Chapter 3
Values and Principles Practices in Agile and Open Source Development

3.1 INTRODUCTION

Values are ideals that a group of people embrace. They can be positive or negative, for example empowerment or control. These values are implicit in the personality or culture of a company. Values are often emotive; they represent driving forces behind people. The word principle stems from the Latin for leader or emperor, however in this context we mean it as a general law or essence, for instance principles of modern physics.

Values and principles are related to practices, which are sets of repeatable actions you perform, e.g., practice developing software by driving with tests.

A practice works in a given context due to an underlying principle. For example, the practice of continuous integration is backed by the principle of reducing bottlenecks to enable flow in software development. Practices produce effects that support one or more values. If a software company values the ability to meet their customer needs, then a practice such as Test Driven Development (TDD) (Beck, 2002) will support that value as it keeps the cost of change low over time. In early 2001, a group of industry experts met to outline the values and principles that would allow software teams to develop quickly and respond to change. They called themselves the Agile Alliance. Over two days they worked to create a statement of values. The

DOI: 10.4018/978-1-59904-681-5.ch003

Copyright © 2010, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
result was the manifesto of the Agile Alliance\(^1\). The document provides a philosophical foundation for effective software development. In the next paragraphs of this chapter, we review values and principles of the Agile Manifesto to determine the extent to which they apply to OSD.

### 3.2 VALUES IN AGILE AND IN OPEN SOURCE DEVELOPMENT

AMs are centered in four major values. Here we briefly introduce them. A comprehensive discussion is in the two editions of Beck’s book (Beck, 1999; Beck, 2004):

1. **Communication:** Developers need to exchange information and ideas on the project among each other, to the managers, and to the customer in a honest, trusted and easy way. Information must flow seamless and fast.

2. **Simplicity:** Simple solutions have to be chosen wherever possible. This does not mean to be wrong or to take simplistic approaches. Beck often uses the aphorism “simple but not too simple”.

3. **Feedback:** At all levels people should get very fast feedback on what they do. Customers, managers, and developers have to achieve a common understanding of the goal of the project, and also about the current status of the project, what customers really need first and what are their priorities, and what developers can do and in what time. This is clearly strongly connected with communications. There should be immediate feedback also from the work people are doing, that is, from the code being produced – this entails frequent tests, integrations, versions, and releases.

4. **Courage:** Every stakeholder involved in the project should have the courage (and the right) to present her/his position on the project. Everyone should have the courage to be open and to let everyone inspect and also modify his/her work. Changes should not be viewed with terror and developers should have the courage to find better solutions and to modify the code whenever needed and feasible.

Agile Modeling (Ambler, 2002) adds a further value: humility. This value states that every one has equal value on a project.

#### 3.2.1 Communication

These values are present in various ways in Raymond’s description of OSD (Raymond, 2000). We will now analyze them one by one and evidence the points of contact. The very same concept of OS is about sharing ideas via the source code, which be-
Are Developers Fixing Their Own Bugs?: Tracing Bug-Fixing and Bug-Seeding Committers
www.igi-global.com/chapter/developers-fixing-their-own-bugs/74664?camid=4v1a