Teaching Agile Software Development at University Level: Values, Management, and Craftsmanship

Martin Kropp, FHNW
Andreas Meier, ZHAW

CSEE&T 2013, San Francisco, May 20, 2013
Agenda

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

• Swiss Agile Study 2012 (SAS)
  – 140 IT companies
  – 194 IT Professionals

• Results?

Company Survey

- Agile 57%
- Non-Agile 43%

www.swissagilestudy.ch
SAS Results: 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%
Motivation

SAS shows very promising results:

• much higher satisfaction with agile methodologies than with plan-driven ones
SAS Results: Agile Influence

How has agile software development influenced the following aspects?

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

CSEE&T '13, San Francisco
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
How has agile software development influenced the following aspects?

- Engineering discipline: 42% Much worse, 42% Worse, 9% Unchanged, 9% Improved, 9% Significantly improved, 9% Don't know
- Development cost: 52% Much worse, 22% Worse, 7% Unchanged, 15% Improved, 9% Significantly improved, 9% Don't know
- Software maintainability / extensibility capability: 55% Much worse, 23% Worse, 12% Unchanged, 9% Improved, 9% Significantly improved, 9% Don't know
- Software quality: 45% Much worse, 35% Worse, 16% Unchanged, 9% Improved, 9% Significantly improved, 9% Don't know
- Productivity: 33% Much worse, 47% Worse, 15% Unchanged, 9% Improved, 9% Significantly improved, 9% 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 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)

Bar chart showing the percentages 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

0%  20%  40%  60%  80%  100%

- Agile IT-Professionals
- Agile Companies
and Agile Education ...

- What does this mean for teaching agile software development?
- Which skills and competences do the students need?
Pyramid of Agile Competences

• Technical skills or *engineering practices*, i.e. 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. are characteristics like mutual respect, openness, and courage

• The most difficult to teach
Agile Education Concept

• All three levels should be considered

• Bachelor and Master level

• Appropriate teaching methods should be applied
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 Lecture

<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
<td>JUnit</td>
</tr>
<tr>
<td></td>
<td>Clean Code / Code Smells</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Planning Poker</td>
</tr>
<tr>
<td>12</td>
<td>Metrics</td>
<td>Agile Game Development</td>
</tr>
<tr>
<td></td>
<td>Agile Teams</td>
<td>Metrics (EMMA)</td>
</tr>
<tr>
<td>13</td>
<td>User Stories</td>
<td>Agile Game Development</td>
</tr>
<tr>
<td></td>
<td>Agile Principles</td>
<td></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