

HOCHSCHULE DARMSTADT  
UNIVERSITY OF APPLIED SCIENCES

**h\_da**

# Agile Software Development

Part 1: Introduction. Manifesto for Agile Software  
Development

Prof. Dr. A. del Pino

# Introduction

## Agile FAQ

- Is *agile software development* the same as *hacking* ?
- Isn't *agile software development* just another term for *pair programming* ?
- Is it true that agile approaches are better than more traditional ones ?
- Someone said that agile development approaches are good for small projects, but not for large ones. Is this true ?
- Is it true that agile software development works only in small- and mid-sized organizations ?

# Introduction

## What is *agile* software development ?

**agil** <lat.> (flink, wendig, beweglich); **Agi|li|tät**, die; -

Source: Duden Band 1. *Die deutsche Rechtschreibung*. 24. Auflage, Dudenverlag, 2006

**agil** *Adj* »beweglich« *erw. fremd.* (17. Jh.). Entlehnt aus frz. *agile* (mit Relatinisierung) oder unmittelbar aus l. *agilis* gleicher Bedeutung. Dieses zu l. *agere* »treiben, betreiben«.

Source: F. Kluge, E. Seebold. *Ethymologisches Wörterbuch der deutschen Sprache*. 24. Auflage, Walter de Gruyter Verlag, 2002

# Introduction

## What is *agile* software development ?

Extreme Programming (XP) is about social change.

Source: K. Beck. *Extreme Programming Explained*. 2nd edition, Addison-Wesley, 2005

[...] agility is not a technology, science, or product but a culture.

Source: P. Kruchten. *Voyage in the Agile Memplex*. ACM Queue, Vol. 5, No. 5, July/August 2007

Adaptive Software Development is ultimately about rethinking how to manage in the turbulent times we have brought upon ourselves.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. xxxi

# Introduction

## What is *agile* software development ?

Scrum represents a new, more accurate way of doing software development that is based on the assumption that software is a new product every time that it is written or composed.

Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice-Hall, 2002, pg. Viii

Core to agile software development is the use of light-but-sufficient rules of project behavior and the use of human- and communication-oriented rules.

Source: A. Cockburn. *Agile Software Development*. Addison-Wesley, 2002, pg. xxii

# Manifesto for Agile Software Development

## The Agile Alliance

Seventeen experts on lightweight development processes met in February 2001 in Snowbird, Utah to discuss what they might have in common, and where they disagree.

*Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*

This group nicknamed itself the [Agile Alliance](#), and the [Agile Software Development Manifesto](#) is a result of this meeting.

We recommend agile approaches to software development because they deliver value to organizations and end users faster and with higher quality.

Source: [www.agilealliance.org](http://www.agilealliance.org)

# Manifesto for Agile Software Development

## The Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

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.


Source: [www.agilemanifesto.org](http://www.agilemanifesto.org)

# Manifesto for Agile Software Development

## 1st Principle

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?

# Manifesto for Agile Software Development

## 2nd Principle

Welcome changing requirements, even late in development.  
Agile processes harness change for the customer's  
competitive advantage.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?

# Manifesto for Agile Software Development

## 3rd Principle

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?


Which difficulties might appear ?

# Manifesto for Agile Software Development

## 4th Principle

Business people and developers must work together daily throughout the project.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?


Which difficulties might appear ?

# Manifesto for Agile Software Development

## 5th Principle

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?

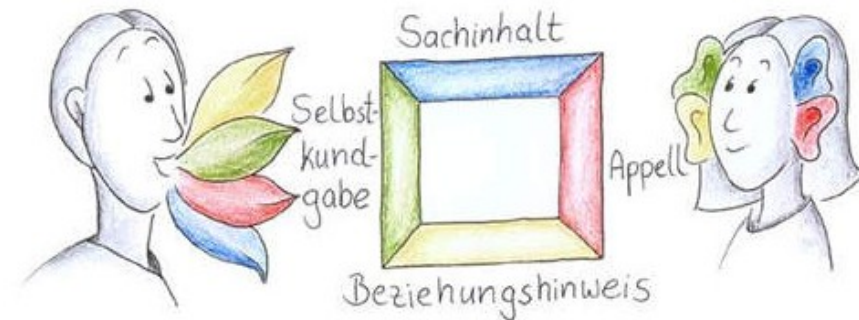
# Manifesto for Agile Software Development

## 6th Principle

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Source: <http://www.agilemanifesto.org/principles.html>

- ❓ Which benefits ?
- How to achieve that ?
- Which difficulties might appear ?




Source: F. Schultz von Thun, 1981

# Manifesto for Agile Software Development

## 7th Principle

Working software is the primary measure of progress.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?

# Manifesto for Agile Software Development

## 8th Principle

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?


Which difficulties might appear ?

# Manifesto for Agile Software Development

## 9th Principle

Continuous attention to technical excellence and good design enhances agility.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?

# Manifesto for Agile Software Development

## 10th Principle

Simplicity - the art of maximizing the amount of work not done - is essential.

Source: <http://www.agilemanifesto.org/principles.html>



*B. Pascal (1656):*

*This letter is a very long one, simply because I had no time to make it shorter.*

Source: Blaise Pascal, Lettres Provinciales, Letter XVI, December 4, 1656

❓ Which benefits ?

How to achieve that ?


Which difficulties might appear ?

# Manifesto for Agile Software Development

## 11th Principle

The best architectures, requirements, and designs emerge from self-organizing teams.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?

# Manifesto for Agile Software Development

## 12th Principle

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: <http://www.agilemanifesto.org/principles.html>

 Which benefits ?

How to achieve that ?

Which difficulties might appear ?