Chapter 6
Other Agile Methods

6.1 INTRODUCTION

The most well known AMs are Extreme Programming (XP) (Beck, 1999) and SCRUM (Schwaber & Beedle, 2001) but there are several more (Abrahamsson et al., 2002):

- Crystal (Cockburn, 2004)
- Dynamic System Development Method (DSDM) (Stapleton, 1997)
- Lean Software Development (LSD) (Poppendieck & Poppendieck, 2003)
- Feature Driven Development (FDD) (Palmer & Felsing, 2002)
- Agile Modeling (AM) (Ambler, 2002)
- …

In this chapter we briefly summarize some of them highlighting their specific features.

DOI: 10.4018/978-1-59904-681-5.ch006

Copyright © 2010, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
In the early ‘90s, the IBM Consulting Group hired Alistair Cockburn to build a methodology for object-oriented development. Cockburn investigated a large number of software projects and asked each team to identify the main reasons for their own success. Cockburn has defined Crystal (Cockburn, 2004) as a family of AMs, because he believed that different kinds of projects require different development methodologies.

All the Crystal methodologies are based on the following paradigm: “strong on communication, light on work products”. Compare to XP (Beck, 1999), the Crystal family shows many differences: XP is based on a well defined set of development rules, on the contrary, Crystal does not include such rigid constraints but gives a lot of freedom to the development team. Hence, this methodology allows a greater individuality inside the team and a more relaxed work habits. Crystal is easier to adopt for a team, but XP produces better results and guarantees a higher productivity.

6.2.1 Crystal Family and its Subgroups

There are several subgroups in the Crystal methodology. These groups, identified with a color, are defined by the number of developers in the team (Table 1).

Table 2 shows project qualities in relation to the size of the development team. On the vertical axis it shows the number of people (from 1 till 1000). The basic idea is that the more people are working on a single project the greater will be the need of coordination among team members. The vertical axis shows the potential damage caused by the hidden defects of the system. In the graph, every square identifies a set of projects that could use the same combination of coordination and politics.

The label of every cell shows the maximum damage and the importance of common coordination in these projects; for example, D40 refers to projects with 20-40 people and a potential loss of the available money.

The different levels show that projects have different priorities; some of them stress the productivity, others the legal responsibility or the costs.
Related Content

Analyzing OSS Project Health with Heterogeneous Data Sources
[www.irma-international.org/article/analyzing-oss-project-health-heterogeneous/68151/](www.irma-international.org/article/analyzing-oss-project-health-heterogeneous/68151/)

Tools Interoperability for Learning Management Systems
[www.irma-international.org/chapter/tools-interoperability-for-learning-management-systems/168978/](www.irma-international.org/chapter/tools-interoperability-for-learning-management-systems/168978/)

Using Open Source Software Components to Implement a Modular Web 2.0 Design for Map-Based Discussions
[www.irma-international.org/article/using-open-source-software-components/51585/](www.irma-international.org/article/using-open-source-software-components/51585/)

Security Management Issues for Open Source ERP in the NGN Environment
[www.irma-international.org/chapter/security-management-issues-open-source/60825/](www.irma-international.org/chapter/security-management-issues-open-source/60825/)

Open Source and Commercial Software Platforms: Is Coexistence A Temporary or Sustainable Outcome?
[www.irma-international.org/chapter/open-source-commercial-software-platforms/52250/](www.irma-international.org/chapter/open-source-commercial-software-platforms/52250/)