Chapter XII
An Agile Perspective on Open Source Software Engineering

Sofiane Sahraoui
American University of Sharjah, UAE

Noor Al-Nahas
American University of Sharjah, UAE

Rania Suleiman
American University of Sharjah, UAE

ABSTRACT

Open source software (OSS) development has been a trend parallel to that of agile software development, which is the highly iterative development model following conventional software engineering principles. Striking similarities exist between the two development processes as they seem to follow the same generic phases of software development. Both modes of development have less emphasis on planning and design and a more prominent role for implementation during the software engineering process. This chapter expounds on this connection by adopting an agile perspective on OSS development to emphasize the similarities and dissimilarities between the two models. An attempt is first made to show how OSS development fits into the generic agile development framework. Then, the chapter demonstrates how the development process of Mozilla and Apache as two of the most famous OSS projects can be recast within this framework. The similarity discussed and illustrated between agile and OSS development modes is rather limited to the mechanics of the development processes and do not include the philosophies and motivations behind development.

INTRODUCTION

As conventional software development methodologies struggle to produce software within budget limits and set deadlines, and that fully satisfies user requirements, alternative development models are being considered as potentially more effective. One such model comes under the general umbrella of agile software development, which prescribes a highly iterative and adaptive development process that adapts not only to the changing software requirements and

operating environments, but also to the "collective experience and skills" of people working in the development teams (Turk, France, & Rumpe, 2005). Proponents of agile methods advocate the superiority of their model in delivering quality software, produced at an economic cost within a fast development period and meeting evolving customer requirements.

A parallel trend to agile software development has been that of open source software (OSS) development, which looks a priori as a random and chaotic process harnessing the abundance of programmers on the Internet to produce software that is deemed of very high quality. However, upon a closer look at both processes, agile and open source, striking similarities exist in terms of the development process itself. Indeed some research has already pointed out that OSS development, although driven by different motivations and economic considerations than agile methods, follows the same generic phases of agile methodologies (Warsta & Abrahamsson, 2003). In this chapter, we expound on this connection by adopting an agile perspective on OSS development. This is not to confuse the two paradigms, which remain distinct, but to emphasize the similarities and dissimilarities between the two approaches to software engineering.

In the first part of the chapter, we attempt to retrofit OSS development within a generic agile software development framework. In the second part, we demonstrate through the example of two landmark open source projects, Mozilla and Apache, how OSS development processes can be recast within the generic agile development model.

BACKGROUND

An Agile Perspective on OSS Development

Agile development implies developing simple designs and starting the coding process immediately. Frequent stops are made to assess the coding process and gather any new set of features or capabilities from clients in view of incorporating them into the software through iterations rather than following a single formal requirements document (Lindquist, 2005). Some of the most prominent agile software development methods are extreme programming (XP), Scrum, feature-driven development (FDD), and adaptive systems development (ASD; Ambler, 2002). Through plotting these agile software development methods into a generic framework for software development (see Table 1), we identified four common phases to all agile processes, which we termed the generic agile development model (see Figure 1). These phases are outlined as follows:

1. **Problem exploration:** Includes overall planning, requirements determination, and scheduling
2. **Iterative development:** Repeated cycles of simple design, coding, testing, a small release, and refining requirements
3. **Version control:** At the end of one iteration or a few concurrent or consecutive iterations, changes are committed to the final program and documented, probably delivering a working version to the customer (possibly installed for use until development ceases).
4. **Final release:** When changes can no longer be introduced to the requirements or operating conditions

Open Source Development from an Agile Perspective

In general, the fundamental difference between open source and conventional software development is that the extremely emphasized and revisited steps of planning, analysis, and design in software engineering are not part of the general open source life cycle; the "initial project founder" is the one who conducts these steps in a brief and oversimplified manner (O'Gara, 2002).