Teaching Agile Software Development

Martin Kropp, FHNW
Andreas Meier, ZHAW

ECSS 2013, Amsterdam, Oct. 8.-9., 2013
Agenda

• Motivation
• Pyramid of Agile Competences
• Agile Software Engineering Course
• Evaluation
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

- Agile 57%
- Non-Agile 43%

www.swissagilestudy.ch
The Good Sides: Satisfaction

How satisfied are you with your current methodology?

- **Non Agile IT-Professionals**
  - Unsatisfied: 21%
  - Somewhat satisfied: 32%
  - Satisfied: 40%
  - Very satisfied: 7%

- **Agile IT-Professionals**
  - Unsatisfied: 2%
  - Somewhat satisfied: 25%
  - Satisfied: 53%
  - Very satisfied: 19%

- **Non-Agile Companies**
  - Unsatisfied: 6%
  - Somewhat satisfied: 31%
  - Satisfied: 52%
  - Very satisfied: 10%

- **Agile Companies**
  - Unsatisfied: 0%
  - Somewhat satisfied: 16%
  - Satisfied: 64%
  - Very satisfied: 20%

*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?

- Requirements management: 29% Much worse, 51% Worse, 13% Improved, 17% Significantly improved, 9% Unchanged, 25% Don't know
- Development process: 17% Much worse, 58% Worse, 22% Improved, 25% Significantly improved, 25% Unchanged, 19% Don't know
- Project visibility: 25% Much worse, 39% Worse, 28% Improved, 28% Significantly improved, 23% Unchanged, 22% Don't know
- Alignment between IT & business objectives: 25% Much worse, 46% Worse, 23% Improved, 23% Significantly improved, 46% Unchanged, 13% Don't know
- Ability to manage changing priorities: 9% Much worse, 45% Worse, 44% Improved, 22% Significantly improved, 45% Unchanged, 28% Don't know
- Time to market: 19% Much worse, 53% Worse, 23% Improved, 23% Significantly improved, 53% Unchanged, 22% Don't know

ECSS '13, Amsterdam
M. Kropp / A. Meier
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
The Dark Sides: Agile Influence

How has agile software development influenced the following aspects?

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Much worse</th>
<th>Worse</th>
<th>Unchanged</th>
<th>Improved</th>
<th>Significantly improved</th>
<th>Don't know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering discipline</td>
<td>42%</td>
<td>42%</td>
<td>9%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development cost</td>
<td>52%</td>
<td>22%</td>
<td>7%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software maintainability / extensibility capability</td>
<td>55%</td>
<td>23%</td>
<td>12%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software quality</td>
<td>45%</td>
<td>35%</td>
<td>16%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity</td>
<td>33%</td>
<td>47%</td>
<td>15%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ECSS '13, Amsterdam
M. Kropp / A. Meier
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

- Unit testing
- Coding standards
- Automated builds
- Continuous integration
- Refactoring
- Test Driven Development (TDD)
- Pair programming
- Collective code ownership
- Continuous delivery
- Automated acceptance testing
- Acceptance Test Driven Development (ATDD)
- Behavior Driven Development (BDD)

Diagram showing the percentage of Agile IT-Professionals and Agile Companies for each practice.
How Agile is Applied

• Managing Practices

- Release planning
- User stories
- Iteration planning
- Daily standup
- Taskboard
- Retrospective
- Burndown charts
- Story mapping
- Open work area
- Continuous delivery
- On-site customer
- Kanban Pull System/Limited WIP

![Bar chart comparing Agile IT-Professionals and Agile Companies on various practices.](image-url)
What the Industries says

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

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

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

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

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

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