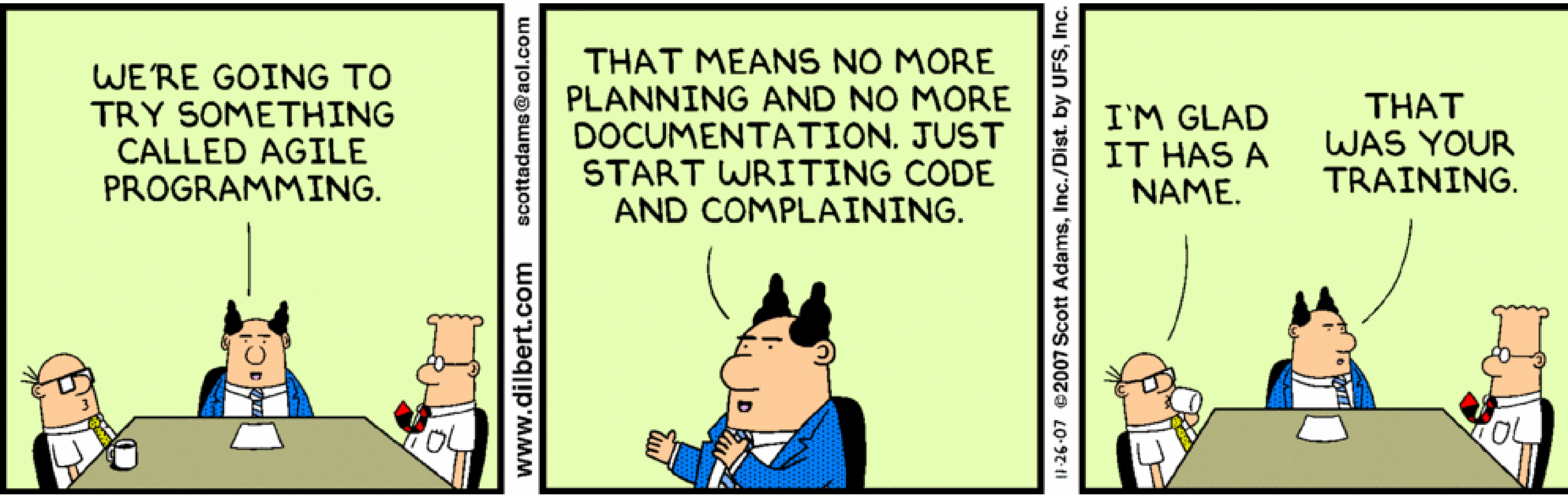
CONSULTING.TECHNOLOGY.OUTSOURCING

What does it mean to be Agile

Marek Majchrzak, Andrzej Bednarz

Wrocław, 11.10.2011

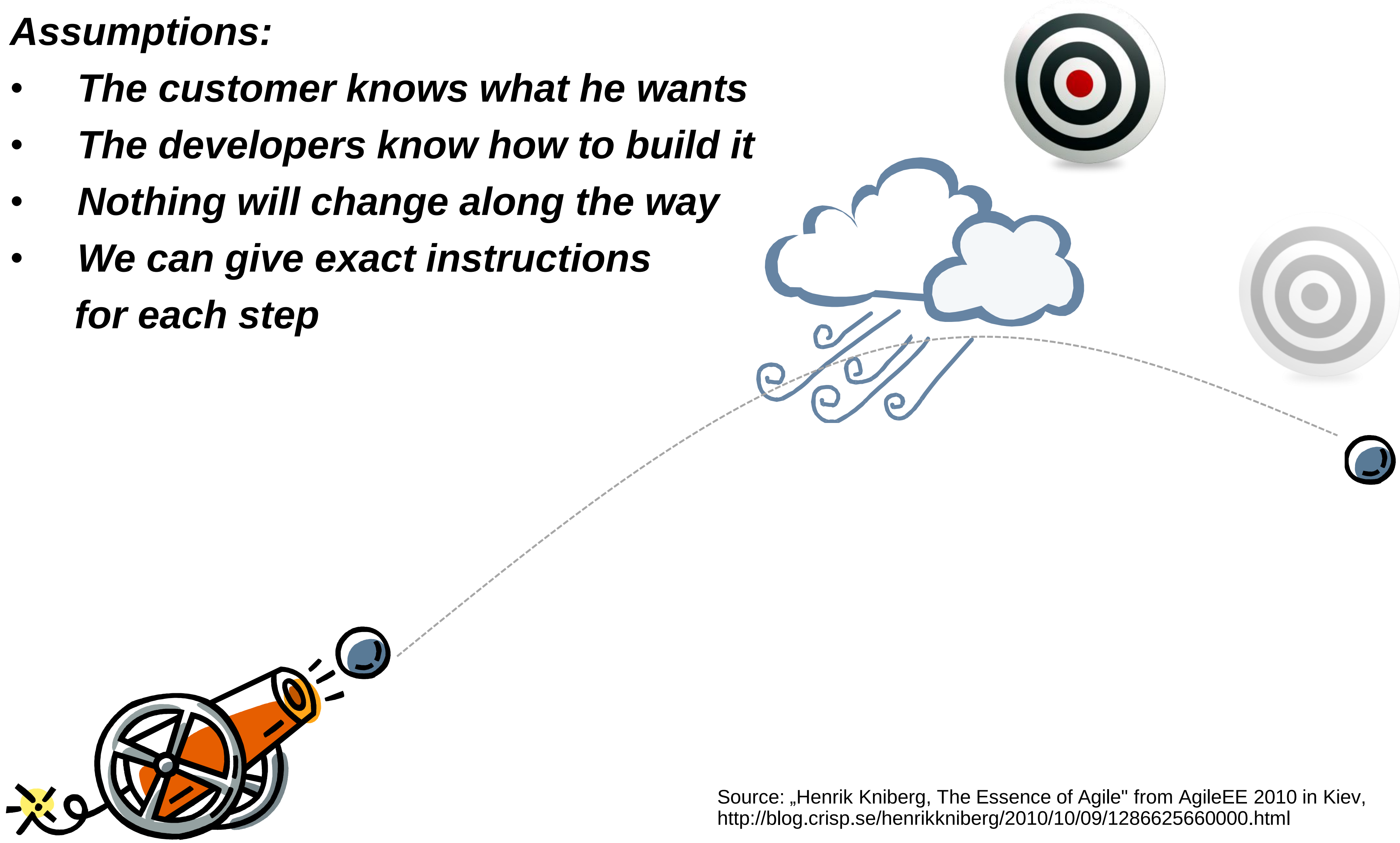
GEMEINSAM. ENERGIEN FREISETZEN.


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
scottadams@aol.com
www.dilbert.com
11-26-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.

Traditional methods

Assumptions:

- ***The customer knows what he wants***
- ***The developers know how to build it***
- ***Nothing will change along the way***
- ***We can give exact instructions for each step***

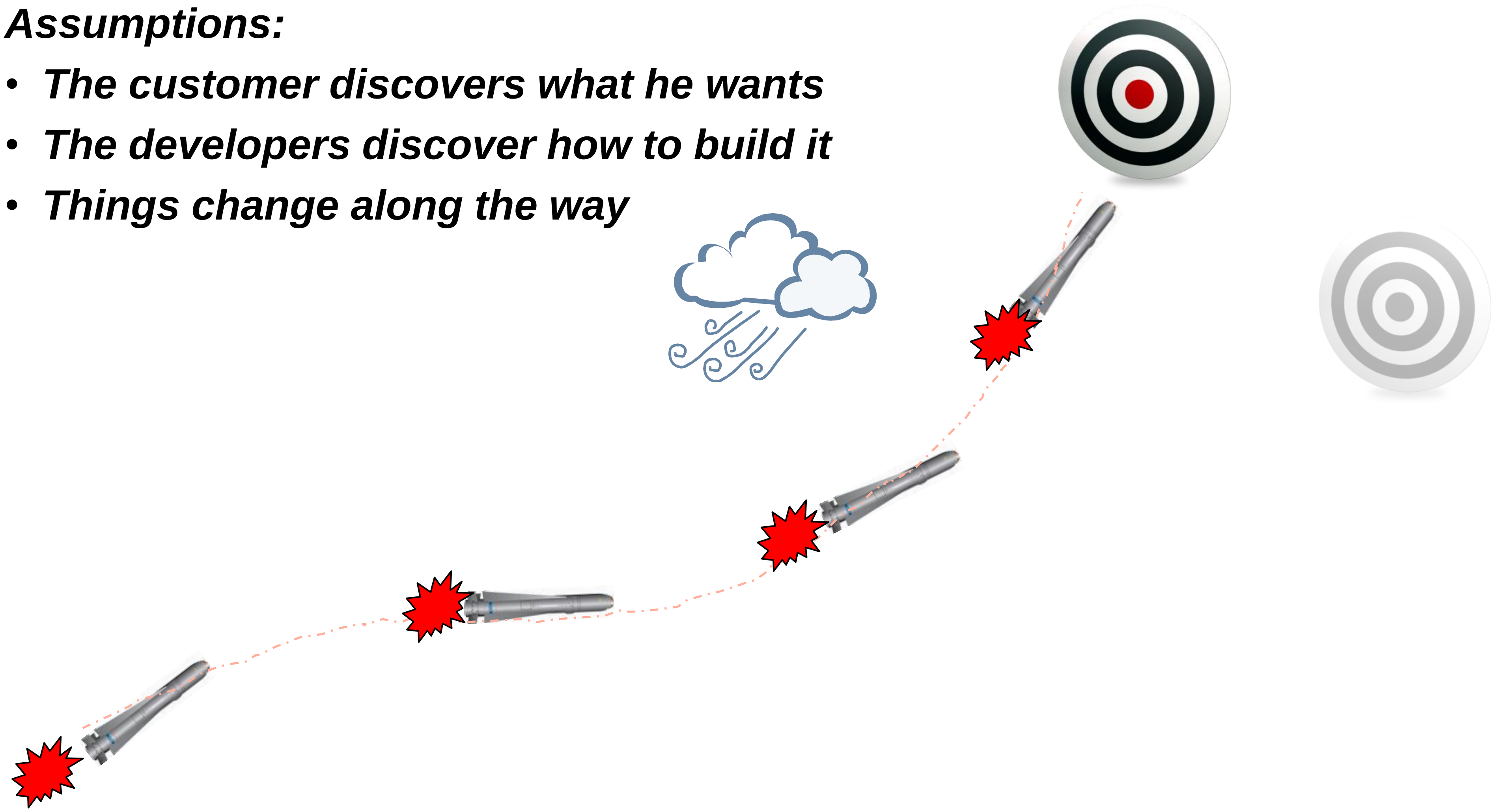


Source: „Henrik Kniberg, The Essence of Agile" from AgileEE 2010 in Kiev, http://blog.crisp.se/henrikkniberg/2010/10/09/1286625660000.html

Agile methods

Assumptions:

- ***The customer discovers what he wants***
- ***The developers discover how to build it***
- ***Things change along the way***

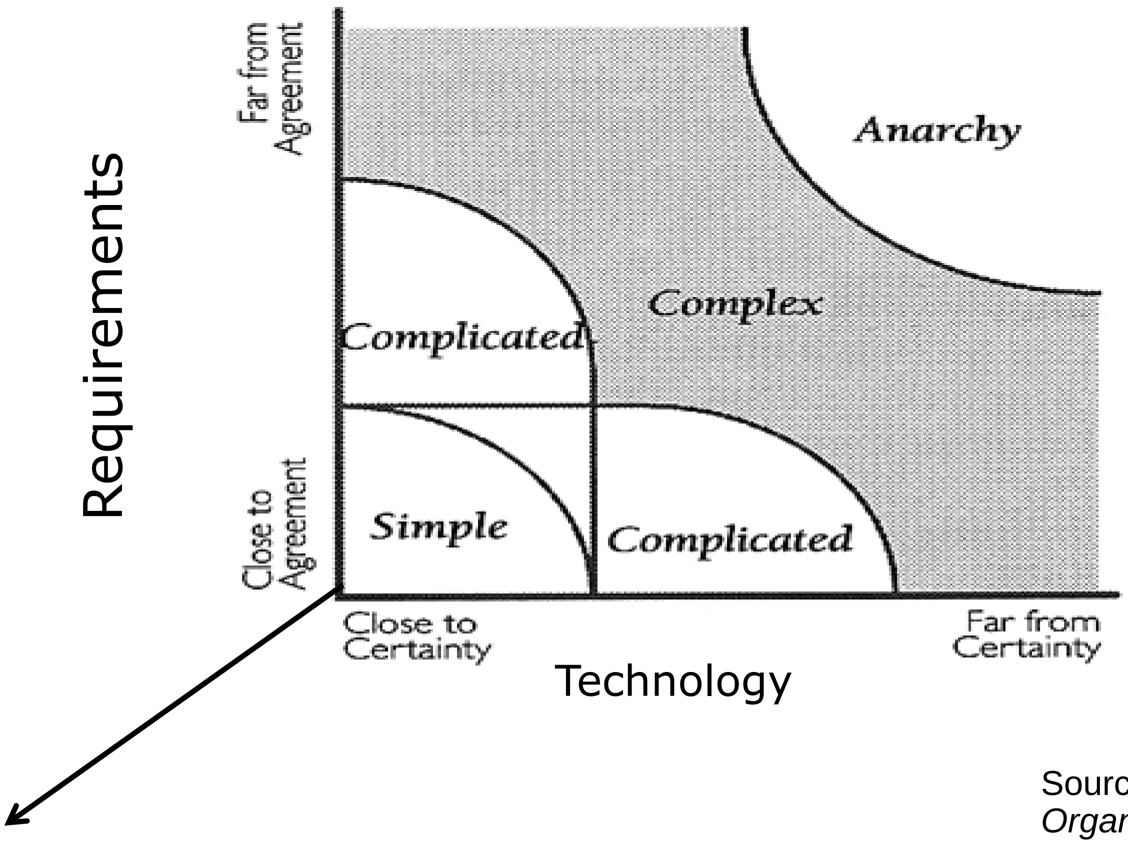


Source: „Henrik Kniberg, The Essence of Agile" from AgileEE 2010 in Kiev, http://blog.crisp.se/henrikkniberg/2010/10/09/1286625660000.html

Defined process vs. Empirical process

Complex processes require an empirical control model.

An empirical control model entails frequent inspection and adaptive response.



People
add another level of complexity

Source: *Strategic Management and Organizational Dynamics* by Ralph Stacey in *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Defined vs. Empirical process

- ***Defined processes***
  - We know all premises
  - We can give exact instructions for each action
  - May be complicated, but ultimately knowable

- ***Empirical processes***
  - Environment and prerequisites are not defined completely
  - Requirements change over time
  - The knowledge about the best approach is incomplete
  - The system is complex, i.e. not simple and never fully knowable

Source: http://www.swissict.ch/fileadmin/sekretariat/AG_FG/Lean_Agile_Scrum/Simon_und_Krishan_Scrum_101.pdf

Environment for Empirical Process Control

- ***Solving complex problems needs constant adaption and requires:***
  - Creativity
  - Initiative
  - Individuals and teams that learn

- ***So that this can flourish, a culture is needed which values:***
  - Trust: not trying to place blame when errors occur and appreciating the learning opportunities
  - Respect: people are not resources

- ***These are codified in the Agile Manifesto and the accompanying principles:***
  - Written in 2001 by 17 prominent figures in the field of software development

Source: http://www.swissict.ch/fileadmin/sekretariat/AG_FG/Lean_Agile_Scrum/Simon_und_Krishan_Scrum_101.pdf

At the heart of Scrum – Agile Manifesto

The Agile Manifesto states

Individuals and interactions	over	Processes and tools
Working software	over	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."

Principles behind the Agile Manifesto

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

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

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

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

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

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

• Working software is the primary measure of progress.

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

• Continuous attention to technical excellence and good design enhances agility.

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

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

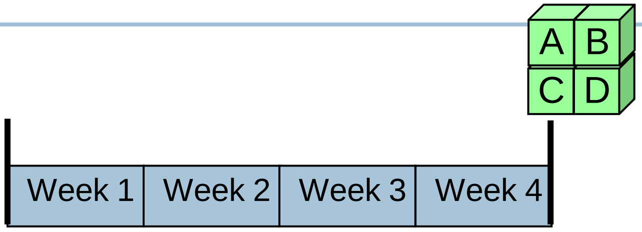
• At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: http://blog.crisp.se/henrikkniberg/

Iterative and incremental

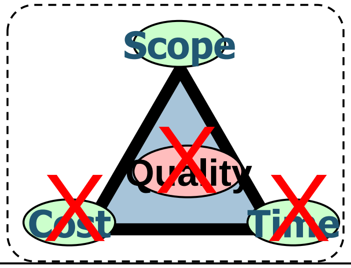
Plan

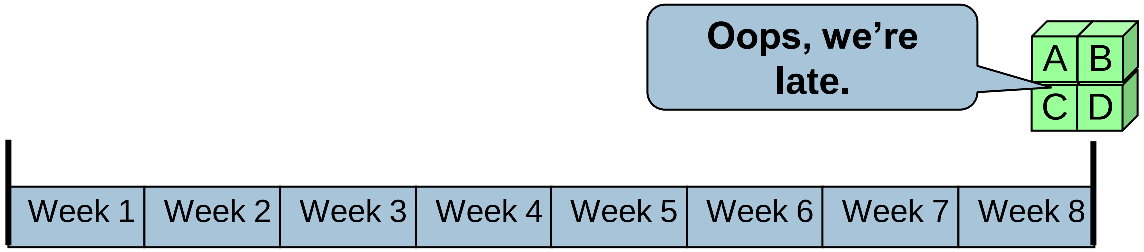
(doomed to fail, but we don't know it yet)



Traditional scenario

”We will deliver ABCD in 4 weeks”



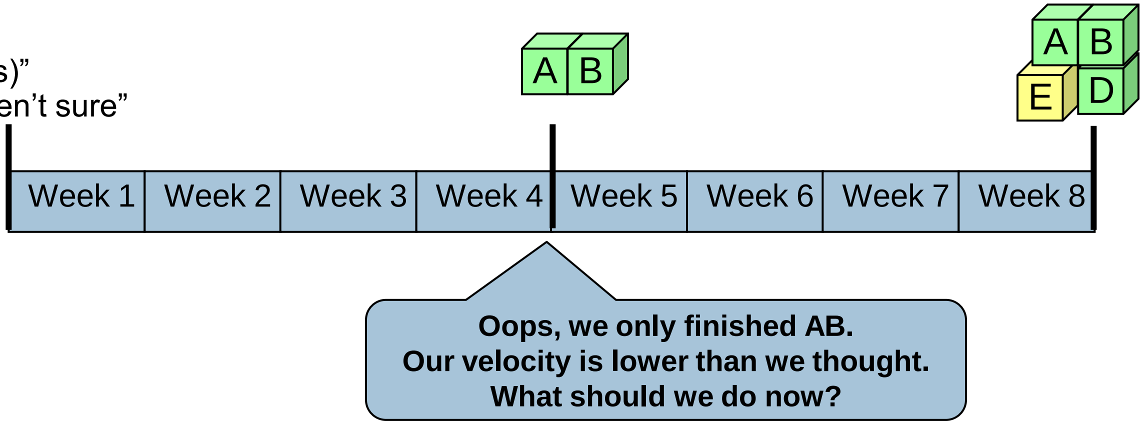


Agile scenario

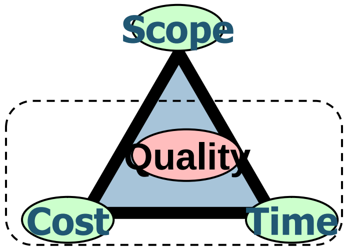
”We always deliver something every sprint (4 weeks)”

”We *think* we can finish ABCD in 1 sprint, but we aren't sure”

”We always deliver the most important items first”



A B
A B
E D
Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 Week 8
Oops, we only finished AB.
Our velocity is lower than we thought.
What should we do now?



Source: Henrik Kniberg

Agile was specifically designed to deal with

Ziv's law - specifications will never be fully understood.

Humphrey's law - the user will never know what they want until after the system is in production (maybe not even then)

Wegner's lemma - an interactive system can never be fully specified nor can it ever be fully tested. This is the software analogy to Godel's theorem.

Langdon's lemma - software evolves more rapidly as it approaches chaotic regions (taking care not to spill over into chaos)

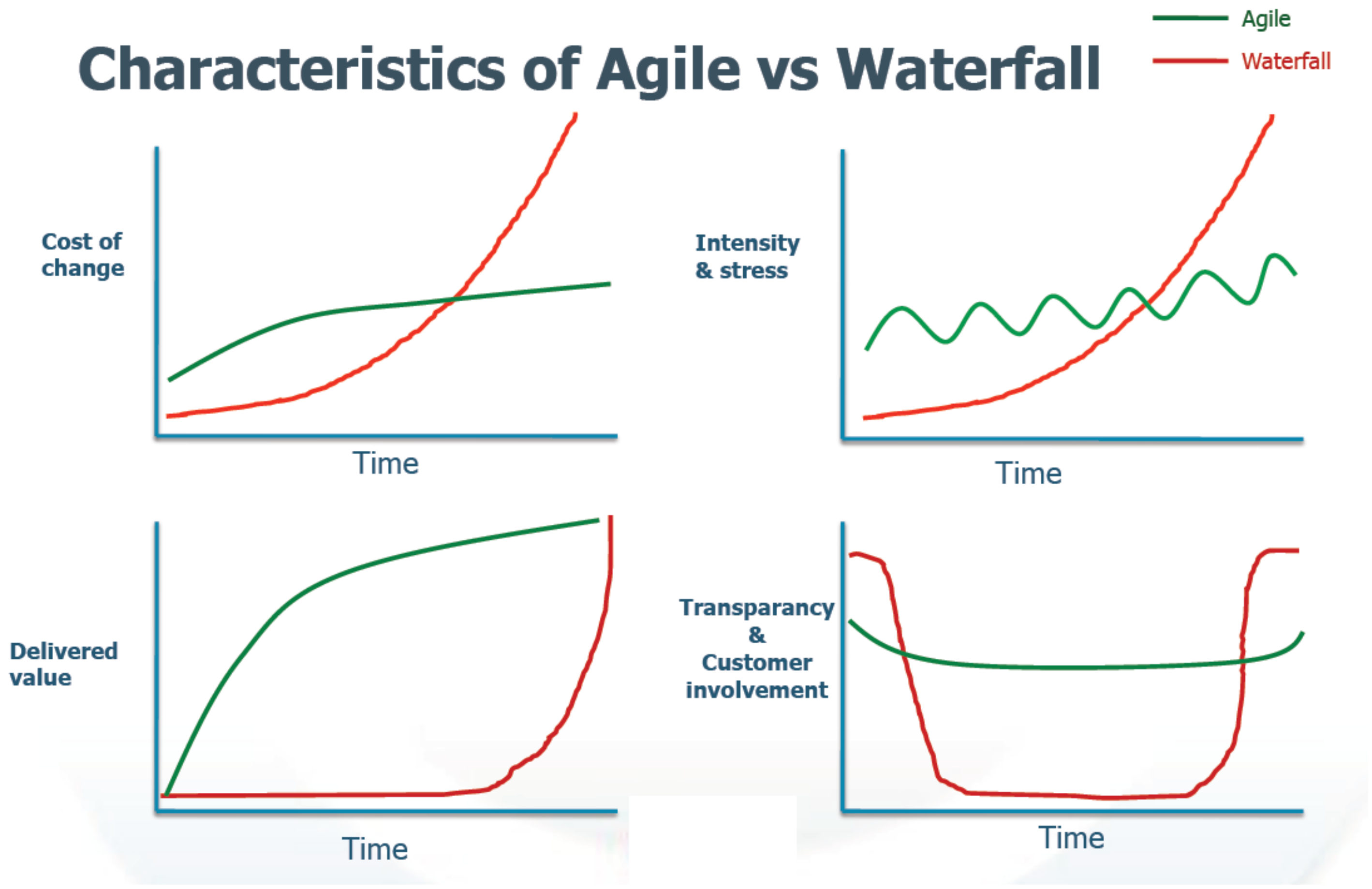
Source: Jeff Sutherland, http://scrum.jeffsutherland.com/2007/07/origins-of-scrum.html

Agile transparency

Agile development will not solve your problems - it will make them so painfully visible that ignoring them is harder

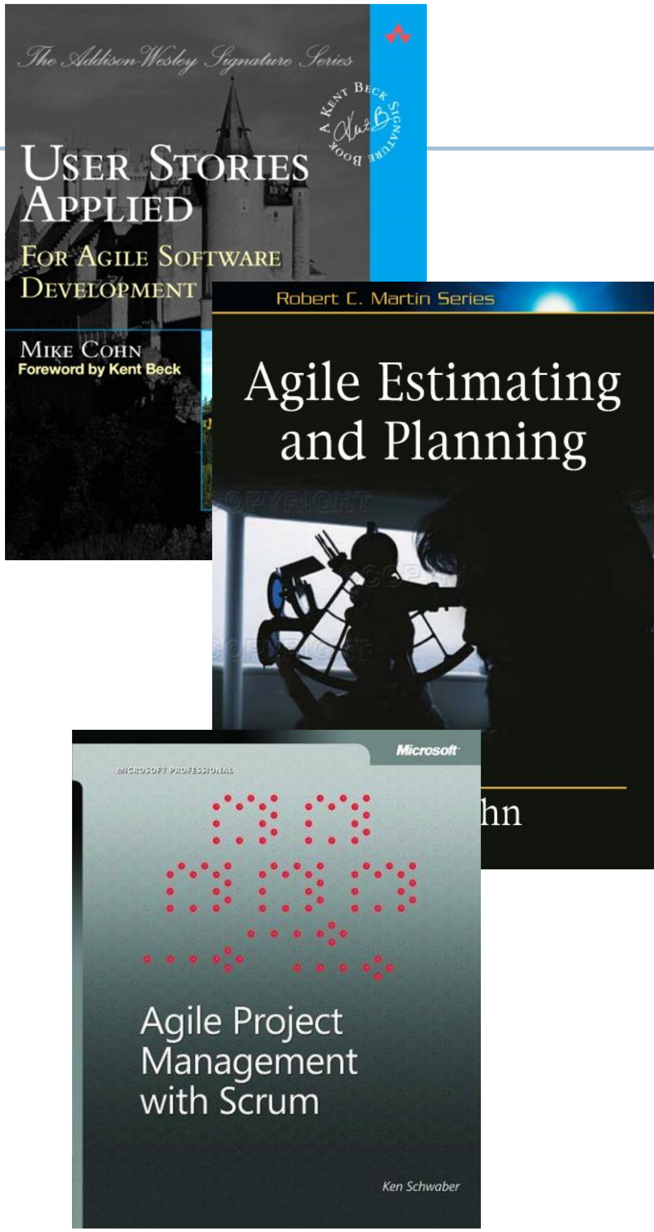
Ken Schwaber

Characteristics of Agile vs Waterfall



Reading list

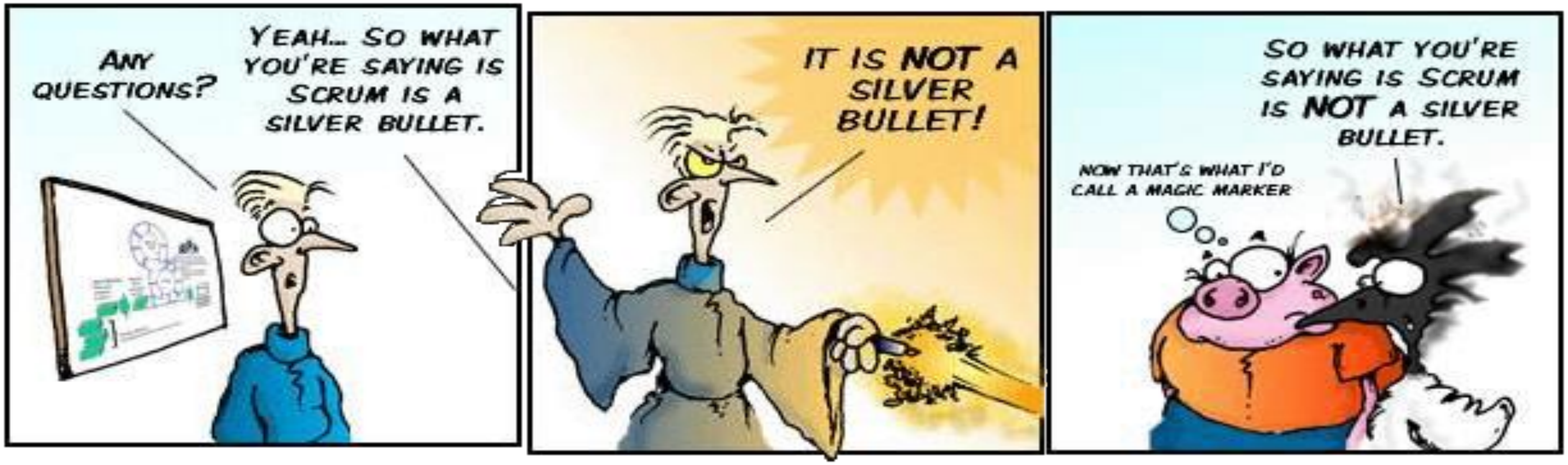
- ***Agile and Iterative Development: A Manager's Guide by Craig Larman***
- ***Agile Estimating and Planning by Mike Cohn***
- ***Agile Project Management with Scrum by Ken Schwaber***
- ***Agile Retrospectives by Esther Derby and Diana Larsen***
- ***Agile Software Development Ecosystems by Jim Highsmith***
- ***Agile Software Development with Scrum by Ken Schwaber and Mike Beedle***
- ***Scrum and The Enterprise by Ken Schwaber***
- ***User Stories Applied for Agile Software Development by Mike Cohn***
- ***Lots of weekly articles at www.scrumalliance.org***


The Addison-Wesley Signature Series
A Kent Beck Signature Book
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck
Robert C. Martin Series
Agile Estimating and Planning
Microsoft
Microsoft Professional
Agile Project Management with Scrum
Ken Schwaber

Sources and references

1. *Henrik Kniberg, "Scrum and XP from the Trenches"*
2. *Henrik Kniberg, „The Essence of Agile" from AgileEE 2010 in Kiev, http://blog.crisp.se/henrikkniberg/2010/10/09/1286625660000.html*
3. *Agile with Scrum, Wrocław Agile Community*
4. *Christoph Mathis, Simon Roberts, Scrum 101, ScrumCenter.com*

Q&A


ANY QUESTIONS?
YEAH... SO WHAT YOU'RE SAYING IS SCRUM IS A SILVER BULLET.
IT IS NOT A SILVER BULLET!
NOW THAT'S WHAT I'D CALL A MAGIC MARKER
SO WHAT YOU'RE SAYING IS SCRUM IS NOT A SILVER BULLET.

Tony D. Clark, © 2006 implementingscrum.com

Vielen Dank für Ihre Aufmerksamkeit!

www.capgemini.com