IRM PRESS
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.irm-press.com
ITB11777

This chapter appears in the book, *Skills for Managing Rapidly Changing IT Projects*
by Fabrizio Fioravanti © 2006, Idea Group Inc.

Chapter IX

Agile Development

The aim of this chapter is not to compare one Agile methodology against the other but to evidence the common factors that are behind each methodology that can be defined as an Agile one in order to give to the reader a basic understanding needed to better comprehend XP (described in Chapter X) and ADPD (described in Chapter XI and exploited in Chapters XII and XIII).

Agile development and Agile methodologies have not been well considered in the past nor in the present by several people that have published papers (Briand, 2003; Rakitin, 2001) and others in different journals considering this approach too similar to hacking to be considered as a methodology for controlling and developing projects. For some project types, it is true, since safety-critical systems and real-time systems can be difficult to approach with Agile methodologies, as already stated in Chapter VII. On the other hand, several other authors and researchers have tried to consider something more — Agile development (paying particular attention to XP but considering Agile methodologies in a large sense) — and to evidence the strong points behind its approach (Beck, 1999, 2000; Boehm, 2002; Cockburn, 2001, 2001a; Paulk, 2001) and also the interaction of social issues with Agile software development (Cockburn, 1996; Highsmith, 2001).

More and more, people have adhered to an already existent Agile methodology or have developed a personal Agile methodology, and therefore, Agile development has evolved in these last years and has reached the important results of defining a manifesto with the principal guidelines that must be present in all Agile development methodologies.

To this end, the Agile Manifesto (Agile Alliance, 2001) has been published by Agile Alliance. The main assumption stated in that programmatic document is:

"Through this work we have come to value:

- *Individuals and interactions* over process and tools,
- *Working software* over comprehensive documentation,
- *Customer collaboration* over contract negotiation, and
- *Responding to change* over following a plan.

That is, while there is value in the items on the right, we value the items on the *left* more."

It is important to comment on these assumptions in order to discover the important messages that are hidden behind them, evidencing the strong points and also facing the problems of the over simplification that an incorrect understanding of these principles can bring. This analysis will be developed following a schema proposed by Cockburn, where for each statement there are possible actions, the benefits of the actions, the possible side effects, and the results of an overdose. This approach allows one to identify advantages and problems related to the argument under consideration.

Individuals and Interactions Over Process and Tools

This principle suggests that it is important to establish process and to have tools for developing and controlling the project, but it is more valuable to address social issues in a development team. As stated before, and as it will be evidenced in the following, this principle does not state that you must have an unregulated process for software development. Neither does it state that you do not have to adopt tools for measuring or planning the project. It simply means that you have to establish processes and tools that take care of individuals and interactions among individuals, since the benefits that can arise in considering social issues are very important.