An Empirical Perspective on Security Challenges in Large-Scale Agile Software Development

Amber van der Heijden
Eindhoven University of Technology
The Netherlands
a.v.d.heijden@student.tue.nl

Cosmin Broasca
Rabobank
The Netherlands
Cosmin.Broasca@rabobank.com

Alexander Serebrenik
Eindhoven University of Technology
The Netherlands
a.serebrenik@tue.nl

ABSTRACT

Background Agile methods have been shown to have a negative impact on security. Several studies have investigated challenges in aligning security practices with agile methods, however, none of these have examined security challenges in the context of large-scale agile. Large-scale agile can present unique challenges, as large organizations often involve highly interdependent teams that need to align with other (non-agile) departments. **Goal** Our objective is to identify security challenges encountered in large-scale agile software development from the perspective of agile practitioners. **Method** Cooperative Method Development is applied to guide a qualitative case study at Rabobank, a Dutch multinational banking organization. A total of ten interviews is conducted with members in different agile roles from five different agile development teams. Data saturation has been obtained. By open card sorting we identify challenges pertaining to security in agile. **Results** The following challenges appear to be unique to large-scale agile: *alignment of security objectives in a distributed setting*, *developing a common understanding of the roles and responsibilities in security activities*, and *integration of low-overhead security testing tools*. Additional challenges reported appear to be common to security in software development in general or concur with challenges reported for small-scale agile. **Conclusions** The reported findings suggest the presence of multiple security challenges unique to large-scale agile. Future work should focus on confirming these challenges and investigating possible mitigations.

CCS CONCEPTS

• **Software and its engineering → Agile software development**; • **Security and privacy → Software and application security**;

KEYWORDS

agile software development, security management, large-scale agile

ACM Reference Format:
Amber van der Heijden, Cosmin Broasca, and Alexander Serebrenik. 2018. An Empirical Perspective on Security Challenges in Large-Scale Agile Software Development. In *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (ESEM '18), October 11–12, 2018, Oulu, Finland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3239235.3267426

1 INTRODUCTION

Today's market requires businesses to adapt continuously in order to keep up with changing customer demands, while at the same time reducing the time-to-market for delivery of products. Hence, many businesses have started practicing iterative software development based on Agile [5]. In a 2017 survey, over 70% of businesses reported using agile approaches sometimes, often or always [14]. A large variety of agile methods has been proposed to aid businesses in adopting an Agile Software Development Life Cycle. Although these methods differ in their features and application domains, all of them share core principles highlighted in the Agile Manifesto [5]. Previous studies have indicated that agile methods improve product quality and team productivity [12, 13], communication [13], and knowledge sharing [12]. However, other studies have signaled a negative impact of agile practices on security[6, 10].

Beznosov and Kruchten [6] have examined security practices in the agile context and found that over half of the tested security practices conflict with the agile methods. Similarly, Goertzel et al. [10] identified potentially negative security implications for 7 out of 13 core principles of the Agile Manifesto [5]. However, these theoretical analyses did not provide insights in the security challenges encountered in agile software development. Bartsch [3] addressed this by conducting in-depth interviews with agile practitioners from small-size companies and reported problems with customer involvement, emerging requirements, implicit security requirements and security awareness and expertise among developers.

We take a complementary perspective and focus on *security challenges in large scale agile.* Unlike small-scale agile, large-scale agile often involves multiple highly interdependent teams that are required to align with non-agile organizational departments. These considerations contribute to unique challenges common in large-scale agile such as diverging agile approaches across teams [8].

We consider therefore the perspective of agile practitioners and answer the following questions:

RQ1: What challenges do agile practitioners identify in addressing security in large-scale agile development?
RQ2: To what extent are these challenges unique to large-scale agile, agile in general and secure development in general?
RQ3: Are there differences in challenges identified by practitioners from different teams or roles?

To answer these questions we conduct semi-structured interviews with agile practitioners working in software development at

ESEM '18, October 11–12, 2018, Oulu, Finland

ACM ISBN 978-1-4503-5823-1/18/10…$15.00
https://doi.org/10.1145/3239235.3267426

Rabobank. We perform open coding of these interviews in order to identify security challenges present in large-scale agile.

The remainder of this paper is organized as follows. Section 2 presents the research design, Section 3 summarizes the security challenges identified (**RQ1**), Section 4 discusses them to address **RQ2** and **RQ3** and Section 5 concludes.

2 EMPIRICAL RESEARCH DESIGN

2.1 Approach

We employ Cooperative Method Development (CMD) [9] as it provides guidance on how to use qualitative research methods to facilitate process improvement in software engineering. CMD enables the collection of in-depth information about the everyday work practices within the software development teams situated in their specific context. It describes an action research cycle of three phases adapted to software engineering: (i) understanding practice, (ii) deliberate improvements, and (iii) implement and observe improvements. In this *emerging results* paper we report on Step (i).

2.2 Context

Rabobank is selected as a case study, as it provides a typical example [15] of traditional unwieldy financial organizations [2] aiming to undergo an organization-wide agile transformation. We focus on Rabobank IT Systems—Wholesale & International (W&I). W&I is responsible for more than 50 critical wholesale banking applications of which the majority have high availability and integrity requirements. These systems are extremely sensitive to service disruptions, data breaches and data integrity issues. Security officers address these matters as external stakeholders of agile teams by expressing the security requirements for each system. Agile teams at W&I are free to implement the agile methodologies that suit their project best. While Scrum [16] is ubiquitous, frameworks such as Kanban [1] and Extreme Programming [4] are commonly applied as well.

2.3 Semi-structured interviews

2.3.1 Interview topics. Interviews are conducted in accordance with the guidelines for case study research in software engineering [15]. We perform semi-structured interviews such that a pre-fixed set of topics based on findings from prior literature could be discussed while allowing for the possibility to focus on certain specifics and explore other topics emerging from the discussion. The complete list of interview questions is available online[1].

We consider five major interview topics. By considering the broad spectrum of interview topics we expect to identify a comprehensive set of security challenges encountered in practice.

Agile and security Goertzel et al. [10] identified mismatches between seven out of thirteen core agile principles of the Agile Manifesto [5]. We discuss each of these seven principles to investigate to what extent the interviewee feels that security is impacted by the implementation of these agile principles within the organization.

System vulnerabilities and testing: Practices such as code/test cases as documentation, (automated) security testing, and security focused code peer reviewing have been suggested to reduce the presumed negative impact of agile principles on security [3, 6]. We aim to identify to what extent are these practices adhered to.

Team security awareness and expertise To reduce the presumed negative impact of agile principles on security and improve security awareness Bartsch suggested such practices as implicit and explicit knowledge sharing, and self-education [3]. Discussing this topic we investigate whether such practices are adhered to and how well they are integrated in Rabobank processes. We also check the level of security awareness and knowledge typically present in teams.

Product owner involvement and expertise Close product owner involvement is mentioned to reduce the presumed negative impact of agile principles on security [3]. We discuss this to get insight into how security requirements are generated and prioritized and how much the product owner is/should be involved in this process.

Current and future security assessment practices Finally, we asked about the limitations of current security practices, how these limitations can be overcome and what is needed to ensure successful implementation of the envisioned improvements.

2.3.2 Selection of the interviewees. Differences in team culture and role responsibilities may result in different perspectives on the topics of interest. We have considered seven different roles. *Solution Architects* are responsible for developing technical solution that fit into existing architecture. *Product Owners* communicate business needs to the development team, and report back to the business about the product development. *Business Analysts* translate the business requirements from the Product Owner into IT requirements. *Software Engineers* write the code for the products under development. *System Owners* are responsible for IT of the system and act as a supplier to the Product Owner. *Test Managers* are responsible for the planning of and reporting about test activities. *Scrum Masters* are responsible for guiding the intra-team agile process.

Given the spectrum of role responsibilities, we apply maximum variation sampling [11] as a purposive sampling technique to obtain a large variation across both teams and project roles, and thus of perspectives. We have contacted five different teams, and conducted two interviews per team. We refer to the interviewees from the first team as I11 and I12, . . . , from the fifth team—as I51 and I52. Initial contact with potential participants was established face-to-face or via email, and followed by a formal email invitation for participation in a ±45 minute interview. We have ensured broad representation of different roles and interviewed a business analyst (I11), a test manager (I12), a solution architect (I41), and multiple software engineers (I21, I22, I31), system owners (I32, I52) and product owners (I42, I51). All interviewees were male.

To ensure that no answers could be rehearsed in advance, the interview guide was not supplied to the participants. Interviewees were given the choice to respond in English or Dutch such that they could express themselves in the most comfortable manner [17]. All interviews were audio-recorded, translated when necessary and transcribed. The transcript was returned to the interviewee for comments or corrections. No repeated interviews were performed.

2.4 Data analysis

Open card sorting [18, 19] is used to structure all collected interview data in a systematic manner. For each of the interview transcripts, short coherent text fragments are printed on physical cards meant

[1] http://www.win.tue.nl/~aserebre/ESEM2018AmberAppendix.pdf

to be sorted into groups that correspond to themes. We start with no presupposed themes and conduct iterative rounds of interviews and subsequent card sorting sessions until data saturation is achieved, i.e. when no new groups/themes emerge from the card sorting.

2.5 Validation

Preliminary validation of the results is obtained through quality feedback sessions at security officer team meetings at Rabobank.

3 RESULTS (RQ1)

Agile and security Interviewees mentioned a number of challenges related to addressing security in the ongoing agile transformation of Rabobank. One of the reported issues is that management still commits to fixed time and budgets for product delivery. This often results in few resources being spend on security considerations. I12 (Test Manager) illustrates this by articulating his team view that "security is not currently seen as part of working software, it only costs extra time and it doesn't provide functionality". Another challenge reported by interviewees is unclarity regarding accountability for security actions. Not all teams have a security officer who is closely involved with the team, however, these teams are still trusted to properly take care of security concerns. This sometimes results in blurred lines regarding the amount of information that is required to be formally recorded. With less focus on providing extensive (security) documentation typical for agile, ineffective knowledge sharing between security officers and agile team members is especially problematic. I42 (Product Owner) explains, "When I have to explain to security management what exactly we have implemented, then all I can do is direct the security officer to the person who did it or who tested it, but I just don't really know [where else to find that information]."

System vulnerabilities and testing According to the interviewees, not many security testing activities were present besides the required penetration test for highly classified systems and the general system security assessments. Benjamin stated that security is only tested if an item for a certain security feature is present on the product backlog. None of the interviewees mentioned performing other security-enhancing activities commonly recommended for agile projects [3, 6], such as performing security code reviews, using security control libraries, or the use of tools for static/dynamic code analysis. The majority of the interviewees expressed an interest in using automated security tests if these were available in the future.

Team security awareness and expertise Interviewees reported wide variations in security awareness and expertise among team members. One interviewee explained that this could be partly attributed to the high turnover in development teams, as it is common to include external experts employed by third-party consultancy companies in development teams. I11 (Business Analyst) states that no specific attention is being paid to security unless enforced from above because "many software engineers are external employees, and they do exactly what they are asked to do, they just have a general attitude of 'your wish is my command' ". The interviewees generally agree that more could be done to provide security education and training to employees. Without prompting, several interviewees mentioned training as an important factor for increasing security awareness and expertise. In addition, interviewees report that in case the information security officer is unavailable they do not know of a central (online) hub for security-related information, such as secure coding practices. I21 (Software Engineer) demonstrated the need for such a central hub by saying that "if such information was readily available, everyone would keep it in mind." Interviewees from teams that frequently have a dedicated information security officer present, generally agree that the close involvement of the information security officer highly benefited the quality of the applications and the security awareness of these teams. I42 (Product Owner) illustrated the advantage of having a security officer close to the team as follows: "when security officers have time to explain why they exist and why they have certain desires, the team becomes much more accepting, and understanding of why certain things are the way they are."

Product owner involvement and expertise Product Owners were found to contribute to security by supplying domain knowledge regarding the production context of the systems under development. Security awareness and involvement, however, highly varied among Product Owners. Interviewees generally reported preferring a highly involved Product Owner over a more distant Product Owner. Several interviewees mentioned that the Product Owner is often not aware enough of the added business value for performing certain security actions. In those situations teams rely on the "System Owner to coerce the Product Owner by indicating that it [a certain security feature] is a requirement before going live [with a system]" to be allowed to spend some time on security.

Current and future security assessment practices Several issues concerning information exchange between security management and development teams were mentioned. Concerns were expressed about unclear and too technical security requirements as formulated by security management. I32 (System Owner) stated, "One thing I noticed during the security assessments is that it [the list of security requirements] is very hard to understand... and I've been doing this for quite some time within multiple teams, so this is not the first time I'm dealing with this. It's just too complex, it's not simple, and not unambiguous." I31 (Software Engineer) comments that "they [security officers] want to keep it generic, but for each application it applies differently". Furthermore, several interviewees expressed concerns about the lack of understanding between the information security officers and team members during the system security assessments, as I32 (System Owner) illustrated, "I will send some people from my team to such an assessment, and when they return and I ask about how the meeting went, they indicate that it feels like they were chasing different goals." Finally several interviewees mentioned that in global systems when multiple security officers (i.e. regional/local/global) had an interest in the security of the system, situations arose in which the team received conflicting requests from different security officers.

4 DISCUSSION

The empirical results suggest several perceived security challenges in large-scale agile development as reported by agile practitioners. These findings extend prior work with a more comprehensive view of security challenges in agile implementations of all sizes.

4.1 Uniqueness of challenges identified (RQ2)

We distinguish between three categories of security challenges identified, i.e., general security challenges, security challenges in agile, and security challenges unique to large-scale agile.

General security challenges These are security challenges independent of the development approach. An example of such challenge reported by the interviewees is "encouraging resource allocation to security". Indeed, project management compromising on security due to limited resources has been reported as a challenge common to software engineering projects of any type [7].

Security challenges in agile The following challenges identified in our study have also been recognized in the previous study of security challenges in small-scale agile [3]: (i) implementing low-overhead security documentation; (ii) spreading security awareness and expertise in teams; (iii) formulating clear security requirements; and (iv) fostering Product Owner commitment to security.

Security challenges unique to large-scale agile Finally, three security challenges reported by interviewees have not been discussed in earlier work: (i) alignment of security objectives in a distributed setting; (ii) developing a common understanding of roles and responsibilities in security activities; and (iii) integration of low-overhead security testing tools. These challenges can be traced back to more general challenges unique to large-scale agile. Security challenges (i) and (ii) are related to general coordination challenges in a multi-team environment (Section 4.2.4 [8]) and security challenge (iii) is related to general quality assurance challenges (Section 4.2.8 [8]).

4.2 Differences across teams and roles (RQ3)

Different teams have varying opinions on the degree of security awareness and expertise of the team's Product Owner. This is to be expected as different teams have different Product Owners.

Respondents in different agile roles perceive the challenges differently. The security challenges encountered by Software Engineers and System Owners, were more hands-on in nature, e.g., how to code securely, and what sorts of security documentation should be formally recorded. This could be explained by the focus of these roles on functionality. On the other hand, challenges encountered by Product Owners, the role that in practice is closer to business operations than to IT, were more organizational in nature, e.g., how to decide when to involve the information security officer in decision-making, and how to deal with differing stances on security from stakeholders. Different agile roles, hence, may have differing needs in terms of security guidance, education and support.

4.3 Threats to validity

As any empirical study our work is subject to threats to validity.

Internal validity All research was performed solely by only one researcher, which could have subjectively influenced the research results. It should also be noted that since most interviews were conducted in Dutch, the translation of statements into English may have affected the interpretation of their meaning.

External validity As an exploratory case study we formulate hypotheses (e.g., presence of unique security challenges in large-scale agile or association between agile roles and challenges perceived) to be confirmed or refuted by a follow up study.

5 CONCLUSIONS

We performed an empirical investigation of security challenges in large-scale agile. To this end we conducted ten semi-structured interviews with agile development team members at Rabobank.

While many challenges identified can be traced back to secure software development in general, or are similar to security challenges previously reported for small-scale agile, we have also identified three security challenges that appear to be unique to large-scale agile, namely *alignment of security objectives in a distributed setting*, *developing a common understanding of the roles and responsibilities in security activities*, and *integration of low-overhead security testing tools*. These unique challenges might stem from more general challenges found in large-scale agile [8].

Future research should aim to confirm the uniqueness of these security challenges and investigate whether solutions aimed at resolving general challenges in large-scale agile can be expected to resolve the associated security challenges in large-scale agile as well. Interesting directions would be to compare the security challenges identified at Rabobank, a highly regulated financial institution with a flat organizational structure, with results of case studies at lowly regulated high-tech companies and hierarchically organized financial institutions.

REFERENCES

[1] D. J. Anderson. 2010. *Kanban: successful evolutionary change in your software business.* Blue Hole Press.

[2] M. Angelshaug and T. Saebi. 2017. The Burning Platform of Retail Banking. *The European Business Review* (May–June 2017), 30–34.

[3] S. Bartsch. 2011. Practitioners' Perspectives on Security in Agile Development. In *International Conference on Availability, Reliability and Security.* 479–484.

[4] K. Beck. 2000. *Extreme programming explained.* Addison-Wesley.

[5] K. Beck. 2001. Agile Manifesto. *The Agile Manifesto* (2001), 2001.

[6] K. Beznosov and Ph. Kruchten. 2005. Towards Agile Security Assurance. In *Workshop on New Security Paradigms.* 47–54.

[7] P. T. Devanbu and S. Stubblebine. 2000. Software Engineering for Security: a Roadmap. In *Future of Software Engineering.* 227–239.

[8] K. Dikert, M. Paasivaara, and C. Lassenius. 2016. Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review. *J Syst Software* 119 (2016), 87–108.

[9] Y. Dittrich, K. Rönkkö, J. Eriksson, C. Hansson, and O. Lindeberg. 2008. Cooperative Method Development: Combining Qualitative Empirical Research with Method, Technique and Process Improvement. *Empir Softw Eng* 13, 3 (2008), 231–260.

[10] K. Goertzel, T. Winograd, H. L. McKinley, L. Oh, M. Colon, T. McGibbon, E. Fedchak, and R. Vienneau. 2007. *Software Security Assurance: State-of-the-Art Report.* Technical Report.

[11] A. Koerber and L. McMichael. 2008. Qualitative Sampling Methods: A Primer for Technical Communicators. *J Bus Tech Comm* 22, 4 (2008), 454–473.

[12] G. Melnik and F. Maurer. 2005. A Cross-Program Investigation of Students' Perceptions of Agile Methods. In *ICSE.* 481–488.

[13] K. Petersen and C. Wohlin. 2010. The Effect of Moving from a Plan-Driven to an Incremental Software Development Approach with Agile Practices: An Industrial Case Study. *Empir Softw Eng* 15, 6 (2010), 654–693.

[14] Project Management Institute. 2017. *Success Rates Rise - 9th Global Project Management Survey.* Technical Report. 32 pages.

[15] P. Runeson and M. Host. 2009. Guidelines for Conducting and Reporting Case Study Research in Software Engineering. *Empir Softw Eng* 14, 2 (2009), 131–164.

[16] K. Schwaber. 1997. SCRUM Development Process. *Business Object Design and Implementation* (1997), 117–134.

[17] J. Singer and N. G. Vinson. 2002. Ethical Issues in Empirical Studies of Software Engineering. *IEEE T Software Eng* 28, 12 (2002), 1171–1180.

[18] Parastou Tourani, Bram Adams, and Alexander Serebrenik. 2017. Code of conduct in open source projects. In *SANER.* IEEE Computer Society, 24–33.

[19] T. Zimmermann. 2016. Card-sorting: From Text to Themes. In *Perspectives on Data Science for Software Engineering.* Elsevier, 137–141.