Security Requirements Engineering in the Agile Era: How Does it Work in Practice?

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

Chong Wang
School of Computer Science
Wuhan University
Wuhan, China
cwang@whu.edu.cn

***Abstract*—Currently many software companies attempt the integration of agile project delivery models and security requirements engineering (RE). However, very little is published on how this is achieved in real-life settings. This paper reports on results from a documentary study initiated to understand the agile-ready security practices that organizations use. We selected seven well-documented Security RE frameworks for Agile projects that have been used in practice and carried out a qualitative thematic analysis based on documents describing the frameworks and their supposed use in detail. This resulted in a list of solution practices that focus on introducing artefacts, organizational roles, competencies and activities in order to make sure that security RE is done systematically in agile project organizations. Our conclusion is that Security RE adds up to the documentation in an agile project, as teams introduce new story types, e.g. evil user stories, abuser stories, security stories. Plus, we found that Security RE relies on investments into the security training of the agile project teams and into organizing hack sessions. Last, if companies take security requirements seriously, it seems that they should consider ignoring the gatekeeping role of the agile product owner.**

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

I. Introduction

Security requirements engineering (RE) does not fit organically into agile project delivery [1]. While agile development processes focus on high priority issues, they don't mandate secure code style or help prioritize security requirements. In response to this realization, many companies attempted an integration of agile and security RE. For example, Synopsys, a prominent secure solutions provider, formulated a manifesto [13] for secure agile delivery (emulating the style of the original Agile manifesto from 2001): *(1) Rely on developers and testers more than security specialists. (2) Secure while we work more than after we're done. (3) Implement features securely more than adding on security features. (4) Mitigate risks more than fix bugs.* Moreover, highly visible software businesses, e.g. Microsoft and SAP, launched their own agile secure development frameworks [4,5]. Despite of these developments, little research has been done systematically to understand the practices perceived important by companies for the successful integration of agile and security RE. This paper reports on results from a documentary [6] study in which we aimed to understand the agile-ready security practices that organizations use. We posed two research questions (RQ): RQ1. *What coping strategies do companies recommend in order to integrate security requirements and agile?* and RQ2. *What concepts do companies perceive as important for integrating security requirements in agile?* We selected seven well-documented Security RE frameworks for Agile projects that have been used in practice and carried out a qualitative thematic analysis [7] based on documents describing the frameworks and their supposed use in detail. This resulted in: (1) a list of solution practices that focus on introducing artefacts, organizational roles, competencies and activities in order to make sure that security requirements are treated systematically in agile organizations, and (2) a conceptual model that aggregates the knowledge embedded in these practices. Below, we describe background, related work, our qualitative analysis process, results and discussion.

II. Related Work

A few authors [1,8,9,11] present approaches to the integration of agile and security requirements, and evaluation of these proposals in realistic settings. Williams et al [8] proposed the Protection Poker, a collaborative and informal security game for agile teams, which was used and evaluated at Red Hat. Ben Othmane et al [9] propose a method for security reassurance of software increments and evaluate it on a realistic case. 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 [1]: security manager, security architect, security master and penetration tester [1]. Finally, Terpstra et al [11] report a list of solution practices that agile practitioners used in projects where security was an important requirement. Although empirical evidence produced in these studies [1,8,9,11] give us early indication on the solutions that may work in industrial settings, we are lacking a comprehensive understanding of the practices and the concepts that practitioners use when thinking of integration solutions. This motivated our documentary study.

III. Research Process

Our research employed Bowen's document analysis methodology [6]. We also drew inspiration from Verner et al. [12] suggesting the use of publically available data for the purpose of exploratory qualitative research. In this study, we selected 7 well documented agile secure development frameworks put forward by companies or non-profit industry organizations supported by companies. The documentation

978-1-5386-8412-2/18/$31.00 ©2018 IEEE
DOI 10.1109/QuaRAP.2018.00008

available about these frameworks was subjected to thematic analysis [7]. We chose this data collection and analysis strategy because of its fit 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 (e.g. from the web sites of the organizations proposing the frameworks). The 7 frameworks are: (1) Microsoft's Security Development Lifecycle for Agile (SDL) [4], (2) the framework of the Open Web Application Security Project (a non-profit organization focused on improving the security of software) [1,14], (3) PRINCE II for Agile [10] of Axelos, a joint venture company, created by UK's Government Office of Commerce and Capita PLC, (4) the agile security framework of the Software Assurance Forum for Excellence in Code (SAFECode), a non-profit organization whose purpose is to help increase trust in IT products and services through the advancement of effective software assurance methods [15], (5) the Agile Security Framework of the Child Welfare Digital Services (CWDS) organization [16], (6) the framework of the SANS Institute [17], an UK-based organization that is the largest source for information security training and security certification in the world, and (7) the Agile Security approach of the Xebia Group, a global consulting company based in the Netherlands [3]. We chose these frameworks because data about their use in context is publically available and also because the first author got into personal contacts with these organizations and attended presentations on their experience in agile and security requirements in person, and had opportunity to ask clarification questions. We expected that the analysis of the information on these frameworks would provide an understanding about the concepts that form the practice of agile security RE, from practitioners' perspective. We treat each framework as a case, and because of having seven in total, we refer to this study as a multi-case study [12] where the unit of analysis is the framework put forward by each organization. We preferred this documentary research approach over an interview-based case studies or a survey, just because it was cost-effective for our situation, e.g. the documents were in textual form, allowing for analyzing qualitative data immediately without the resource intensiveness of voice recording and transcription [12]. We wanted to obtain indicative results which could be an informative pre-step in the design of a follow-up full-blown interview-based study.

IV. Results

A. Understanding the coping strategies

Table I (see the next page) presents the concepts that indicate coping strategies for security requirements in agile. We make the note that our goal was to collect all possible practices that the seven organizations included in their frameworks and considered as instrumental to the integration of security RE and agile. We did not look at comparison of the frameworks, nor we searched for reasons of why one framework might be called "better" in a particular respect than another. This goal could be pursued through future case study research in organizations that use these frameworks. Table I reports the number of occurrences in a practice across the seven frameworks (see the 3rd and 7th columns). For example practice S1 was pointed out in five frameworks. Next, we found that the coping strategies in Table I, could be divided in the following three groups: ***(G1)*** Solutions addressing the artefacts dealing with security requirements; ***(G2)*** Solutions addressing the human factors in agile projects; and ***(G3)*** Solutions addressing the agile process itself. This is indicated in the 4th and 8th column of Table I.

B. The conceptual model

Using iterative coding practices ([7]), we aggregated the concepts from Table I into a conceptual model (Figure 1). It describes the high-level categories that the 7 frameworks include when recommending solution practices to cope with security requirements in agile projects. We note that (1) as this study is exploratory, the purpose of this model is descriptive only. It explicates the practitioners' understanding of the possible solutions to security RE as per the included frameworks; 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. We note that the boxes in Figure 1 indicate categories which the arrows indicate an "Include" relationship between categories.

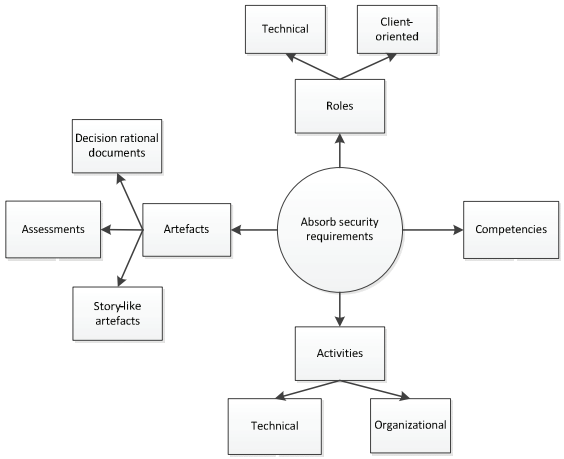

Figure 1. Categories perceived important by organizations when integrating security requirements in agile.

Our analysis suggests that the central overarching category that describes the coping strategies proposed in the 7 frameworks is *Absorb Security Requirements.* It means that the development team absorbs the needs and the responsibility for engineering security requirements. According to the 7 reviewed frameworks, absorbing security means considering the following conceptual categories: (1) security-requirements-specific activities, (2) artefacts, (3) roles, and (4) competencies. The paragraph positioned after Table I, explains each of these conceptual categories. We add in brackets, some examples of concepts from Table I that support these conceptual categories.

TABLE I. CONCEPTS INDICATING COPING STRATEGIES

ID.	Concepts	Nr. of occ.	Group	ID	Concepts	Nr. of occ.	Group
S1	Integrate security features in the definition of 'done'.	5	G1	S24	If you have a requirements board, make the security expert part of the board.	1	G2
S2	Modify the definition of 'done' in response to critical organizational needs such as network and end point data security.	2	G1	S25	Get a security official to join the product owner's meetings.	2	G2
S3	Make security part of the acceptance criteria.	4	G1	S26	Carry out continuous security risk analysis to maintain a security awareness culture.	3	G3
S4	Allocate time for security in every sprint.	5	G3				
S5	Use periodic security sprints	2	G3	S27	Set up a virtual security group composed of everyone with security-related role at the company, to share security issues and get answers to security questions.	1	G2
S6	Use security (user) stories.	3	G1				
S7	Use evil user stories.	1	G1				
S8	Use abuser stories.	3	G1	S28	Educate the business about security risks.	7	G2
S9	Use feature-based treat modelling.	5	G3	S29	Systematically train developers on security engineering topics.	7	G2
S10	Do security impact labelling.	1	G3	S30	Organize security hack sessions.	2	G2
S11	Maintain a sprint security bucket that covers three areas: verification tests, design reviews, and response planning.	2	G3	S31	Implement automated security monitoring process.	6	G3
				S32	Own security requirements as you own user experience requirements.	2	G3
S12	Carry out security risk analysis.	4	G3	S33	Review the code on security.	7	G3
S13	Keep track of security debt. Security debt must paid in future sprints, or addressed as residual risk.	1	G1	S34	Let security specialists use the same whiteboards, sticky notes, or online tools that the development team uses.	2	G3
S14	Prioritize the security risks that are worth protecting.	2	G3	S35	Publish a list of approved tools and associated security checks.	2	G3
S15	Order the backlog based on security assessments	2	G3	S36	Ban unsafe functions to reduce potential security bugs.	1	G3
S16	Document risk acceptance decisions.	2	G1	S37	Certify software prior to release.	1	G3
S17	Associate each security-related story with a list of security tasks in the backlog.	1	G3	S38	Defining minimum acceptable levels of security and privacy quality.	1	G1
S18	Perform operational security tasks in each sprint. These are not directly related to the security stories, but are handled like a continuous maintenance work in a sprint.	1	G3	S39	Induce program failure by deliberately introducing random data.	1	G3
				S40	Carry out attack surface analysis and review.	2	G3
S19	Add a 'security champion' to the development team.	1	G2	S41	Identify and document cryptographic design requirements for the agile project.	1	G1
S20	Introduce a 'security stakeholder' in the dev. team responsible for translating the security requirements into business value.	1	G2	S42	Prepare an incident response plan for the agile project.	1	G1
S21	Introduce a 'security evangelist' to act as a domain expert and help the dev team increase their awareness of security requirements.	1	G2	S43	Take an all-rounder view of security risks, by looking at business processes, monitoring, and legal contracts, and discuss the risk to the business, not the just the parts pertaining to coding.	5	G3
S22	Introduce a 'security master' to work together with the scrum master.	2	G2	S44	Nail down security control points in the agile process.	2	G3
S23	Introduce a 'security architect' translating the business value of security requirements into technically implementable features.	1	G2	S45	Definition of Readiness (a kind of clear order check to see if everything is clear enough to start design / built/ test cycle);	2	G1
				S46	Implement hybrid security and functionality testing throughout the project.	2	G1

1. Activities. All frameworks consider critical the introduction on some security-specific activities, be it organizational e.g. suggesting certain decisions to be made by the development team or the project manager when facing certain circumstances (S16) or technical (e.g. to carry various kinds of analyses and reviews (S12, S33). These activities could be executed either as a dedicated security sprint (S5), or made part of every sprint (S4). E.g. Microsoft's SDL makes a distinction among three types of security-requirements-specific activities: every sprint's practices, one-time practices (for the entire project) and bucket

practices that cover multiple sprints. ***2.Artefacts***. These could be either story-like descriptions (e.g. evil or abuser stories, see S5-S8) complementing the user stories in an agile project, or risk assessments, security debt assessments, and risk-related decisions and their rationale. ***3. Roles***. These are either organizational (security stakeholder, S20) or technical (security master, S23). Roles either mirror the agile project roles of scrum master, domain expert and product owner, or just assign a security specialist in a technical or organizational capacity, to the agile team. ***4. Competencies***. These refer to capabilities that organizations consider to be in place, so that the security requirements are integrated in predictable way. Examples of competencies are [1]: penetration testing, secure architecture, secure domain analysis. We note that this category crystalized while coding the concepts of the OWASP's framework, which is specifically geared towards very large organizations. We added it as a separate category and did not merge it with the Roles category, just because the OWASP's framework made a clear distinction between organizational competencies built up in an organization and the roles of people who exercise them.

V. Discussion

This research has some practical implications. Absorbing security requirements into agile means adding up to the project documentation. Would this decrease the "agility" of agile processes in an organization? We think companies should confront this question, and in turn determine for themselves what would be the right balance of agility and documentation in their organizations. Based on their context, they may choose those practices that match the desired balance between agility and amount of extra artefacts, activities and roles. Second, Table I suggests that training the development team is a pillar in the integration of security RE and agile. One might think that if a company wants to see tangible results, developers' training on security seems to be the most important investment. Last, Table I does not discuss the role of the product owner (PO), in contrast to other studies [11] where educating the PO on security was identified as a solution practice. In fact, some frameworks recommend the newly-introduced security roles match the POs of a project. This makes us believe that tacitly, a collaboration is assumed between the security-focused roles and the PO. Or, there is a tacit assumption that organizations should consider ignoring the gatekeeper's part [1] of the PO's job. As in [11], the PO owns the product backlog and in turn it is well possible that the PO constantly puts new items at the top of the backlog, ignoring the existing backlog (security-related) items completely. How POs may share the ownership over backlogs with security requirements specialists is interesting to understand; case study research in real-world organizations is the best way to gain insights into the interaction of a PO and security roles and competencies in agile. This is a research line for future.

Next, we acknowledge some limitations [6,7] of this work. We looked at 7 frameworks and we think it is likely that if we include more frameworks in our research, we would obtain a larger list of practices and a more detailed conceptual model (Figure 1). We therefore consider Table I as a living document that will grow as our work progresses. Moreover, we think that the competency category in Figure 1 could be elaborated much more. This category is supported by practices recommended in one framework only ([1,14]). Our next activity is twofold: to include more frameworks from other large companies (SAP, Accenture, Oracle and Shell), and to start case study research in the Xebia Group in the Netherlands.

VI. Conclusions

This documentary study discerned 46 practices which support the understanding that security RE can be integrated into agile by introducing either new artefacts, organizational and technical roles, competencies and organizational and technical activities. All of these help security requirements to be absorbed into the agile delivery (Figure 1). Our conclusions are that: (1) Security RE in agile strongly recommends investments in developers' education on security and in writing security-specific artefacts (e.g. evil stories, abuser stories). This, in turn, adds up to voluminous project documentation; (2) the introduction of security roles in agile projects calls for redefining the boundaries of requirements ownership of the PO. POs should be willing to share ownership and power with one or more security roles, or companies may otherwise consider ignoring the PO's gatekeeping role.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 61702378.

References

[1] D. Baca et al., A novel security-enhanced agile software development process applied in an industrial setting, ARES'15, 11-19.

[2] W. Alsaqaf, et al., Quality Requirements in Large-Scale Distributed Agile Projects - A Systematic Literature Review. REFSQ'17, 219-234.

[3] xebia.com/agile-software-security

[4] www.microsoft.com/en-us/SDL/discover/sdlagile.aspx

[5] A. D. Brucker, Agile Secure Software Development in a Large Software Development Organisation, ASSD 2015.

[6] Bowen, G. A. (2009). Document analysis as a qualitative research method. Qualitative Research Journal, 9(2), 27-40.

[7] L. Nowell, et al., Thematic Analysis - Striving to Meet the Trustworthiness Criteria, Int. J of Qual Methods, 16, 2017, 1-13.

[8] L. Williams, et al., Protection Poker: The New, Software Security Game. IEEE Security & Privacy, 8(3), 14-20.

[9] L. Ben Othmane, et al., Extending the Agile Development Process to Develop Acceptably Secure Software. IEEE TDSC. 11(6): 497-509

[10] www.axelos.com/best-practice-solutions/prince2/prince2-agile

[11] E. Terpstra, M. Daneva, C. Wang (2017). Agile Practitioners' Understanding of Security Requirements: Insights from a Grounded Theory Analysis. RE Workshops, 439-442

[12] J. Verner, et al., Guidelines for Industrially-Based Multiple Case Studies in Software Engineering. RCIS'09, 313-324

[13] www.synopsys.com/software-integrity/resources/ebooks/ agile-security-manifesto.html

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

[15] www.safecode.org/publication/SAFECode_Agile_Dev_Security0712.pdf

[16]github.com/ca-cwds/Information-Security/wiki/Agile-Security-Framework

[17]www.sans.org/reading-room/whitepapers/securecode/agile-defensive-perimiters-forming-security-test-regression-pack-35617[19]