Agile Requirements Engineering practices

Seminar „Beiträge zum Software Engineering“
WS 2018/19
Motivation

Understand:

- what RE is
- which benefits and challenges do agile RE practices present
- and where do they come from
Definition: Agile

Agile – „A group of software development methodologies based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.” [SGSE]
Definition: Requirements

Requirement – „A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.” [SGSE]

Good requirements: [BSWE]
- Unambiguous: „Find the best route from a start location to a destination location“
  What is „the best“ route ?
- Verifiable: „Process more work orders per hour than are currently being processed.“
  How many work orders is “more?”

Avoid unquantified comparatives („faster“, „better“) and vague commands („minimize“, „improve“)!
Definition: Requirements Engineering

Requirements Engineering:

„A sub-discipline of systems engineering (...) concerned with determining the goals, functions, and constraints of hardware and software systems“ [SGSE]

Usually associated with identifying requirements, but actually consists of many activities – discovering, qualifying, tracing …
## Scientific papers: choice

<table>
<thead>
<tr>
<th>Elizabeth Bjarnason</th>
<th>Lan Cao</th>
</tr>
</thead>
<tbody>
<tr>
<td>Krzysztof Wnuk</td>
<td>Balasubramaniam Ramesh</td>
</tr>
<tr>
<td>Björn Regnell</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AREW ‘11</td>
<td>IEEE Software</td>
</tr>
</tbody>
</table>

### Study 1

### Study 2
## Scientific papers: basic comparison

<table>
<thead>
<tr>
<th>Research method</th>
<th>Case</th>
<th>Agile methods</th>
<th>Research questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study 1</td>
<td>Single case study</td>
<td>Industrial projects at a large company</td>
<td>Partially introduced Scrum-like development process</td>
</tr>
<tr>
<td>Study 2</td>
<td>Multi-site case study</td>
<td>Sixteen different organizations</td>
<td>No specific or XP or Scrum or both</td>
</tr>
</tbody>
</table>
Study 1: data collection and analysis

Data collection and analysis

- semi-structured interviews
- 9 practitioners interviewed
- content analysis of the transcripts
Study 2: Data collection and analysis

Data collection and analysis

- semi-structured interviews, participant observations, documentation review
- 1-7 practitioners per company interviewed (total: 59)
- grounded-theory method

<table>
<thead>
<tr>
<th>Adoption level</th>
<th>Face-to-face communication</th>
<th>Iterative RE</th>
<th>Extreme prioritization</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Medium</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>None</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constant planning</th>
<th>Prototyping</th>
<th>Test-driven development</th>
<th>Reviews &amp; tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>8</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>
## Overview of agile RE practices

<table>
<thead>
<tr>
<th>Practice Study 1</th>
<th>Practice Study 2</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Continuous Scope Flow</td>
<td>Requirement prioritization goes extreme</td>
<td>Scope</td>
</tr>
<tr>
<td>Gradual &amp; Iterative Detailing of</td>
<td>Iterative Requirements Engineering</td>
<td>Iterative detailing of requirements</td>
</tr>
<tr>
<td>Requirements</td>
<td>Face-to-face-communication over written specifications, Prototyping, Use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>review meetings and acceptance tests, Integrated RE</td>
<td></td>
</tr>
<tr>
<td>Cross-Functional Development Teams</td>
<td></td>
<td>Communication</td>
</tr>
<tr>
<td>User Stories &amp; Acceptance Criteria</td>
<td>Test-driven development</td>
<td>Extreme Programming (XP)</td>
</tr>
</tbody>
</table>

**Ungrouped practice:**
Managing requirements change through constant planning (Study 2)
Scope

- Practices:
  - One Continuous Scope Flow (Study 1)
  - Requirement prioritization goes extreme (Study 2)

- Benefits:
  - Reduced overscoping
  - Multiple opportunities for prioritisation
  - Better understanding of customer priorities

- Challenges:
  - Accomodating of non-functional requirements
Iterative detailing of requirements

- Practices:
  - Gradual & Iterative Detailing of Requirements (Study 1)
  - Iterative Requirements Engineering (Study 2)

- Benefits:
  - Clear and stable requirements
  - More satisfactory relationship with the customer

- Challenges:
  - No requirements picture at the beginning
  - Inaccurate cost and schedule estimations
  - Neglect of nonfunctional requirements
Communication (1/2)

- Practices (subsection Scrum):
  - Cross-Functional Development Teams (Study 1)
  - Face-to-face-communication over written specifications (Study 2)
  - Use review meetings and acceptance tests (Study 2)

- Benefits:
  - Unclarities in the requirements can be resolved early on
  - Saving time (less need for documentation)

- Challenges:
  - (C-F Teams) Ensuring sufficient test competence within the team.
  - (C-F Teams) Getting the development teams to document requirements
  - Depends on availability and trust
Communication (2/2)

- Practices:
  - Prototyping (Study 2)
  - Integrated Requirements Engineering process (Study 2)

- Benefits:
  - (Prototyping) Quick feedback from customer
  - (IRE) Support active discussions

- Challenges:
  - (Prototyping) Risk of deployment in production
  - (Prototyping) Unrealistic expectations of customers
Extreme Programming (XP)

- Practices:
  - User Stories & Acceptance Criteria (Study 1)
  - Test-driven development (Study 2)

- Benefits:
  - (User Stories) Better communication between business and engineers
  - (Acceptance Criteria) Increased motivation of developers
  - (TDD) Documentation linked to production code

- Challenges:
  - (TDD) Requires a lot of discipline

As a <role>
I want <goal>
So that <benefit>

Acceptance criteria:
...

Agile Requirements Engineering
Practice: Managing requirements change through constant planning (Study 2)

- **Benefits:**
  - Minimized need for major changes
  - Lower cost of change

- **Challenges:**
  - Inadequate architecture
Limitations and criticism

- Covers only practices adopted by the case company
- Some of them were not fully implemented → no challenges mentioned for them
- “development process partly influenced by Scrum”

<table>
<thead>
<tr>
<th>Study 1</th>
<th>Study 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 10 (out of 16) organizations didn’t explicitly follow any specific agile methods</td>
<td></td>
</tr>
<tr>
<td>- Interviewees: sometimes mix of managers and developers, sometimes only an architect or a single developer</td>
<td></td>
</tr>
</tbody>
</table>
Change requirements at any time?

LOOK THIS NEW AGILE THING: TO DEAL WITH UNPREDICTABLE EVENTS AND THINGS WE CANNOT CONTROL IN OUR PROJECTS

WE CAN PRIORITIZE, REDUCE THE SCOPE, CHANGE REQUIREMENTS AT ANY TIME AND INCREASE THE CHANCES OF SUCCESS OF THE PROJECT

LOOK, THIS IS YOUR NEW PROJECT, WITH FIXED DEADLINE, FIXED SCOPE AND FIXED QUALITY: YOU CAN BE “AGILE” INSIDE THIS TRIANGLE !!!

Dilbert characters Scott Adams Inc.
### Summary

- Agile RE practices remedy some of the problems of classical RE
- It is a hard and comprehensive task to transfer an organization follow agile RE practices
- Agile environment leads to a set of specific RE practices
- Intensive communication is the most important practice
- Both studies identified benefits and uncovered challenges of Agile RE practices

<table>
<thead>
<tr>
<th>Study 1</th>
<th>Study 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Freie Universität Berlin
Sources

https://dl.acm.org/citation.cfm?id=2068786
http://www.theagingexperience.com/wp-content/uploads/2017/05/Face-to-Face-3.jpg
https://c1.staticflickr.com/7/6087/6059269914_dce8761fff_b.jpg
https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template


Christof Ebert, Requirements Engineering, 2011

https://dzone.com/articles/applying-8020-rule-software
https://www.cmcrossroads.com/article/agile-difference-scm
http://www.agilemodeling.com/essays/costOfChange.htm
https://dzone.com/articles/software-requirement-document-1