Chapter 9
Design Approaches

9.1 INTRODUCTION

In this chapter we compare agile and OS development in terms of the adoption of design practices.

We review the practices of AMs to identify the agile approaches to the design and we inspect the code of a set of open source projects to determine whether these approaches are undertaken by OS projects.

Software design is a process that defines the solutions to software implementation at the early stages of the software development process. It comprises software requirements and software architecture modeling. In the waterfall approach this process is in general realized with the Big Design Up Front (BDUF). The method consists in creating big structured models of design before any coding to ensure the transparency of the overall software development. In literature there are various models of software design. The most known are diagram specifying requirements, architecture of the system, components, technologies, classes and interfaces. In object oriented programming the design is implemented by the series of UML diagrams – the structure, the behavior, and the interaction diagrams (Ambler, 2004; Fowler, 1999).

DOI: 10.4018/978-1-59904-681-5.ch009
A modern concept of design modeling is performed through the design patterns. A design pattern is a solution to a problem of design that repeatedly occurs and that can be implemented in the code. The model in general is independent from a specific language and a given application domain, but it might be related to a type of programming like the object oriented programming. The design pattern is used to speed up the process of development reusing stable solutions to specific problems of development. A design pattern is not an algorithm though as is not related on computational solution of the conceptual system.

In object oriented programming the major reference on design patterns is the book of the *Gang of Four* (Gamma et al., 1994). The design patterns are classified by the type of problem they solve. In Gamma et al. (1994), patterns are classified in three categories: structural, behavioral, and creational.

The structural patterns relate to class and objects and their composition, the behavioral patterns refer to class communication, and creational patterns concern class instantiations.

In the following section we describe the design practices and patterns used in the agile development.

### 9.2 Agile Approaches to Design

Agile Methodologies focus on incremental development without a single and large upfront design. Namely, they adopt the Big Design Up Front Anti-pattern (BDUFA) that embrace changes adopting envisioning modeling of design (requirements and architecture) just when needed. The usual approach is to mix and refine with short iterations the design, coding, and the testing phases. Therefore, the code is subject to change frequently, whenever requirements change due to a deeper understand or because the customer has changed idea. As such the design approach in the agile methods can be readily identified in the source code. In this chapter we shall deduce whether the developers of OS adopt BDUFA by analyzing the code changes.

In the following we shall just give some examples and discuss the most famous facts related to design approaches in AMs.

In general, AMs use design patterns when the language of programming is object oriented. Namely, AMs share the common principle of reusing existing working objects from previous projects or project iterations to avoid waste and useless activities. This applies also to design objects proving the use of design patterns successful used in previous project’s iteration or projects in AMs.

A design practice peculiar to development with the AMs concerns testing. Many of the AMs embrace testing in the early phase of the development and all across the development itself. In this sense, AMs extensively use the concept of acceptance