Robert Heim

Agile Software Development
Motivation

Agile Methods
  - Attributes – What is „Agile“?
  - Problems
  - Evaluation criteria
  - Examples + differences

Current movements: Programs of Agile Alliance

Discussion of Adoption
  - Software Process Improvement
  - Standards
  - Example: British Telecom
  - Empirical studies

Conclusion & Recommendations
  - Agile Framework (process)
  - Architecture Framework for Agile Processes (software)

Resources

Notes on some methods and additional slides for further explanations

17.01.2013
Motivation

- Problems:
  Delivery on time, QM, late changes, etc.

- Rapidly growing: internet and mobile apps

- Need: lighter, faster, more flexible software development process
Content

• Motivation
• **Agile Methods**
  ◦ Attributes – What is „Agile“?
  ◦ Problems
  ◦ Evaluation criteria
  ◦ Examples + differences
• Current movements: Programs of Agile Alliance
• Discussion of Adoption
  ◦ Software Process Improvement
  ◦ Standards
  ◦ Example: British Telecom
  ◦ Empirical studies
• Conclusion & Recommendations
  ◦ Agile Framework (process)
  ◦ Architecture Framework for Agile Processes (software)
• Resources
• Notes on some methods and additional slides for further explanations

17.01.2013
Agile Software Development

- „Late 1990 several methodologies began to get increasing public attention. Each had a different combination of old ideas, new ideas, and transmuted old ideas. But they all emphasized close collaboration between the programmer team and business experts; face-to-face communication (as more efficient than written documentation); frequent delivery of new deployable business value; tight, self-organizing teams; and ways to craft the code and the team such that the inevitable requirements churn was not a crisis.” – Agile Alliance
Agile Methods - Attributes

**Incremental** (max. 3 month\(^1\))
- rapid dev. cycle

**cooperative**
- close customer

**straightforward**
- easy to learn / flexible methods

**adaptive**
- last minute changes

For more see: http://agilemanifesto.org/principles.html
Agile Methods - Problems

- E.g. Agile Alliance
  - 2000+ papers since 2002\(^2\)
  - Agile Conference\(^1\) 2012:
    - ~200 talks

- But still: **lack of scientific studies** and **empirical evidence**\(^2\)

- People not aware of approaches/suitability for real-life situations\(^2,\, 14,\, \ldots\)

→ How to rate, choose and benefit from specific agile methods?
Agile Methods – Evaluation Criteria²

- Software dev. life-cycle
- Project management support
- Abstract ↔ concrete
- Universally ↔ situational
- Empirical evidence
Agile Methods – Examples and Differences

[Diagram showing project life cycle stages for various agile methods including ASD, AM, Crystal, DSDM, XP, FDD, ISD, PP, and Scrum, with notes on project management support and concrete guidance.]
Agile Methods - Choosing

- Cannot answer the questions with ease: „Should I use ASD in my upcoming project? And which methods are suitable?“

- NO STANDARD, yet (even if some ASDs are well known by many people, e.g. XP, Scrum)
Content

- Motivation

- Agile Methods
  - Attributes – What is „Agile“?
  - Problems
  - Evaluation criteria
  - Examples + differences

- **Current movements: Programs of Agile Alliance**

- Discussion of Adoption
  - Software Process Improvement
  - Standards
  - Example: British Telecom
  - Empirical studies

- Conclusion & Recommendations
  - Agile Framework (process)
  - Architecture Framework for Agile Processes (software)

- Resources

- Notes on some methods and additional slides for further explanations

17.01.2013
Agile Alliance – Current Programs

- "Supporting Agile Adoption: It’s about Change"
  → organizational change (compare Lichter)

In 2012:

- Characteristics of agile organizations
- AWG – A sustainable Engine for Enterprise Agile Adoption
- Project Studies Program: longitudinal studies
Content

• Motivation

• Agile Methods
  ◦ Attributes – What is „Agile“?
  ◦ Problems
  ◦ Evaluation criteria
  ◦ Examples + differences

• Current movements: Programs of Agile Alliance

• Discussion of Adoption
  ◦ Software Process Improvement
  ◦ Standards
  ◦ Example: British Telecom
  ◦ Empirical studies

• Conclusion & Recommendations
  ◦ Agile Framework (process)
  ◦ Architecture Framework for Agile Processes (software)

• Resources

• Notes on some methods and additional slides for further explanations

17.01.2013
Agility and Software Process Improvement (SPI)\textsuperscript{16}

- **Success factors**
  - **People**, technology, process
  - Team needs **skills + tools** required to apply process

- "Never apply process models for their own sake!“\textsuperscript{16}

- **Fitting Portfolio** of Process Models (PM)

- Monitor process, to see if it still fits
  \rightarrow Quality Management e.g. software metrics\textsuperscript{12}
Standards

- 2003: **XP connectable with current ISO “software life cycle process”**
  - Main idea: “working software over comprehensive documentation“
  - practical evidence missing?!
  - Conclusion: *project specific standards*

- Agile Alliance Program (2012): *Costs of IT Projects*
  in coop. FASB (founded 1973):

  **Agile Accounting Standard**
  - “define and standardize internal IT development costs […]“
  - “[…] for organizations that use an […] agile software development methodology”

- →But: no standard for ASD it self ←
Example – British Telecom

- Environment:
  - 2006, ~8000 IT professionals
  - Before: waterfall with CMMI, 2 years later: DSDM + Scrum
  - Aim/Focus: Delivering Business Value

- Problems / Risks:
  - Architecture of IT-Organization + mindsets of people have to be changed
  - Complexity
  - Outsourced parts of code, where no tests exists

- Practice:
  - Before: 12month + cycles
  - Now: 90 days cycle
  - First 3 days cross-functional teams exploring one business problem
  - At day 90 want deployable, fully-tested solution
  - within that 90days use internal agile methods (like 2-4week cycles).

- Result: challenging, but early success and they would not turn back

17.01.2013 11/22
Results of (good) empirical study

- Austria 2008
- Interviewed developer and manager (but answers did not differ as much as expected)
- 100 companies, 42% acceptance-rate
- 40.5% large organizations (>250)
- Focus: XP, especially Pair-Programming
- Results:
  - Most know XP (~40%) and Scrum (~30%)
  - General awareness of ASD exists
  - But lack of knowledge of concrete practices
Results of (good) empirical study

- Results
  - People are main obstacle to practical application of ASD,
    Most important:
    - Lack of knowledge and time
    - refusal of management

- Concrete Reality:
  Agile methods used on demand
  - Test-first:
    - 75% rarely use it (only when needed)
  - Pair-programming:
    - 32% only for complex tasks
    - 7.1% for tutoring (or on demand)

→ No really systematic approach towards agile development in practice (2008)
Results of quantitative analysis\(^7\) (to be understood carefully)

- 2010 Pakistan
- Online survey, 200 requests, 42 responses (21%), **Focus: Impact on Productivity**
- Results:
  - Mainly Scrum, MSF and XP known/used
  - Fewer but *competent/experienced developers* deliver outputs in lesser time and better quality
  - *Train people* to be up to date with tools and technologies
Results of quantitative analysis\(^7\) (to be understood carefully)

- 100% do knowledge sharing as an essential constituent of agile methodologies
- Techniques used:

![Chart showing various techniques with percentage of use]
Results of quantitative analysis\(^7\) (to be understood carefully)

Increase in productivity / quality

![Bar chart showing comparison between productivity and quality across different levels of increase: Much Lower, Somewhat, No Change, Somewhat Higher, Much Higher, Don't Know. The chart indicates a significant increase in productivity compared to quality across all categories.](chart.png)
Content

- Motivation
- Agile Methods
  - Attributes – What is „Agile“?
  - Problems
  - Evaluation criteria
  - Examples + differences
- Current movements: Programs of Agile Alliance
- Discussion of Adoption
  - Software Process Improvement
  - Standards
  - Example: British Telecom
  - Empirical studies
- Conclusion & Recommendations
  - Agile Framework (process)
  - Architecture Framework for Agile Processes (software)
- Resources
- Notes on some methods and additional slides for further explanations
Conclusion

- **Pro‘s**
  - Fast delivery in praxis
  - Easy to learn (compared to other SPI-methods)
  - Early success/results in most companies could been shown

- **Con‘s**
  - Missing evidence → difficult method-decision
  - Self-organizing teams not always best
  - Customer‘s role stressful and not sustainable over long periods
  - May fail in large organizations (controversial)
    → High commitment
Conclusion – State of Research

- 2008 - Roadmap for empirical research: Scientific research and experimental evidence needed
  - Done: XP, Scrum, especially Pair-Programming
  - Test-driven: productivity increase less obvious and as seen rarely used in reality

- Research should concentrate on:
  - Include criteria of other fields / theories
  - Experienced ASD teams and organizations
  - Longitudinal studies
  - Prioritize management-oriented approaches

- If interested: see problem definition and some practical results (mostly Scrum): 2007 - Exploratory study from Microsoft Research calling for further studies
Conclusion – State of Research

- 2008 and Goal for 2015

![Spider diagram showing various aspects of research maturity and coverage.](image)
Conclusion - Reality

- Adoption / choosing methods:
  - ASD needs to be **accepted** by all participants
  - Study **characteristics** of
    - Project
    - **Organization structure & culture** \(^{18,19}\)
    - Regulations
    - compare to methods‘ characteristics to find matches
  - **Train** people (best in University\(^{20}\))

See [10] for further practical & realistic results
Content

• Motivation
• Agile Methods
  ◦ Attributes – What is „Agile“?
  ◦ Problems
  ◦ Evaluation criteria
  ◦ Examples + differences
• Current movements: Programs of Agile Alliance
• Discussion of Adoption
  ◦ Software Process Improvement
  ◦ Standards
  ◦ Example: British Telecom
  ◦ Empirical studies
• Conclusion & Recommendations
  ◦ Agile Framework (process)
  ◦ Architecture Framework for Agile Processes (software)
• Resources
• Notes on some methods and additional slides for further explanations

17.01.2013
Refactoring to change internal code without affecting external functionality

See [7] for more details
Problem: "ASD is an excuse for developers to implement as they like, coding away without proper planning or design [...] and consequently causing suboptimal design decisions."

Approach 2010
Architecture Framework for Agile processes (AFA)

Idea: **Decoupling Components:**

- Simple APIs of internet addressable containers
- Layers with different responsibilities
- Pro's: allowing changes in architecture with minimal effects on other components → less testing → better time-to-market
- Con's/Feature work: benchmark/performance because additional layers

Compare [6] (2010) for more detailed and practical scenarios and decision making, including freezing the architecture and how to decide when to do it, calling for more research
Literature / Resources

Literature / Resources


17] Agile Alliance - Resources http://www.agilealliance.org/resources


Fee-based


25] AIS - Association for Information Systems - 2011 - Overview and Guidance on Agile Development in Large Organizations http://aisel.aisnet.org/cais/vol29/iss1/2/
APPENDIX
(E)Xtreme Programming (XP)

- Developed at Chrysler ~1996 – 2000
- Collection of concrete best practices
- E.g.
  - Short iterations
  - Pair programming
  - Rapid feedback / close customer
  - Communication
  - Continuous refactoring + integration + testing
  - ...
Scrum

• ~1995
• Manage Software development process
• Empirical base, focus on
  ◦ Flexibility
  ◦ Adaptability
  ◦ Productivity
  ◦ Frequent management activities + Sprints
  ◦ Developer chooses specific techniques/practices for implementation
Adaptive Software Development (ASD)

- ~2000
- Adaptive/incremental paradigm instead of waterfall
- Constant prototyping
- Framework for guidance to prevent chaos, but not suppressing emergence and creativity
Agile Modeling (AM)

- ~2002
- Focus: modeling and cultural principles
- Modeling using agile philosophy
- Advanced models
- Aim: amount of models and documentation as low as possible
Crystal Family

- ~1998
- Different methods
- Select best fitting based on „color-rating“ (size and criticality)
- Tailoring methods to fit needs
- Integrate other agile methods like XP/Scrum
Dynamic Systems Development Method (DSDM)

- One of the first truly agile methods (~1994)
- Normally: fixed functionality + adjust time/resources
  Here: fix time/resources and adjust functionality
Feature Driven Development (FDD)

- ~2002
- Process oriented for business systems
- Design and building phases
- Iterative
- Quality aspects in process → monitoring of progress
Internet Speed Development (ISD)

~2001

Need: Fast releases in a chaotic fast moving process

Descriptive, management-oriented framework to handle fast iterations:
  ◦ Time-drivers
  ◦ quality dependencies
  ◦ good people = less process
Pragmatic Programming (PP)

- ~2000
- Collection of best practices
- ~70 so called „tips“ focusing day-to-day problems
- Incremental, iterative development, testing, user-centered design
Description of current status and suggested goal for 2015

<table>
<thead>
<tr>
<th>Roadmap Area</th>
<th>Description</th>
<th>Status 2008</th>
<th>Goal 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maturity</td>
<td>Cohesiveness of the field</td>
<td>Some joint large research efforts, however mainly in Europe</td>
<td>Several large research projects. Research methodology as mature as in information systems.</td>
</tr>
<tr>
<td></td>
<td>Research methodology</td>
<td>Methodological problems as in software engineering</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial standardization work under way</td>
<td></td>
</tr>
<tr>
<td>Coverage</td>
<td>Process/technique/phase</td>
<td>XP and some techniques covered for inexperienced teams</td>
<td>Significantly increased coverage of agile software development methods for experienced software development teams and large organizations. Good coverage of management-oriented approaches.</td>
</tr>
<tr>
<td></td>
<td>Problem domain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Artefact scale</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subject expertise level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understanding</td>
<td>Evidence</td>
<td>Intermediate level in certain areas, nascent on most</td>
<td>Mature research on certain topic areas, intermediate on a number of areas. Significantly better understanding on what agile software development is.</td>
</tr>
<tr>
<td>Impact</td>
<td>Industry</td>
<td>Little impact on how agile development is carried out in industry, and little impact on other research fields, some impact in education</td>
<td>Research has a significant impact on certain areas of agile software development on industry, science and education.</td>
</tr>
<tr>
<td></td>
<td>Science/Engineering</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Education</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>