I assume:
Audience has heard the term „Agile Software Development“ and has heard of practices like „Pair programming“, „Test-Driven-Development“, „Prototyping“, ...

I will: be abstract on the methods itself (one can easily look them up if further info needed), but be detailed on how to rate methods
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
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
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

deployable business value = less documentation
A cycle is e.g.: 1. Functionality needed immediately, 2. Fast implementation 3. Collect feedback 4. react rapidly to business + technology changes

[16] Lichter, reference models, SWC department
Agile Methods - Problems

- E.g. Agile Alliance
  - 2000+ papers since 2002
  - Agile Conference 2012: ~200 talks
- But still: lack of scientific studies and empirical evidence
- People not aware of approaches/suitability for real-life situations
- How to rate, choose and benefit from specific agile methods?


[14] Austrian study
Agile Methods – Evaluation Criteria

- Software dev. life-cycle
- Project management support
- Abstract ↔ concrete
- Universally ↔ situational
- Empirical evidence

Abstract / concrete: principles ↔ concrete guidance
Universally = fit per se in all agile dev. situations
Agile Methods – Examples and Differences²

Inception = project begin/founding of project

ASD = Adaptive Software Development
AM = Agile Modeling
DSDM = Dynamic Systems Development Method
FDD = Feature Driven Development
ISD = Internet Speed Development
PP = Pragmatic Programming
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: a nonprofit organization with global membership, committed to advancing Agile development principles and practices

Compare: my seminar 10/11 on improvement models: business has to address the culture of the organization

AWG = Agile Working Group = Scrum team, with product: enterprise implementation of Agile, evidence of management support
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)

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

- “Never apply process models for their own sake!”

- Fitting Portfolio of Process Models (PM)

- Monitor process, to see if it still fits
  (→ Quality Management e.g. software metrics)

Portfolio = Organization choose the mix of old (like CMMI) and new (like XP) models
PM Selection criteria:
  Organization-culture, developer-team, project-constraints, business-goals
  Needs to be accepted from all involved people (that it supports their daily-work)
  (compare [14] Austria study)

[16] = Lichter Software Processes in an Agile World. Department did many research on Software-Process-Improvement Models, Seminar WS10/11 SCRUM in large organizations works
[  7] = study from Pakistan, Agile Software Development : Impact on Productivity and Quality
[12] = software metrics for ASD, University of Magdeburg, Germany
Projects often need to fit ISO standard (due to organization, management, regulation)

11: **Standards and Agile Software Development.** → abstract, XP and ISO norm for "software life cycle" integration, but only abstract. To fit to docu-standards-requirements: in XP as system requirement

FASB (financial accounting standards board): develop generally accepted accounting principles (GAAP) within the United States in the public's interest
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 exist

- 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 2-4 week cycles).

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

CMMI = Capability Maturity Model Integration

DSDM = Dynamic systems development method

High risk = program has business commitments to make and little scope to take extra risk of adopting new practices

→ Deliver business value or close program if it fails to deliver value over some cycles
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)

Bad Mindset: “Adoption of some practices is enough for agile dev.”
Lack of knowledge (but only 1/3 wanted more information)
Results of quantitative analysis (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

MSF = Microsoft Solutions Framework
Regression test = test to ensure that a new change has not spawn new bugs. 
CI: small incremental changes, developer often commit, each commit = build hole system + ideally tests 
TDD a little controversial to other study, cause they say test-FIRST rarely used
Results of quantitative analysis
(to be understood carefully)
Increase in productivity / quality

![Bar chart showing productivity and quality increases](image-url)
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
Self-organizing not best: in complex, non-linear problem-solving situations

Controversy: experienced developer tend to adopt ASD better, risk higher, see British Telcom

[10 ] = 2005 What do we know about Agile Software Development?
Conclusion – State of Research

- 2008 - Roadmap for empirical research: *Scientific research and experimental evidence needed* [22, 10]
  - Done: XP, Scrum [22], 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 [23]

Other fields: problem: SE references only SE research, but what is with not directly related to agile methods, e.g. task-technology-fit, social and organizational learning theory, knowledge-based theory of firm, ...

Longitudinal studies: to see effects on other components of business

[10] = 2009, What do we know about ASD, Frauenhofer Center for Experimental SE
[23] = promising Exploratory Study from Microsoft
Most important:
Expertise = competence
Research methodology
Evidence / impact on industry

Cohesiveness = how established ASD is


Additional slide for explanation at end
Conclusion - Reality

- Adoption / choosing methods:
  - ASD needs to be accepted by all participants
  - Study characteristics of
    - Project
    - Organization structure & culture
    - Regulations
    - compare to methods' characteristics to find matches
  - Train people (best in University)

Participants: developers, managers, customers, etc.

[10] = 2009, What do we know about ASD, Fraunhofer Center for Experimental SE
[18] = 2012 Agile Alliance - Characteristics of Agile Organizations. 2012,
[20] = 2009 Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum
### 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
Recommended Agile Framework (2010)

- **Refactoring** to change internal code without affecting external functionality

- See [7] for more details

emphasis flexible design, to be able to integrate late changes
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


Container = collection of entries accessible via basic API

[13] = guys from Vienna University of Technology, Austria

Coordination: e.g.: Fifo instead of Lifo can be changed, without touching algorithms for data computation

Literature / Resources


17.01.2013 L1
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/iss1/2/17.01.2013 L2
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
  - ...

17.01.2013 A1
Scrum
• ~1995
• Manage Software development process
• Empirical base, focus on
  ◦ Flexibility
  ◦ Adaptability
  ◦ Productivity
  ◦ Frequent management activities + Sprints
  ◦ Developer chooses specific techniques/practices for implementation

17.01.2013 A2
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 2015</th>
<th>Goal 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics</td>
<td>Knowledge of the field, Research methodology</td>
<td>Some joint lazy research efforts, however mainly in Europe</td>
<td>Several large research projects. Research methodology as a service to inform systems.</td>
</tr>
<tr>
<td>Coverage</td>
<td>Process techniques (time, Peloton domain, Artificial scales) Subject expertise levels</td>
<td>Intervention techniques: covered for most unprecedented forms</td>
<td>Significantly increased coverage of agile software development methods for experimental software development topics and large organisations. Good coverage of management-oriented approaches.</td>
</tr>
<tr>
<td>Understanding</td>
<td>Evidence</td>
<td>Intermediate level or certain areas, assurance most</td>
<td>Nature research in certain topic areas, intensive on a number of axes. Significantly better understanding on what agile software development is.</td>
</tr>
<tr>
<td>Impact</td>
<td>Industry, Science, Engineering, Education</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. Industry, science, and education.</td>
</tr>
</tbody>
</table>