„Agile Software Engineering and Design Thinking: Efficiency and Innovation in Lean Software Product Development“
Agenda

- Introduction and Overview
- Lean and Agile Development
- Agile Software Engineering Techniques
- Design Thinking and Innovation
- Summary and Conclusion
Overview: Two Main Questions...

(1) How to build and deliver the software efficiently?

(2) Where do innovative ideas and requirements come from?
Why lean and agile development at SAP?
Development Process prior to Lean

- Analyze
- Design
- Code
- Test

Fear of delivery

Year 1  Year 2  Year 3  Year 4
Lean Product Development Flow

How to efficiently deliver innovative enterprise software to customers?
Underlying Principles

The Principles of Product Development FLOW
Second generation Lean Product Development
Donald G. Reinertsen

1. Take an Economic View
2. Actively Manage Queues
3. Exploit Variability
4. Reduce Batch Size
5. Apply WIP Constraints
6. Control Flow: Cadence and Synchronization
7. Apply Fast Feedback
8. Decentralize Control?

Source: Donald G. Reinertsen (2009) – Lean Product Development Flow
Why do Scrum teams adopt agile engineering techniques?
Agile Software Engineering

- **Specific agile techniques**: Pair programming, test-driven development, continuous integration, refactoring
- **Agile method as process framework**: Teams with Scrum master, product owner, developers and an associated manager for people development
- **Lean principles**: Eliminate waste, build quality in, learn constantly, deliver fast, engage everyone, keep getting better

---

Armin Heinzl & Tobias Hildenbrand
## Adoption of ASE Techniques

<table>
<thead>
<tr>
<th></th>
<th>Adoption intensity</th>
<th></th>
<th>REF</th>
<th>CI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PP</td>
<td>TDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interview result</td>
<td>Average development time used for PP</td>
<td>Interview result</td>
<td>Average of new code written in a test-driven mode</td>
<td>Average development time used for REF</td>
</tr>
<tr>
<td>Team 1</td>
<td>Low</td>
<td>18%</td>
<td>Low</td>
<td>18%</td>
</tr>
<tr>
<td>Team 2</td>
<td>Low</td>
<td>36%</td>
<td>Varying</td>
<td>26%</td>
</tr>
<tr>
<td>Team 3</td>
<td>High</td>
<td>60%</td>
<td>High</td>
<td>36%</td>
</tr>
<tr>
<td>Team 4</td>
<td>High</td>
<td>76%</td>
<td>High</td>
<td>36%</td>
</tr>
</tbody>
</table>
Adoption Patterns for ASE

• Minor variations for refactoring (REF) and continuous integration (CI)
• Relatively low adoption rate for refactoring
• Significant variations for pair programming and test driven development
• Note: empirical inquiry represents a snapshot-view
Where do the requirements come from and how does innovation happen?
An Analytic Approach Doesn’t Work

<table>
<thead>
<tr>
<th>Idea</th>
<th>White paper</th>
<th>Detailed specification</th>
</tr>
</thead>
</table>

Armin Heinzl & Tobias Hildenbrand
Diverging and Converging in Iterations

"Wicked problem"

Less wicked problem

Solvable problem

Development
Design Thinking

A team-based, iterative approach to innovation
Lecture 3: What the Hell is Design Thinking?

Jochen Gürtler
Tobias Hildenbrand
BRAINSTORMING RULES.

Build on the Ideas of Others

Defer Judgment

Stay Focused on Topic

Fail early and Often

Be Visual

One Conversation at a Time

Think User Centric

Go for Quantity

Encourage Wild Ideas
MAKE IDEAS TANGIBLE by building prototypes.
TEST AND VALIDATE YOUR IDEAS
from the very beginning.
Three “Space of Innovation”

1. Human Desirability
2. Technical Feasibility
3. Economic Viability

Based on: Tim Brown (2009): Change by Design
Innovation = Design & Development

1. Inspiration

2. Ideation

3. Implementation
How does innovation happen in practice?
Intertwining Lean and Design Thinking

Product Backlog

User Story Mapping

Personas

Teams

Prototyping

Synthesis

User Research

Armin Heinzl & Tobias Hildenbrand
Building People-Centric Software with Lean and Design Thinking

**Case Study**

**Phase 1:** Research, Envisioning & Trainings (Scrum Mode)

**Phase 2:** Synthesis → Vision Refinement, Story Map & Backlog Elaboration

**Phase 3 (ongoing):** Development with Scrum incl. Continuous Backlog Refinement („Grooming“) and Prototyping Sessions

**LEAN & Agile Practices**

**360° Research**
- Kiel Week
- German Championship (venue: Travemuende)

**Design Thinking Methods**
- World Championship (venue: Perth)
- Olympic Games 2012 (venue: London)
What are the three things to remember?
Summary and Conclusion

(1) *Design Thinking* helps teams to find problems worth solving and create innovative solutions.

(2) *Agile Software Engineering* techniques help teams to build these solutions efficiently.

(3) Both approaches can complement the *large-scale implementation of Lean* significantly...
This is the end!

• Thank you for your attention!

• Contact info:
  – heinzl@uni-mannheim.de
  – tobias.hildenbrand@sap.com
BACKUP MATERIAL
Study with 4 Development Teams

<table>
<thead>
<tr>
<th>Team</th>
<th>Scrum Experience</th>
<th>Agile Experience</th>
<th>Team continuity</th>
<th>Software Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>24</td>
<td>7</td>
<td>24</td>
<td>open source</td>
</tr>
<tr>
<td>Team 2</td>
<td>12</td>
<td>3</td>
<td>24</td>
<td>proprietary</td>
</tr>
<tr>
<td>Team 3</td>
<td>48</td>
<td>5</td>
<td>24</td>
<td>open source</td>
</tr>
<tr>
<td>Team 4</td>
<td>18</td>
<td>8</td>
<td>12</td>
<td>open source</td>
</tr>
</tbody>
</table>
Evaluation of ASE Techniques

- **Do you enjoy the following agile techniques?**
  - Test-driven development
  - Pair programming
  - Continuous integration
  - Refactoring

- **Do you consider the following agile techniques beneficial?**

The graph shows the percentage of respondents who strongly disagree, disagree, mixed, agree, and strongly agree with the enjoyment and benefit of each agile technique.
Agile Software Engineering (ASE)