Beniamin Sypianski
Institut für Informatik der FU Berlin

Freie Universität Berlin

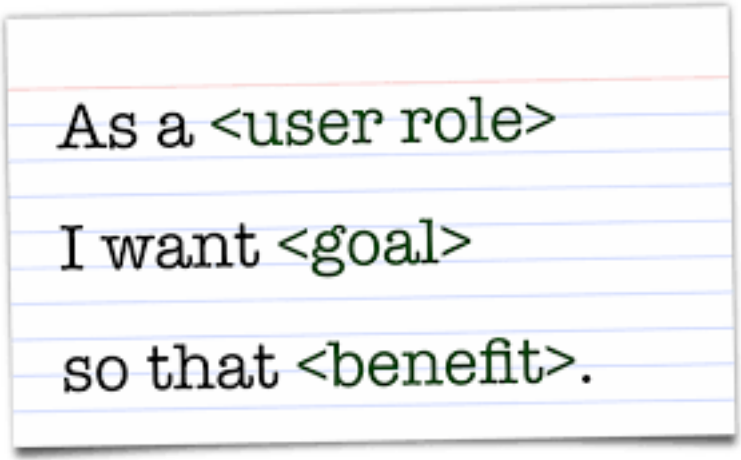
Agile Requirements Engineering practices

Seminar „Beiträge zum Software Engineering“ WS 2018/19

Motivation

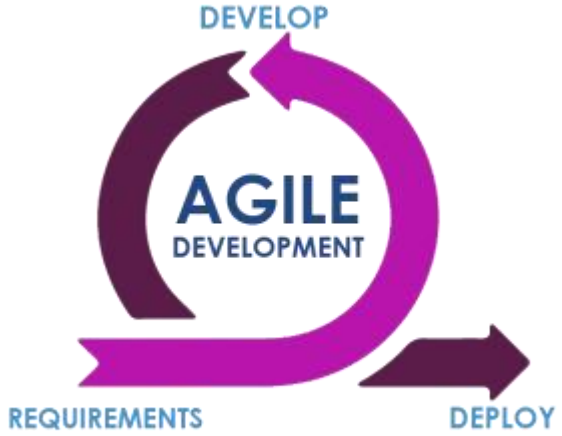
Understand:

- 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]

- Unambigous: „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?”

<u>Avoid unquantified comparatives („faster“, „better“)</u>

<u>and vague commands („minimize“, „improve“) !</u>

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 asscociated with identifying requirements, but actually consists of many activities – discovering, qualifying, tracing …

Scientific papers: choice

Elizabeth Bjarnason **Krzysztof Wnuk** **Björn Regnell**	**Lan Cao** **Balasubramaniam Ramesh**
A Case Study on Benefits and Side-Effects of Agile Practices in Large-Scale Requirements Engineering (2011) AREW ‘11	Agile Requirements Engineering Practices: An Empirical Study (2008) IEEE Software
Study 1	Study 2


A Case Study on Benefits and Side-Effects of Agile Practices in Large-Scale Requirements Engineering

Elizabeth Bjarnason, Krzysztof Wnuk, Björn Regnell

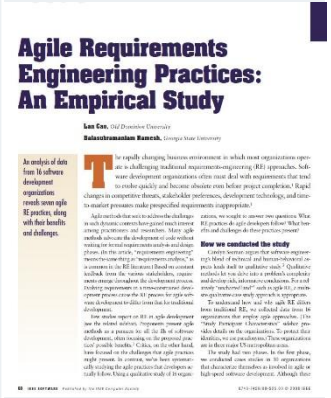
Department of Computer Science, Lund University, Lund, Sweden

{elizabeth.bjarnason | krzysztof.wnuk | bjorn.regnell}@cs.lth.se

ABSTRACT

1. INTRODUCTION

2. THE CASE COMPANY


Agile Requirements Engineering Practices: An Empirical Study

Lan Cao, Old Dominion University

Balasubramaniam Ramesh, Georgia State University

An analysis of data from 16 software development organizations reveals seven agile RE practices, along with their benefits and challenges.

How we conducted the study

Scientific papers: basic comparison

	Study 1	Study 2
Research method	Single case study	Multi-site case study
Case	Industrial projects at a large company	Sixteen different organizations
Agile methods	Partially introduced Scrum-like development process	No specific or XP or Scrum or both
Research questions	How do agile RE practices address the challenges of traditional RE ? What new challenges do agile RE practices incur ?	What RE practices do agile developers follow? What benefits and challenges do these practices present?

Study 1: data collection and analysis

Data collection and analysis

- semi-structured interviews
- 9 practitioners interviewed
- content analysis of the transcripts

Study 1

Agile RE practices:	One cont scope flow	Cross-functional teams	Integrated RE	Gradual detailing	User stories & ATC
Addressed RE Challenges					
Communication gaps	1	9	4	5	2
Overscoping	6	4		4	
Keeping SRS Updated		4		4	2
Dev work not monitored fr reqs		2		1	
Unclear requirement coverage		1			
Customer expectations not met		2			1
Low motivation for reqs work	1	1			1
Quality issues		4			
Waste	2	1		4	1
Low reqs quality		1			1
Unreliable SRS				1	
Unstable SRS		1			
Challenges of the agile practices					
Planning for agility	3			3	
Weak reqs prioritization	3				
Weak effort estimates	1				
Quality issues	1				
System completed late	1				
Capturing innovation	1				
Lack of documented reqs		1			
Customer-proxy role		2			
Ensuring competence (RE, VV)		5			
Motivating teams for reqs work		3			
Weak requirements at start				2	

Study 2: data collection and analysis

Data collection and analysis

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

Study 2

Adoption level	Face-to-face communication	Iterative RE	Extreme prioritization
High	8	9	10
Medium	8	5	6
Low	0	2	0
None	0	0	0

Constant planning	Prototyping	Test-driven development	Reviews & tests
8	8	5	11
6	3	1	4
2	0	0	1
0	5	10	0

Overview of agile RE practices

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

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

Scope

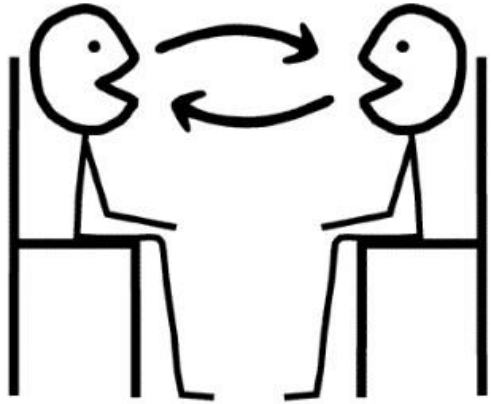
- Practices:
  - One Continuous Scope Flow (Study 1)
  - Requirement prioritization goes extreme (Study 2)
- Benefits:
  - Reduced overscoping
  - Multiple opportunities for prioritisation
  - Better understanding of customer priorities
- Challenges:
  - Accomodating of non-functional requirements

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

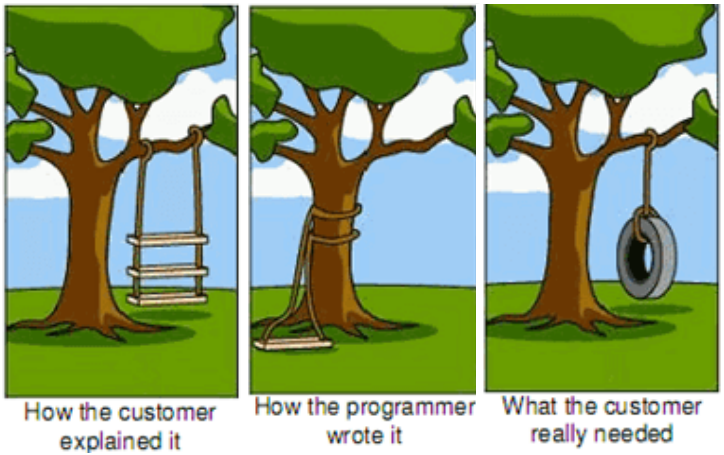
Communication (1/2)

- 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

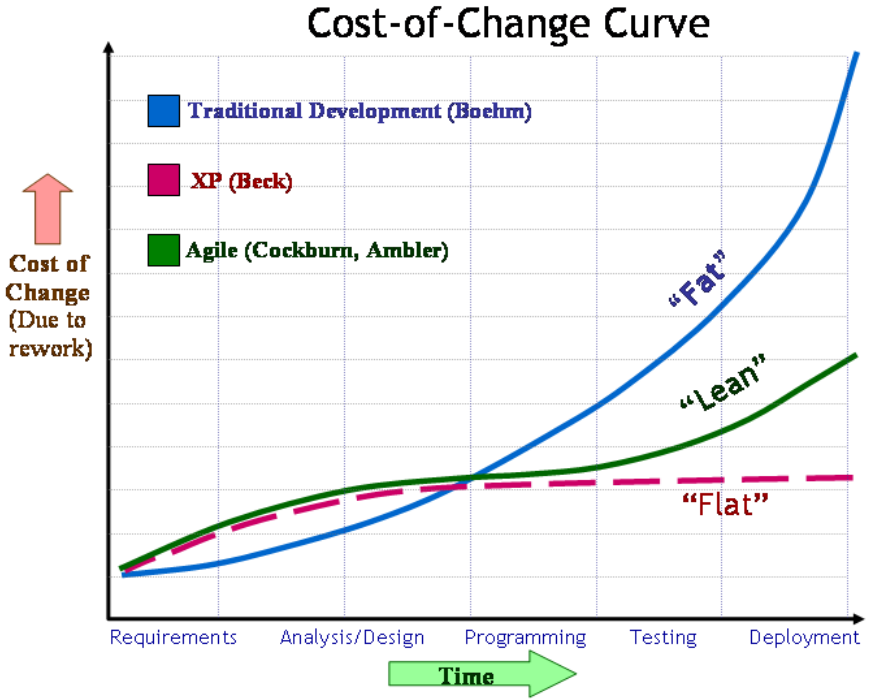


Extreme Programming (XP)

- 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

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

Study 1	Study 2
- 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”	- 10 (out of 16) organizations didn’t explicitly follow any specific agile methods - Interviewees: sometimes mix of managers and developers, sometimes only an architect or a single developer

Change requirements at any time ?

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	
Study 1	Study 2

Sources

https://dl.acm.org/citation.cfm?id=2068786

https://ieeexplore.ieee.org/document/4420071?reload=true

https://project-management.com/wp-content/uploads/2014/04/Prioritizing-and-Planning-300x193.png

http://researchaccess.com/wp-content/uploads/2013/12/iStock_agile_cycle.jpg

https://www.testingexcellence.com/wp-content/uploads/2016/05/User_Story.jpg

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://studying-in-canada.org/wp-content/uploads/2014/08/Studying-Canada-Admission-Requirements-700x420.jpg

http://www.keystepstosuccess.com/wp-content/uploads/2016/08/agile_manifesto_original.jpg

https://www.scrumwithstyle.com/wp-content/uploads/2018/02/User-Story-Card.png

https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template

[SGSE] https://www.astqb.org/documents/Standard-glossary-of-terms-used-in-Software-Engineering-1.0-IQBBA.pdf

Rod Stephens, *Beginning software engineering*, Indianapolis, Indiana : Wrox, 2015

Christof Ebert, Requirements Engineering, 2011

https://dzone.com/articles/applying-8020-rule-software

https://link.springer.com/book/10.1007%2F978-3-319-61073-3

http://tamingdata.com/wp-content/uploads/2010/07/tree-swing-project-management-large.png

https://www.cmcrossroads.com/article/agile-difference-scm

http://www.agilemodeling.com/essays/costOfChange.htm

https://dzone.com/articles/software-requirement-document-1