Teaching Agile Software Development

Martin Kropp, FHNW

Andreas Meier, ZHAW

Agenda

- Motivation

- Pyramid of Agile Competences

- Agile Software Engineering Course

- Evaluation

n|w University of Applied Sciences Northwestern Switzerland

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Motivation

- "Agile" has become mainstream
  - Swiss Agile Study 2012 (SAS)
    - 140 IT companies
    - 194 IT Professionals

- Identified as one important innovation
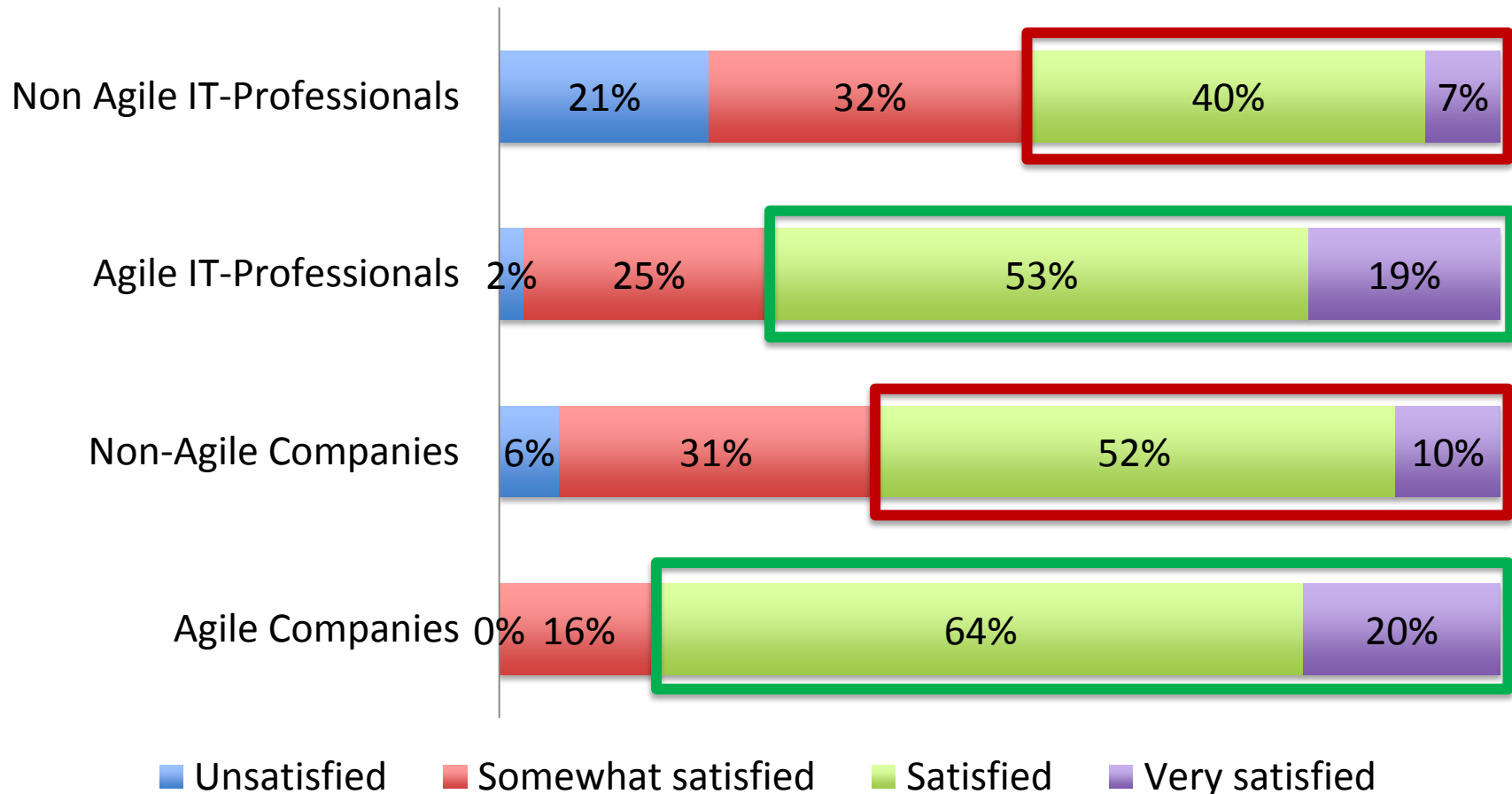
- Two sides of the medal...

Company Survey

Non-Agile 43%

Agile 57%

www.swissagilestudy.ch

University of Applied Sciences
Northwestern Switzerland

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

School of
Engineering

InIT Institut für angewandte
Informationstechnologie
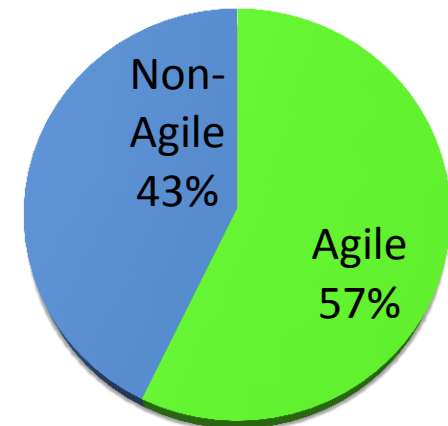
The Good Sides: Satisfaction

How satisfied are you with your current methodology?

Category	Unsatisfied	Somewhat satisfied	Satisfied	Very satisfied
Non Agile IT-Professionals	21%	32%	40%	7%
Agile IT-Professionals	2%	25%	53%	19%
Non-Agile Companies	6%	31%	52%	10%
Agile Companies	0%	16%	64%	20%

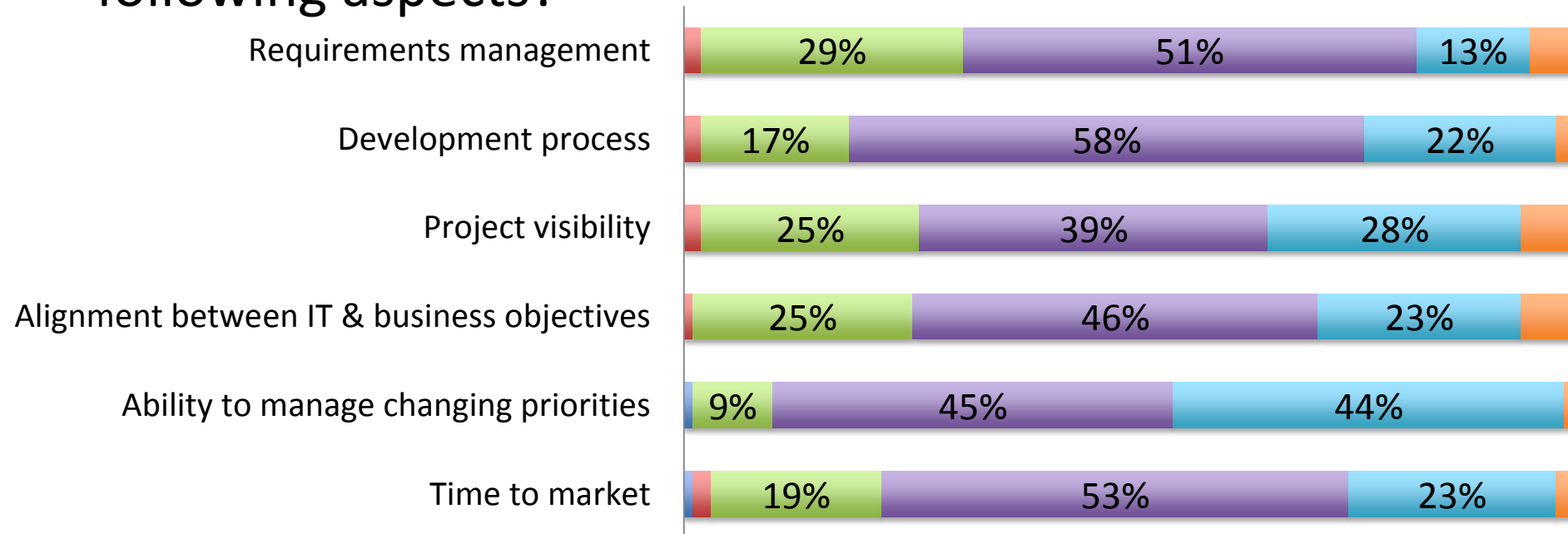Legend: Unsatisfied · Somewhat satisfied · Satisfied · Very satisfied

Motivation

SAS shows very promising results:

- much higher satisfaction with agile methodologies than with plan-driven ones

The Good Sides: Agile Influence

How has agile software development influenced the following aspects?

Aspect	Unchanged	Improved	Significantly improved
Requirements management	29%	51%	13%
Development process	17%	58%	22%
Project visibility	25%	39%	28%
Alignment between IT & business objectives	25%	46%	23%
Ability to manage changing priorities	9%	45%	44%
Time to market	19%	53%	23%

Legend: ■ Much worse ■ Worse ■ Unchanged ■ Improved ■ Significantly improved ■ Don't know

Motivation

SAS shows very promising results:

- significant improvement in the ability to manage changing priorities

- improvement of the development process in general

- much faster time-to-market

University of Applied Sciences
Northwestern Switzerland

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

The Dark Sides: Agile Influence

How has agile software development influenced the following aspects?

Aspect	Unchanged	Improved	Significantly improved
Engineering discipline	42%	42%	9%
Development cost	52%	22%	7%
Software maintainability / extensibility capability	55%	23%	12%
Software quality	45%	35%	16%
Productivity	33%	47%	15%

■ Much worse ■ Worse ■ Unchanged ■ Improved ■ Significantly improved ■ Don't know

Motivation

SAS shows very promising results at first view, there are also disappointing findings

- Development cost

- Software quality

- Software maintainability

have not really improved as much as expected

Motivation

Pros:

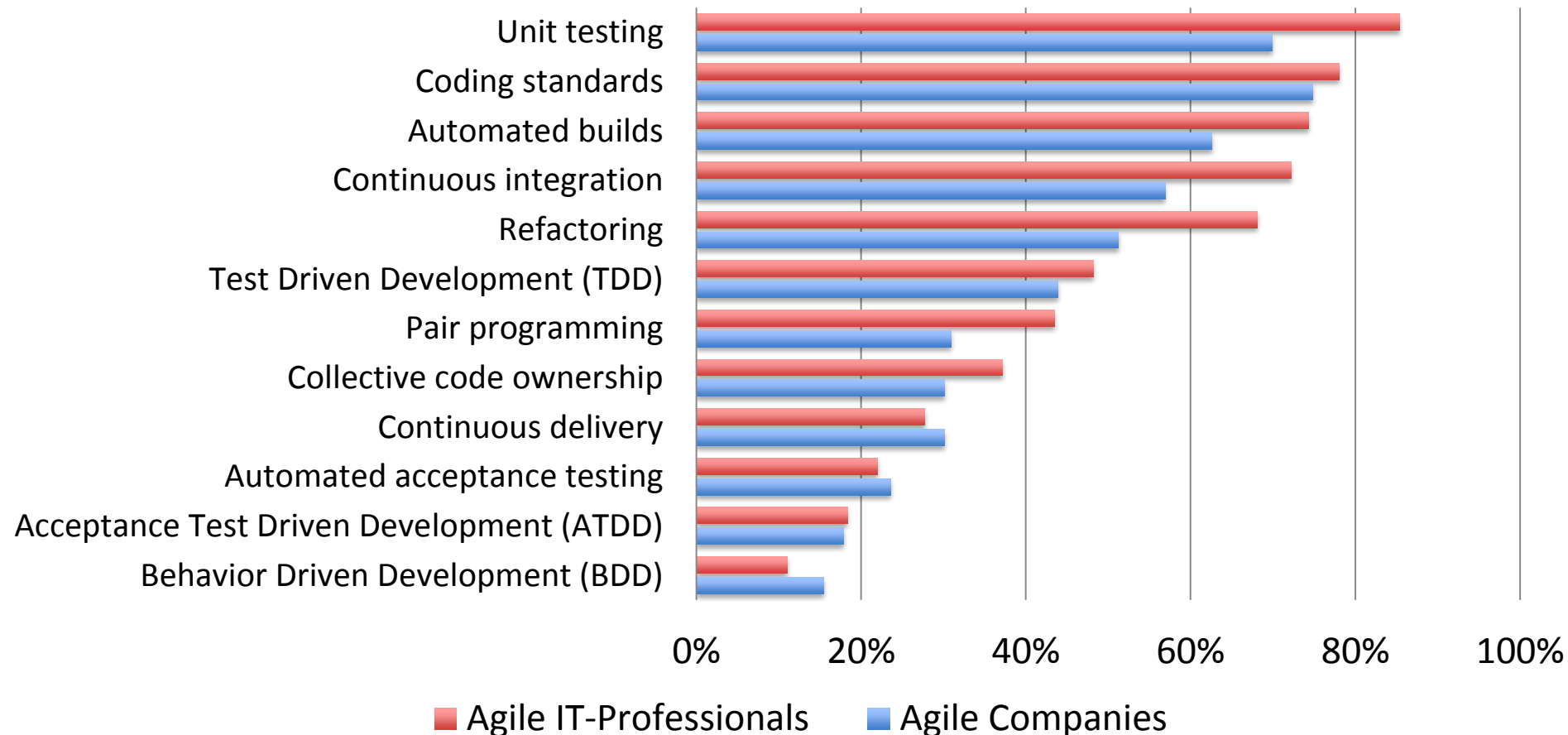- Major improvements in some project management aspects

Cons:

- Minor or no improvements in financial, technical or quality aspects
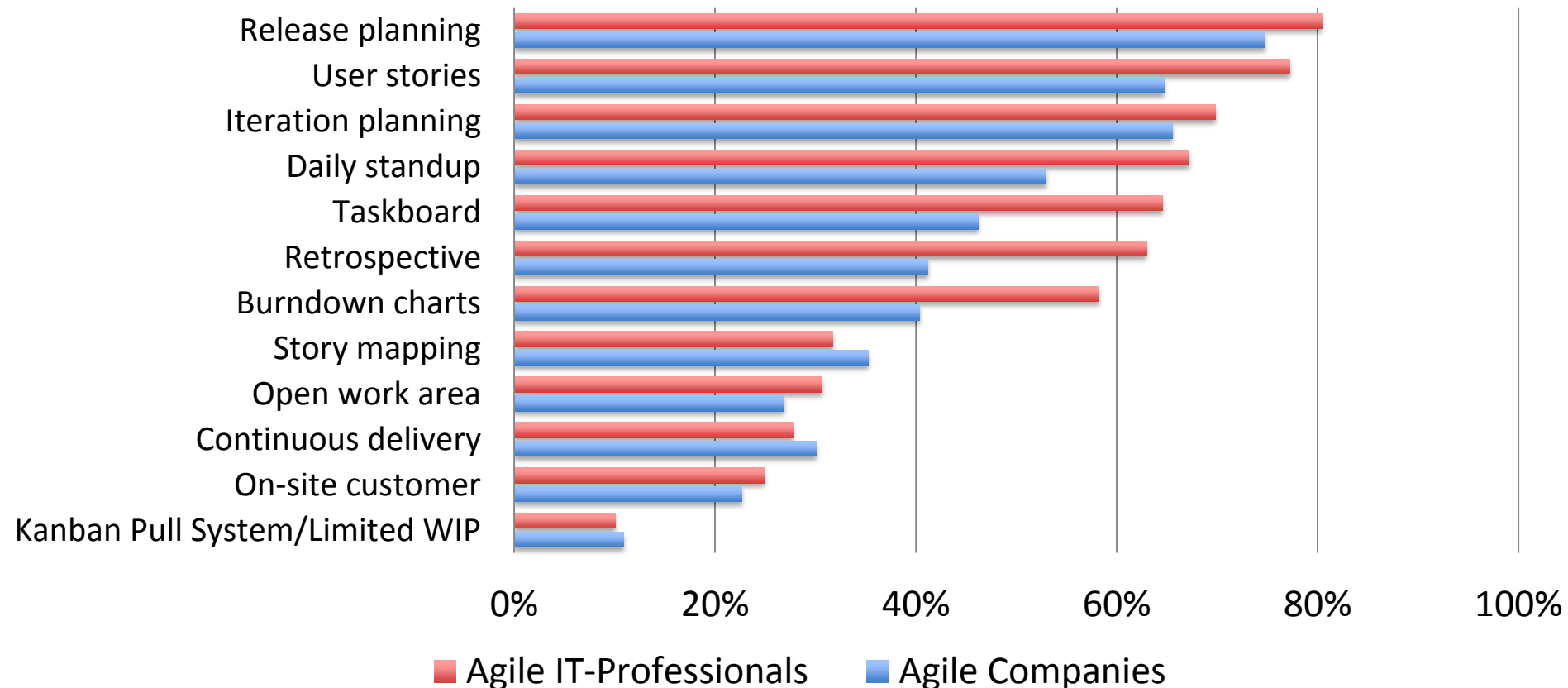
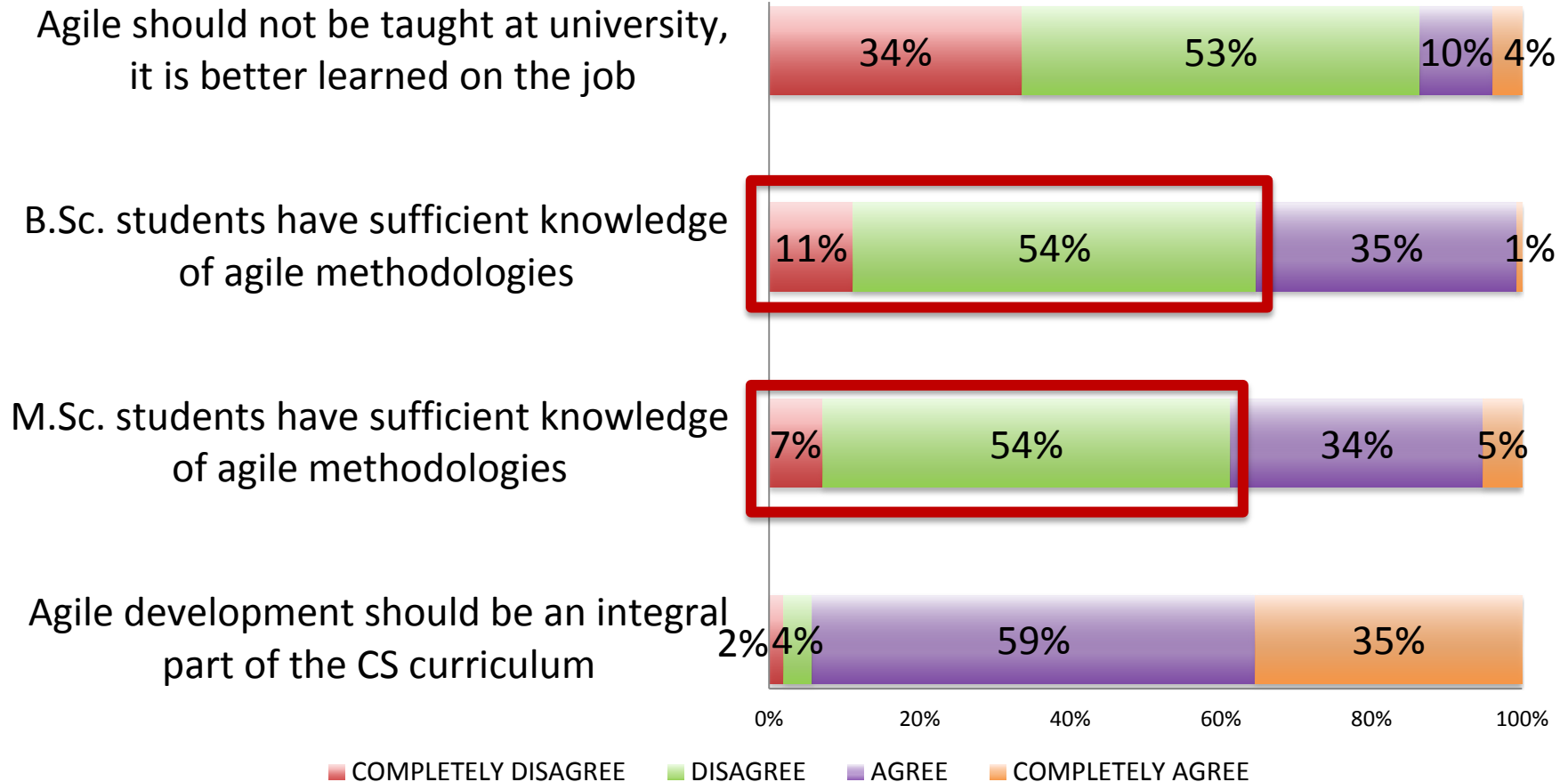## Reasons?

How Agile is Applied

- ## Engineering Practices

Chart: "Engineering Practices" horizontal bar chart comparing Agile IT-Professionals (red) and Agile Companies (blue).

Practice	Agile IT-Professionals	Agile Companies
Unit testing	~85%	~70%
Coding standards	~78%	~75%
Automated builds	~73%	~62%
Continuous integration	~72%	~57%
Refactoring	~67%	~50%
Test Driven Development (TDD)	~48%	~43%
Pair programming	~43%	~31%
Collective code ownership	~37%	~30%
Continuous delivery	~28%	~30%
Automated acceptance testing	~22%	~24%
Acceptance Test Driven Development (ATDD)	~19%	~18%
Behavior Driven Development (BDD)	~11%	~16%

Legend: ■ Agile IT-Professionals ■ Agile Companies

How Agile is Applied

- Managing Practices

What the Industries says

Agile should not be taught at university, it is better learned on the job
- COMPLETELY DISAGREE 34%
- DISAGREE 53%
- AGREE 10%
- COMPLETELY AGREE 4%

B.Sc. students have sufficient knowledge of agile methodologies
- COMPLETELY DISAGREE 11%
- DISAGREE 54%
- AGREE 35%
- COMPLETELY AGREE 1%

M.Sc. students have sufficient knowledge of agile methodologies
- COMPLETELY DISAGREE 7%
- DISAGREE 54%
- AGREE 34%
- COMPLETELY AGREE 5%

Agile development should be an integral part of the CS curriculum
- COMPLETELY DISAGREE 2%
- DISAGREE 4%
- AGREE 59%
- COMPLETELY AGREE 35%

Legend: COMPLETELY DISAGREE | DISAGREE | AGREE | COMPLETELY AGREE

and Agile Education …

- What does this mean for teaching software engineering?

- Do we provide the right courses?

- Which skills and competences does an agile worker need?

Pyramid of Agile Competences

- Technical skills or *engineering practices*,
  i.e. programming, unit testing, clean code, test-driven development, collective code ownership etc.

- Engineering practices are mostly competences that refer to the single individual

- Software Craftsmanship

- builds the foundation of the pyramid

Engineering Practices
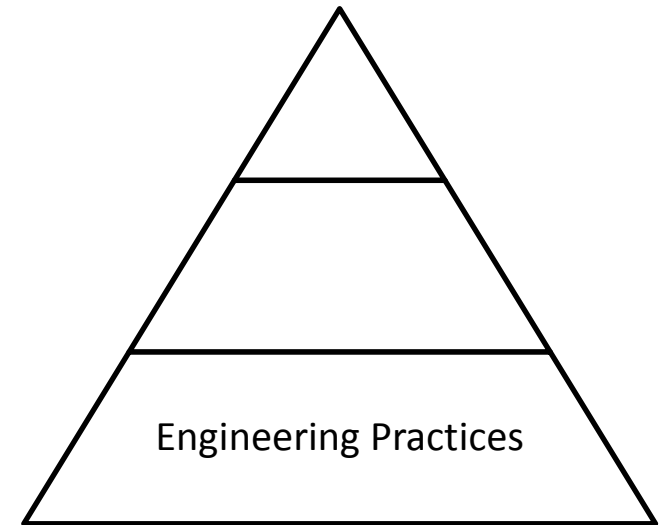
Pyramid of Agile Competences

- Agile *management practices* define how agile projects are organized and run

- I.e. iterative planning, short release cycles, small releases, strong customer involvement and highly interactive teams

- Management practices are typically team aspects, which require appropriate social competences

Management Practices

Engineering Practices
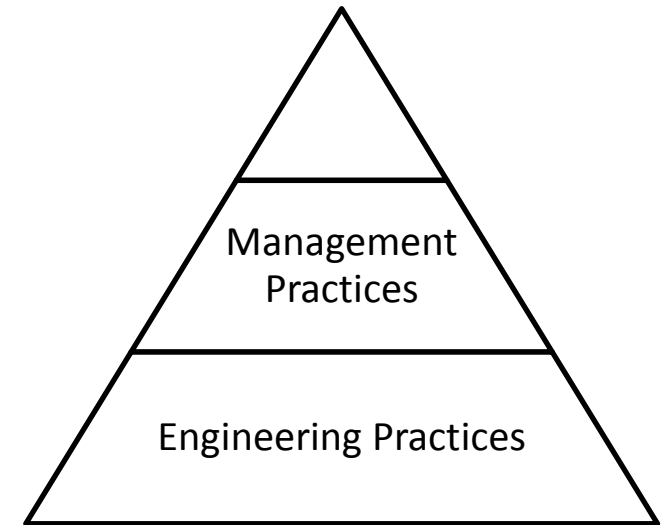
Pyramid of Agile Competences

- On top of these competences come the *agile values*, which are articulated in the *Agile Manifesto*
  i.e. behavior like mutual respect, openness, and courage
- Difficult to teach

Agile Values

Management Practices

Engineering Practices
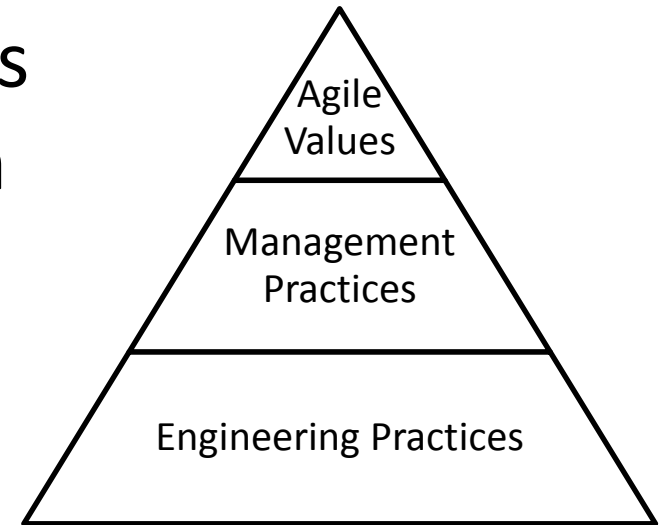
Agile Education Concept

- All three levels must be considered

- Bachelor and Master level

- Appropriate teaching methods should be applied
  - courses, simulations, case studies, group work

Approaches

- Horizontal Teaching
  - Separate modules for each level

- Vertical Teaching
  - Integrate several levels into one module
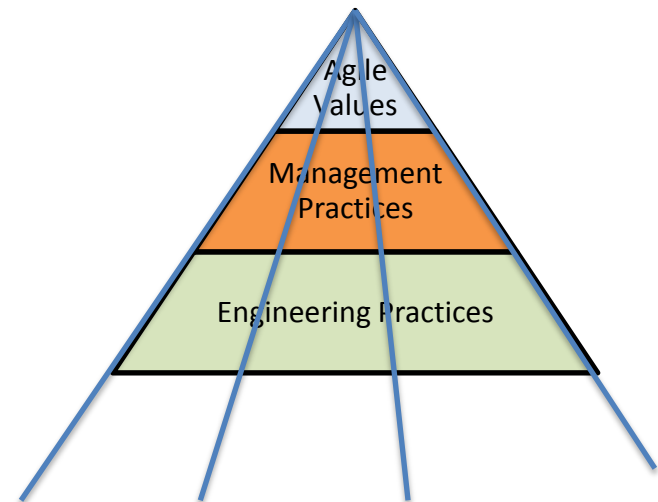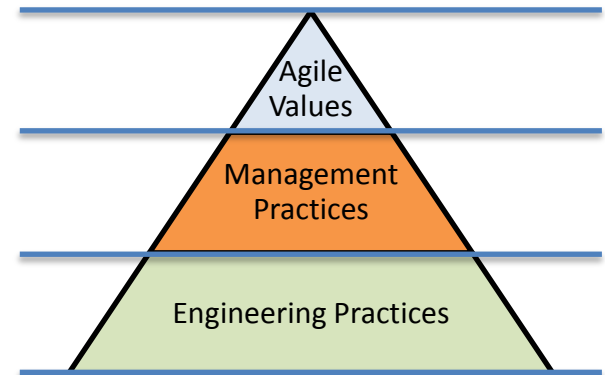
Agile Software Engineering Course

- 16-week semester class in the last year of the undergraduate level (B.Sc.)

- The students completed one Java programming project in an agile team of six to eight members during the course of the semester

- Per week there were a 2 hours lecture with the whole class and a 2 hours programming workshop with half the class

- 27 students were enrolled

Agile Software Engineering Course

- Distribution of lectures, workshops and self-study:

Lectures	32 h
Workshops	32 h
Self-study	56 h
Total	120 h

Agile Software Engineering Course

W	Lecture	Workshop
1	eXtreme Programming	Installation IDE and Plug-Ins Coding Assessment 1
2	eXtreme Programming Version Control	Coding Assessment 2 Version Control System (SVN)
3	eXtreme Programming Project Automation	Build Scripts (Ant)
4	Continuous Integration	CI (Jenkins Build Server)
5	Unit Testing	JUnit
6	Unit Testing / Mock Objects Clean Code / Code Smells	JUnit EasyMock
7	Refactoring	Refactoring
8	Introduction to Test-Driven Design / Scrum	TDD, The Craftsman articles
9	Scrum	Agile Game Development
10	Scrum	Agile Game Development
11	Agile Estimating and Planning	Agile Game Development Planning Poker
12	Metrics Agile Teams	Agile Game Development Metrics (EMMA)
13	User Stories Agile Principles	Agile Game Development
14	Demonstration of computer games	Agile Game Development

Agile
Values

Management
Practices

Engineering
Practices

Agile Game Development

M. Kropp / A. Meier
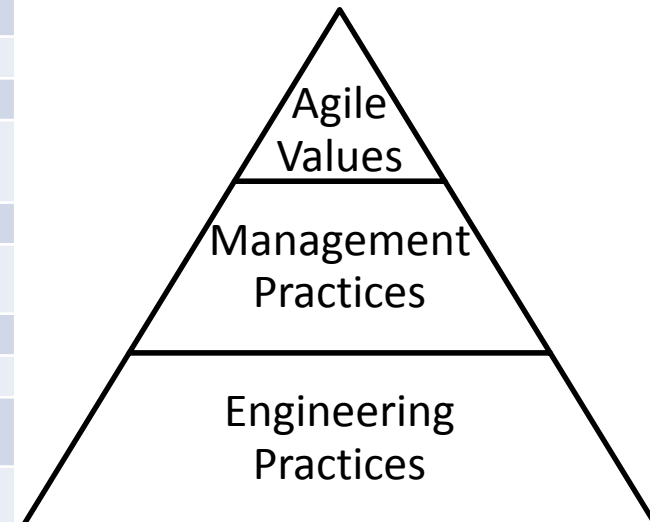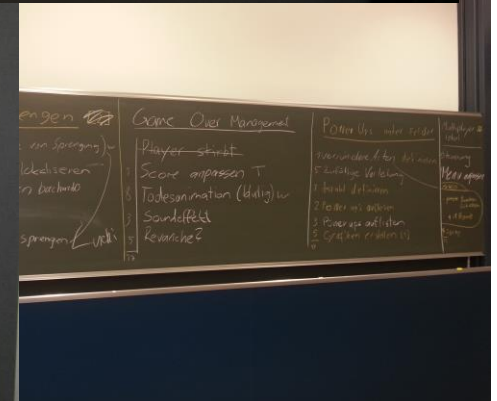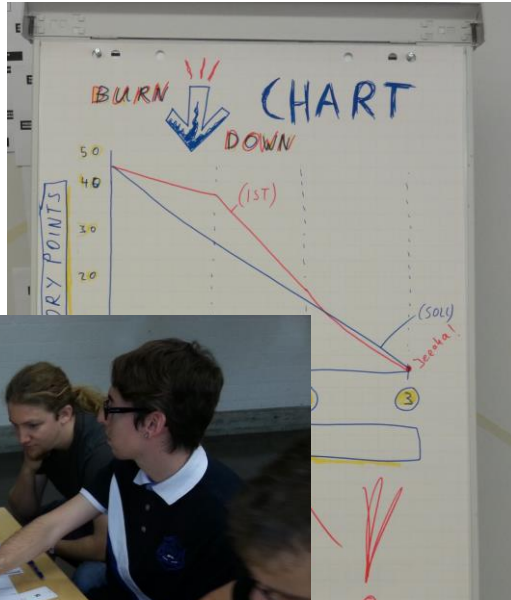
Course Evaluation

Items	Excellent	Good	Bad	Very bad
The content of this course is...	12	11	0	0
This course was divided into engineering- and management practices and agile values. How would you judge this concept?	12	11	0	0
How did the agile values come across in the lectures and workshops?	1	19	1	0
In the student project, you worked in a Scrum team of 6 to 8 fellow students. How would you judge this concept?	9	11	4	0
How would you judge the workshops in part one?	1	20	1	0
How would you judge the workshops in part two?	6	14	3	0

Course Evaluation

What did you like best about the course?

"… the development of the computer game in a Scrum team".
"… that the material in the course was not only covered theoretically but I also had the opportunity to apply and deepen it in the workshops".
"… the practical relevance".
"… that the topics covered were interesting and important. I had the opportunity to practice the newly learned in the student project. That was great!"

Kontakt

- Martin Kropp
  Institut für Mobile und Verteilte Systeme
  Fachhochschule Nordwestschweiz
  martin.kropp@fhnw.ch

- Andreas Meier
  Institut für angewandte Informationstechnologie
  Zürcher Hochschule für Angewandte Wissenschaften
  meea@zhaw.ch