People drop security because they perceive it a fight not worth fighting.
C5	Different people prioritize security differently.
C6	Security gets a low priority from the customers' perspective (in the context of mobile app development).
C7	People do care about security, but do not think about it.
C8	Security requirements are often poorly defined and owned.
C9	Security requirements get often delivered in the last minute.
C10	Non-functional requirement stakeholders are absent during planning sessions
C11	Developers do not know or do not care about security issues.
C12	Development teams have no security training.
C13	The product owner has often too much power and instills his attitude of treating non-functional requirements.
C14	The product owner has often too little knowledge on security requirements.
C15	The product owner is sometimes acting like a business owner or stakeholder and pushes only for features.
C16	Agile is often not implemented in a sub-optimal way.
C17	Agile is relying on people's tacit knowledge.
C18	Agile techniques are vulnerable for forgetting things like security.
C19	Most agile processes miss a feedback loop regarding non-functional requirements.
C20	Agile techniques depend on the knowledge of a few developers.
C21	Organizational structure can make or break modifications related to security requirements.

Based on our iterations of the GT analysis practices, we attempted to aggregate these concepts into a conceptual model. Specifically, we worked to arrive at a model that describes the kinds of questions practitioners were asking themselves when trying to comprehend security requirements challenges in their agile projects. This exercise yielded the model presented in Fig 1. We make the note that (1) as this study is exploratory in nature, the purpose of this model is descriptive only. It is to explicate the agile practitioners' understanding and reasoning of the challenges surrounding security requirements in their projects; and (2) the model takes the perspective of the technical agile team – not the client. This model is to help those professionals in an agile project team, who are concerned with security, to 'zoom-in' into the contextual settings of their projects and see those concepts which are important to consider when devising a suitable coping strategy for security requirements. It describes what happened in all those experiences of the participants that are shared in the LinkedIn group.



Does the definition of "done" need a revision?
Is the organizational context conductive to quickly fixing the problems?
Is the business case clear on security requirements?
Who is responsible for security requirements?
Security requirements
Why people do not care?
Developers?
Business representatives/ Product owner?
No incentives to do something about these requirements?
No understanding of what the requirements mean?
Are there different perceptions of priority?

Figure 1. Understanding the security requirements challenges from the agile practitioners' perspective.

Our analysis suggests that practitioners consider the following conceptual categories: (1) ownership of security requirements, (2) definition of "done", (3) business case, (4) attitude towards security requirements, (5) organizational setup, (6) perceptions of priority. In our conceptual model, we formulated questions about these concepts that practitioners treat, when developing an understanding of security requirements in their agile settings. Below we explain each of these conceptual categories and their impact on the crafting of a coping strategy for security requirements. We add in brackets, some examples of concepts from Table I that support these conceptual categories. (We make the note that Table I includes 21 concepts, however because of space limitation, we use only 9 as examples).

1.Who owns the security requirements? In the participants' experience, business representatives and product owners usually have little awareness of security requirements and rarely work towards their elaboration early on (e.g. C14, C15). On the other side, developers who understand risks associated with poorly treated security requirements, may not know how to communicate the possible security issues to their product owner and convincingly present him with information on how much it would cost if not fixed and if a problem arises.

2.Does the organization's culture help or hurt the engineering of security requirements? The group's practitioners found that organizations differ in their way of educating agile developers (e.g. C21). If an organization provides security courses to agile developers, this would be conductive to sort out security requirements issues timely and more systematically.

3.Does the definition of "done" (DoD) need a revision? In agile projects, the DoD is a list of criteria to be met by an artefact increment – be it a feature, a user story, a sprint or a release. Two participants shared the view that the DoD should represent the requirements generated by an organization's need to implement security measures and their priorities (e.g. C6). In their perception, the DoD is to ensure that a team delivers functionality that meets not only the requirements of the product owner and the business users, but also the requirements of the organization. If security is part of the DoD, then this is conductive to maintaining the conversation over security requirements.

4.Is the business case clear? Security requirements may or may not be part of the project's business case in an organization. One practitioner shared that those security requirements unaccounted for in the business case, were perceived as a costly component to add to a later project stage (e.g. C1, C2). Developers, in turn, were reluctant to add them (e.g. C11).

5.What is the attitude of the team members? The participants elaborated that there are cases when team members *"do not care"* about security requirements just because there is no incentive to do so (e.g. C1, C4). Or, because no one really understands completely what these requirements are (e.g. C11).

6.What are the perceptions of priority at inter-iteration time? In the experiences of the participants, often the business representatives drive the priorities; plus, their perceptions of the priorities of security requirements differ from those of the developers (e.g. C5, C6, C15). In the words of one participant, *"the PO may need to understand the correlation between security issues and business damage before priorities change"*.

B. Understanding on the coping strategies

Table II presents those concepts that we found in the posts, indicating the solutions that the practitioners experienced.

TABLE II. CONCEPTS INDICATING COPING STRATEGIES

ID.	Concepts
S1	Integrate security features in the definition of 'done'.
S2	Integrate security features in the estimates.
S3	Make security part of the acceptance criteria.
S4	Make technical stories from security features.
S5	Bake security features in the user stories.
S6	Make functional requirements from security requirements.
S7	Use a security regulation to justify the sec. requirements.
S8	Prioritize the security risks that are worth protecting.
S9	Add an expert to the development team.
S10	Educate the business about security risks.
S11	Raise awareness in the development team for security.
S12	Make sure the product owner is supporting security.
S13	Cross functional streams help not forgetting about security.
S14	Automate security checks.
S15	Review the code on security.

We found that the coping strategies of the participants could be divided in the following three groups: ***(1) Solutions addressing the artefacts dealing with security requirements***. Concepts S1, S2, S3, and S5 show the various possible ways to embed security requirements into existing agile artefacts, be it the DoD (S1), the estimates (S2), the acceptance criteria (S3) and the user story (S5). Concept S4 suggests the introduction of a new artefact called 'technical stories' to complement the user stories in an agile project. ***(2) Solutions addressing the human factors in agile projects.*** Concepts S9 to S13 suggest coping strategies focusing on the people involved in agile projects, such as e.g. obtain the product owner's commitment and support to security requirements (S12) and include a security expert (S9). ***(3) Solutions addressing the agile process itself.*** Concepts S6, S7, S8, S14 and S15 suggest introducing security-focused practices into agile, e.g. prioritizing security risks, reviewing security code, or automating security checks.

V. Discussion on Implications

This paper has some implications. For practitioners, our findings suggest that those working with security requirements in agile projects, should be proactive. Specific actions that seem to be particularly sensible are: reserving a budget for security issues, educating all the agile team members (both product owners and developers), on security and letting them document the security requirements (as the risks due to undocumented requirements may be too high). Next, we found that agile practitioners can choose among a broad range of coping strategies and that some of those may be more expensive than others, e.g. engaging a security expert would be more expensive than embedding security requirements into estimates. We think that it is important for practitioners to be aware of these choices and their organizational implications in terms of cost involved, scope of the change they instill, and the fit with organizational culture.

Furthermore, there are two research implications. The conceptual model in Fig. 1 could serve as a starting point for designing follow-up case studies in agile companies in order to look deeper into how practitioners diagnose security RE problems and match solutions to them. Next, as we see, practitioners did not search for complex and mathematics-grounded techniques. Instead they applied simple practices with a focus on incorporating those in the social environment of their projects. This could indicate that more research is needed to understand the social aspects of security RE in agile and the ways of leveraging this knowledge for better RE.

VI. Validity Threats

Research methodologists [3,9,10] suggest that external validity is the most important threat in a qualitative research such as ours. To what extent could the findings be similar to observations that we could possibly obtain, if other agile discussion groups on LinkedIn – e.g. the SCRUMstudy group, the Agile Networking Group, Agile Business Analysts group, have been chosen? Or, if personal interviews would have been done with other practitioners from other organizations? We cannot claim universal generalizability of our findings to all possible agile project settings in which security requirements might happen to be engineered. However, if practitioners engaging in online discussions in other LinkedIn groups (e.g SCRUMstudy) or in other platforms (e.g. blogs), are working in agile project organizations that have similar work practices, attitude, and culture to those in which our participants work, one could make observations similar to those in our study. Based on similarities of contexts, we could assume possible similarities in the observations that these contexts would produce about the same phenomena. Second, a threat to validity is the extent to which authors of posts only one-sidedly report on their security RE experiences [9,10]. This might pass bias in our study. We consider this an important issue and therefore think that more research is needed to collect rich contextual information and detailed accounts of practitioners through in-depth interviews.

VII. Summary and Outlook

This study explicated the ways in which agile practitioners reason about security requirements, the factors they consider important when searching for coping strategies and the possible coping strategies that practitioners observed in their projects. We did this based on practitioners' posts in a LinkedIn professional group. Our most important conclusions are that (i) the challenges are traceable to the people in a project, to the agile process and to the nature of security requirements itself; and (ii) the solution strategies rather focus on people and non-technical aspects than on searching for tools and sophisticated methods.

Our immediate future work is to extend the study by including other agile discussion groups on LinkedIn. Plus, we plan to design an interview-based study with practitioners from a broad range of companies, in order to understand more completely the solution strategies and explicate why they work.

References

[1] Z., Racheva, M. Daneva, K. Sikkel, "Value Creation by Agile Projects: Methodology or Mystery?" PROFES'09, 141-155.

[2] W. Alsaqaf, M. Daneva, R. Wieringa, „Quality Requirements in Large-Scale Distributed Agile Projects - A Systematic Literature Review". REFSQ'17, 219-234.

[3] K. Charmaz, "Constructive Grounded Theory", Sage, 2009.

[4] K. Beznosov and P. Kruchten, "Towards agile security assurance", NSPW '04, 47-54.

[5] G. Boström, J. Wäyrynen, M. Bodén, K. Beznosova, P. Kruchten, "Extending XP practices to support security requirements engineering", SESS'06, 11-18.

[6] H. Chivers, R. Paige and X. Ge, "Agile security using an incremental security architecture". XP'05, 1325-1327, 2005.

[7] D. Baca, M. Boldt, B. Carlsson, et al. "A novel security-enhanced agile software development process applied in an industrial setting", ARES'15, 11-19.

[8] A. Vähä-Sipilä. "Software security in agile product management". 2011.

[9] I Verner, J.M., Sampson, J., Tosic,V., Bakar, N., Kitchenham, B.,"Guidelines for Industrially-Based Multiple Case Studies in Software Engineering". RCIS'09, 313-324

[10] N. Hookway, "'Entering the blogosphere': some strategies for using blogs in social research", Qualitative Research, 2008 8(1) 91–113.