Chapter 47
Fitting Security into Agile Software Development

Kalle Rindell
Informaatioteknologian laitos, University of Turku, Turku, Finland

Sami Hyrynsalmi
Tampere University of Technology, Pori, Finland

Ville Leppänen
Department of Information Technology, University of Turku, Turku, Finland

ABSTRACT

Security objectives in software development are increasingly convergent with the business objectives, as requirements for privacy and the cost of security incidents call for more dependable software products. The development of secure software is accomplished by augmenting the software development process with specific security engineering activities. Security engineering, in contrast to the iterative and incremental software development processes, is characterized by sequential life cycle models: the security objectives are thus to be achieved by conflicting approaches. In this study, to identify the incompatibilities between the approaches, the security engineering activities from Microsoft SDL, the ISO Common Criteria and OWASP SAMM security engineering models are mapped into common agile software development processes, practices and artifacts.

1. INTRODUCTION

Software development organizations are hard pressed to meet the increasing demand for secure software (Boehm and Turner, 2005; Subashini and Kabitha, 2011; Fitzgerald and Stol, 2014). Value-driven software development processes are seen lacking in ability to produce secure software, essentially a risk-based process. Responsibility for software security is placed on elements external to the development teams (Beznosov and Kruchten, 2004), deepening the separation of business objectives and security objectives in software development. In agile development, the lessened emphasis to preliminary planning,

DOI: 10.4018/978-1-7998-3016-0.ch047

and the absence of fixed milestones may cause difficulties incorporating external security processes into the iterative development processes: organizations may effectively end up running a non-agile security development life cycle along the agile software development processes. Aligning the business and security objectives, and aligning and integrating the activities is necessary to avoid sacrificing neither the efficiency of the agile processes, nor the long-term security objectives.

Agile software development processes call for agile organization, infrastructure and business models according to Baskerville et al. (2005). Self-organizing teams and non-deterministic implementation processes result in task implementation patterns remarkably different from those produced by sequential and pre-planned counterparts of these models. In addition to the organizational dissimilarities, security engineering processes are ultimately driven by risk rather than business value; unlike the agile development processes, they also rely on planned activities executed in a sequence as first outlined by Viega and McGraw (2002), and Howard and Brooke (2006). Sequential software development methodologies aim to reduce the security risk by executing pre-planned tasks at fixed points in the development life cycle. Lightweight, iterative, and incremental processes utilize a profoundly differently structured implementation and verification cycle; thus, security mechanisms fully integrated into agile development processes are required. There exists no inherent obstacle to utilizing agile processes to achieve the security objectives: implement the required security functionality and security assurance, and verify the absence of known security vulnerabilities (cf. Savola et al., 2012).

The agile methods improve productivity by narrowing the scope of implementation into specific features within a fixed time frame (Abrahamsson et al., 2002). Focused development allows for meticulous concentration on the quality and functionality selected into the iteration backlog. By selection of the tasks, the team and the customer can be reasonably assured that the work is done in order to achieve the objectives currently considered most important for the software product under development.

The differences between the methodologies have been broadly categorized: in one categorization approach by Boehm and Turner (2003), software development methods are considered to be either risk-driven or value-driven; hybrid models, such as Disciplined Agile Delivery by Ambler and Lines (2012), set out to reintroduce a set of planned activities – a sequential element – extra character x into the iterative work flow. To find out the reasons for the difficulties experienced by the software and security engineers, software security processes must first be defined, and the activities analyzed. These differences between the approaches, values and even the paradigms of software engineering and system engineering methodologies lead to the primary research question:

RQ: How are the software security engineering activities integrated into the agile practices?

This question, and the motivation of the research, stems from empiric reports, according to which the software development processes run into difficulties when charged with security objectives (see e.g. Türpe and Poller, 2017; Lorünser et al., 2018). Tying security engineering activities directly into the software development activities, albeit only in a theoretical framework, has the potential to make security easier to adopt for software developers, and help providing the "security mindset" necessary for successfully and efficiently applying a security development lifecycle in a software development project.

The research question is considered and discussed primarily from the viewpoint of agile software engineering in the following chapters. In Chapter 2, the issues in software security and the current adaptation of agile software security engineering activities, practices and artifacts are examined. Chapter 3 presents used research approach. In Chapter 4, an exhaustive list of common software security activities