Chapter 46
Security Assurance in Agile Software Development Methods:
An Analysis of Scrum, XP, and Kanban

Kalle Rindell
University of Turku, Finland

Sami Hyrynsalmi
https://orcid.org/0000-0002-5073-3750
Tampere University of Technology, Finland

Ville Leppänen
University of Turku, Finland

ABSTRACT

Agile software development was introduced in the beginning of the 2000s to increase the visibility and efficiency software projects. Since then it has become as an industry standard. However, fitting sequential security engineering development models into iterative and incremental development practices in agile methods has caused difficulties in defining, implementing, and verifying the security properties of software. In addition, agile methods have also been criticized for decreased quality of documentation, resulting in decreased security assurance necessary for regulative purposes and security measurement. As a consequence, lack of security assurance can complicate security incident management, thus increasing the software's potential lifetime cost. This chapter clarifies the requirements for software security assurance by using an evaluation framework to analyze the compatibility of established agile security development methods: XP, Scrum, and Kanban. The results show that the agile methods are not inherently incompatible with security engineering requirements.

DOI: 10.4018/978-1-6684-3702-5.ch046

INTRODUCTION

During the last decade, agile software development methods have become an industry *de facto* standard. The aim of these methods has been to improve efficiency as well as transparency of software development (Abrahamsson et al., 2002). The methods promote iterative development and informal interaction, and put a lower or even negative value to strict processes. This is particularly stressed in cases where documentation is used as a means of communication, whether used to convey the customer requirements to the development team, or for communication within the team itself, e.g., in the form of specifications (Beznosov and Kruchten, 2004; Ko et al., 2007; LaToza et al., 2006).

Introducing strict security requirements to the software development process usually results in creation of excess security assurance, such as a formal security architecture, out of necessity to fulfill the strict external security criteria. Integrating the security requirements, such as reviews, security testing, processes and documentation into an agile method, the cost of the development effort is very likely to increase (Beznosov and Kruchten, 2004). The entire extra 'management overhead' is in direct contradiction with agile methods' core philosophy of leanness and informality (Beck et al., 2001). Thus, applying the security processes to the agile or lean development methods has the potential of rendering the methods, by definition, something that is neither agile nor lean.

On the other hand, the need for software security has been always one of the main drivers in software development. While quality assurance remains a key process to ensure software robustness, effectiveness and usability, security assurance provides the means to develop and deploy software components and systems that protect the system's data, their users' privacy and the system resources.

The operating environment of the software products and services has been evolving and changing due to extensive use of the Internet and public services as well as the ever-increasing pervasiveness and ubiquitous characteristic of software solutions. In addition, the software industry itself has gone through an unprecedented shift from sequential development methods (e.g. waterfall-type) towards iterative and incremental software development methods (e.g. agile and lean). In addition, due to the large scale adaptation of agile methods in the industry (Licorish et al. 2016, VersionOne 2018), the new agile development methods seem to be able to reclaim at least some of their claimed benefits.

Furthermore, the need for security has also been realized in the form of several commercial, international and national standards. To comply with these, several security frameworks and security-focused development methods have been presented. However, knitting together strict security engineering practices and adaptable agile software methods is not straightforward and may cause remarkable problems.

Furthermore, the selection of a software development method to be used in a development project has consequences into the software architecture and design. While the manifesto for agile software development states that the best architectures and design emerges from self-organized teams (Beck et al., 2001), this statement has been often criticized. For example, renowned software engineering researcher Philippe Kruchten (2010) has repeatedly questioned whether the concept of 'agile architecture' combines two incompatible approaches. In the context of security sensitive projects, this question is even more topical as it is a hard and arduous task to embed security into a product afterwards.

Therefore, the objective of this chapter is to study how well the selected agile methods are adaptable to security development practices. For the purposes of this study, we have selected three widely-used development methods, Scrum, XP and Kanban. We use Microsoft Secure Development Lifecycle (SDL) model as a benchmark for the evaluation – as the model is designed for high regulation environment and