

# What You Always Wanted To Know About Agile Methods But Did Not Dare To Ask

Frank Maurer  
University of Calgary  
2500 University Drive NW  
Calgary, Alberta, T2N 1N4, Canada  
+1-403-220-3531  
maurer@cpsc.ucalgary.ca

Grigori Melnik  
University of Calgary  
2500 University Drive NW  
Calgary, Alberta, T2N 1N4, Canada  
+1-403-210-9710  
melnik@cpsc.ucalgary.ca

## ABSTRACT

A fleet of emerging agile methods is both gaining popularity and generating lots of controversy. Real-world examples argue for (e.g. [4]) and against (e.g. [6]) agile methods. Several leading software engineering experts suggest that synthesizing the two (agile with traditional) may provide developers with a comprehensive spectrum of methods (e.g. [1], [2], [5]). This high-level overview tutorial provides background to understand how agile teams are trying to solve modern software development issues.

## Categories and Subject Descriptors

D.2.9 [Software Engineering]: Management – *Life Cycle, Productivity, Programming Teams, Software Process Models.*

## General Terms

Management, Performance, Design, Economics

## Keywords

Agile methods, eXtreme Programming, Scrum, Agile modeling, DSDM, Crystal, Lean

## 1. AGILITY VS. TAYLORISM

The principles of software engineering life cycles formulated by Waterfall methodology and its variants promote Tayloristic principles and strong conformance to a plan through upfront requirements gathering and upfront systems design. They also encourage strict Tayloristic division of labor and the use of role-based teams (of business analysts, system architects, programmers, testers etc.) We argue that these factors plus the reliance on repeatability of the process are the main reason Tayloristic methods are failing in software development [3]. We further provide a detailed comparison of agile methods vs. Tayloristic methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright is held by the author/owner(s).  
ICSE'05, May 15–21, 2005, St. Louis, Missouri, USA.  
ACM 1-58113-963-2/05/0005.

## 2. MAIN AGILE PRACTICES

A brief overview of the main practices of individual methods (including eXtreme Programming, Scrum, Agile Modeling, DSDM, Crystal, Lean Programming) is given. In order to highlight agile methods strengths and limitations, some empirical evidence is presented and analyzed.

## 3. DÉJÀ VU

Participants will likely recognize that most ideas behind agile methods are not new (e.g., software inspections were introduced in 1970s, rapid prototyping – in 1980s). We will show how the conceptual foundation of agile methods and the way individual practices melted together, enhance these “older” practices and exhibit a competitive advantage.

## 4. CROSSING THE CHASM

Participants will discuss what is needed for agile methods to cross the chasm from early adopters to the mainstream of software development. Among other topics, the tutorial addresses issues of knowledge sharing, project management, and social implications of agile methods.

## 5. AUDIENCE

This tutorial is aimed at developers, managers, QA specialists, business customers and academics who are new to agile methods and who would like to acquire basic knowledge of underlying agile values, principles and practices as well as to discuss – from a business perspective – the benefits of and issues with applying agile approaches. No prior experience with agile methods needed.

## 6. SUMMARY

Since all problems are different, all solutions and processes are situational. They depend on the context of the project and on the environment. Agile methods help to succeed in unpredictable environments, which are the reality today. They do it by encouraging continual realignment of development goals with the needs and expectations of the customer. Agile methods concentrate on significantly improving communications and interactions among all stakeholders, promote constant feedback, focus on “clean code that works”, transparency, and merciless testing to achieve higher quality.

## 7. PRESENTERS' PERSONAL PAGES

<http://ebe.cpsc.ucalgary.ca/Frank.Maurer/>

<http://sern.ucalgary.ca/~melnik>

## 8. REFERENCES

- [1] Boehm, B. Get Ready for Agile Methods, with Care. *IEEE Computer*, 35, 1 (Jan. 2002), 64-69.
- [2] Boehm, B. and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, Reading, MA, 2004.
- [3] *Chaos Report*. The Standish Group, West Yarmouth, MA, 1995, 1997, 1999, 2001, 2003.
- [4] Highsmith, J. and Cockburn, A. Agile Software Development: The Business of Innovation. *IEEE Computer*, 34, 9 (Sep. 2001): 120-122.
- [5] Humphrey, W. Comments on eXtreme Programming. IEEE Computer Society Dynabook. Online, retrieved Jan 23, 2005 from <http://computer.org/software/dynabook/HumphreyCom.htm>
- [6] Rakitin, S. Letters to the Editor. *IEEE Computer*, 34, 12 (Dec. 2001): 4.