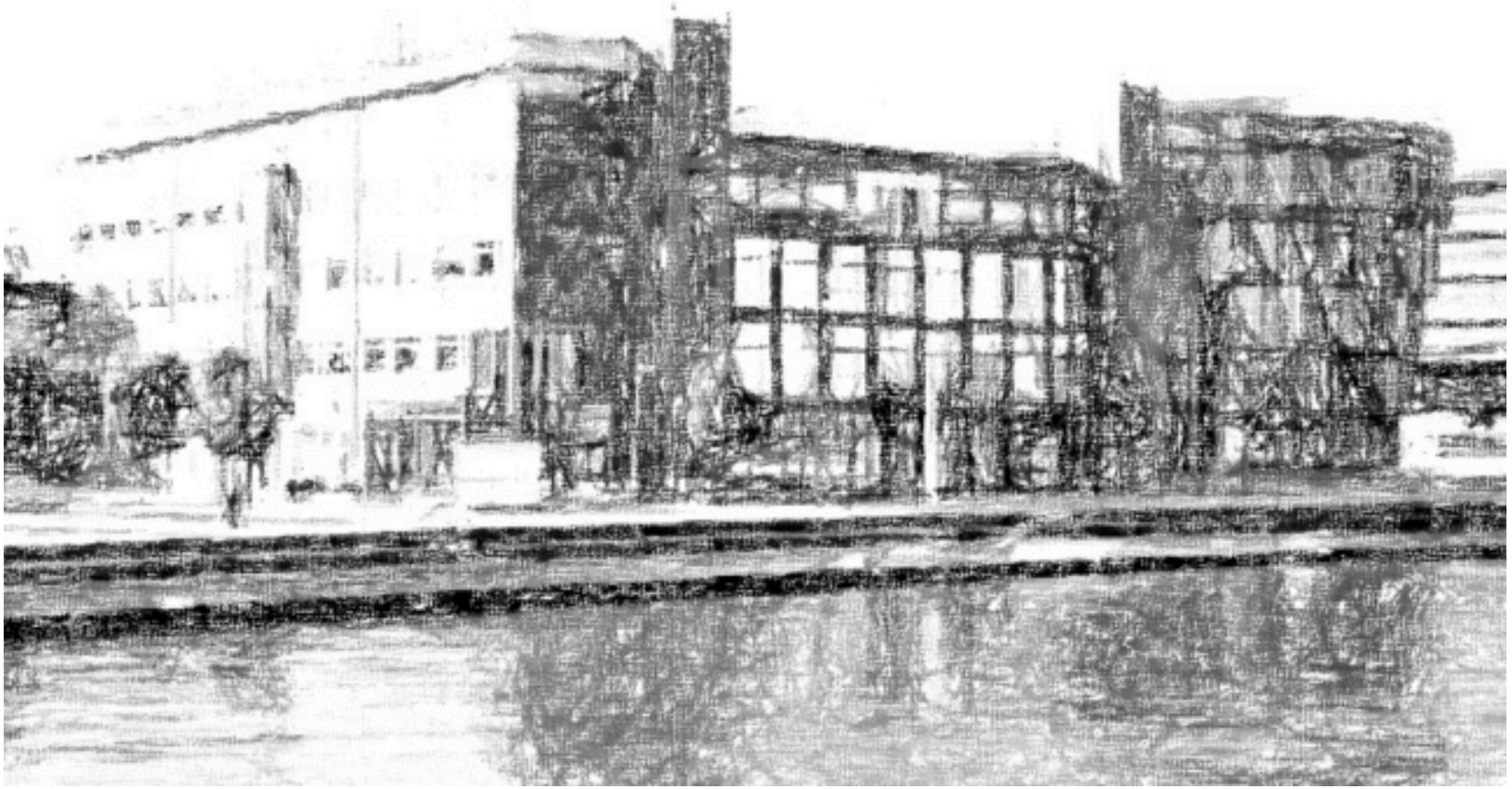
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Software Engineering Experience Circle

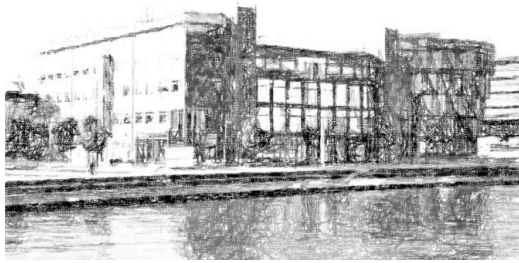
16th Mar 2015, RE in agile context

brought to you by:

Division of Software Engineering, Lindholmen Science Park, and Software Center



SEEC Workshop Series



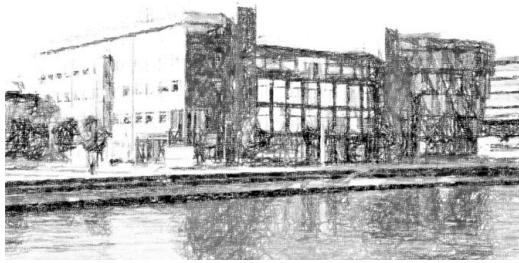
SEEC | Eric Knauss
http://seecgot.wordpress.com

- Today: 5th SEEC Workshop

- January 2014: Requirements Engineering and Testing
- June 2014: Organizational Change and Learning
- August 2014: Software Ecosystems
- November 2014: Agility, Modeling, and Quality Assurance in a hot pot

- ***March 2015: Do we still need requirements in an agile world?***

Summary

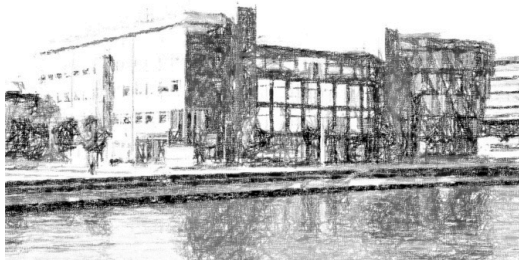


SEEC | Eric Knauss
http://seecgot.wordpress.com

- Co-organized by
  - Division of Software Engineering, Chalmers | Univ. of Gothenborg
  - Software Center
  - Lindholmen Science Park

- Goal: Offer a **forum for practitioners** to exchange **experiences about hot software engineering topic**.
  - **Open discussion** of SE challenges and potential approaches among practitioners from different organizations
  - **Networking** around specific SE topics will help individuals to be more effective in the challenging environment of engineering and managing today's software.

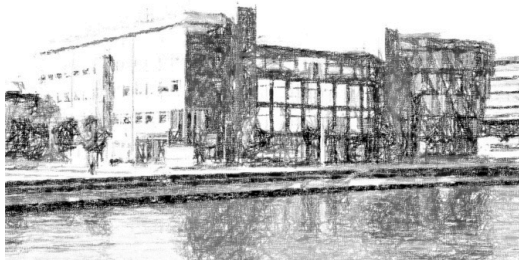
Make it Your SEEC



SEEC | Eric Knauss
http://seecgot.wordpress.com

- You have a burning topic and/or presentation?
  → get it into SEEC
- You have a relevant resource?
  → share it on seecgot.wordpress.com
- You are looking for more information on a SEEC topic?
  → find it on seecgot.wordpress.com
- And of course: Discussion
  → short round of introduction

Agenda



Time	Activity
13:00 – 13 :10	Welcome
13:10 – 13:25	**Eric Knauss** – Introduction to Workshop Theme
13:25 – 13:55	**Baldvin Gislason Bern** – Successful without Requirements Engineering
13:55 – 14:00	**Split into discussion groups**
14:00 – 15:00	*Group discussions*
15:00 – 15:15	Fika
15:15 – 15:45	*Presentation of results from group discussion*
15:45 – 16:00	**Workshop closing**

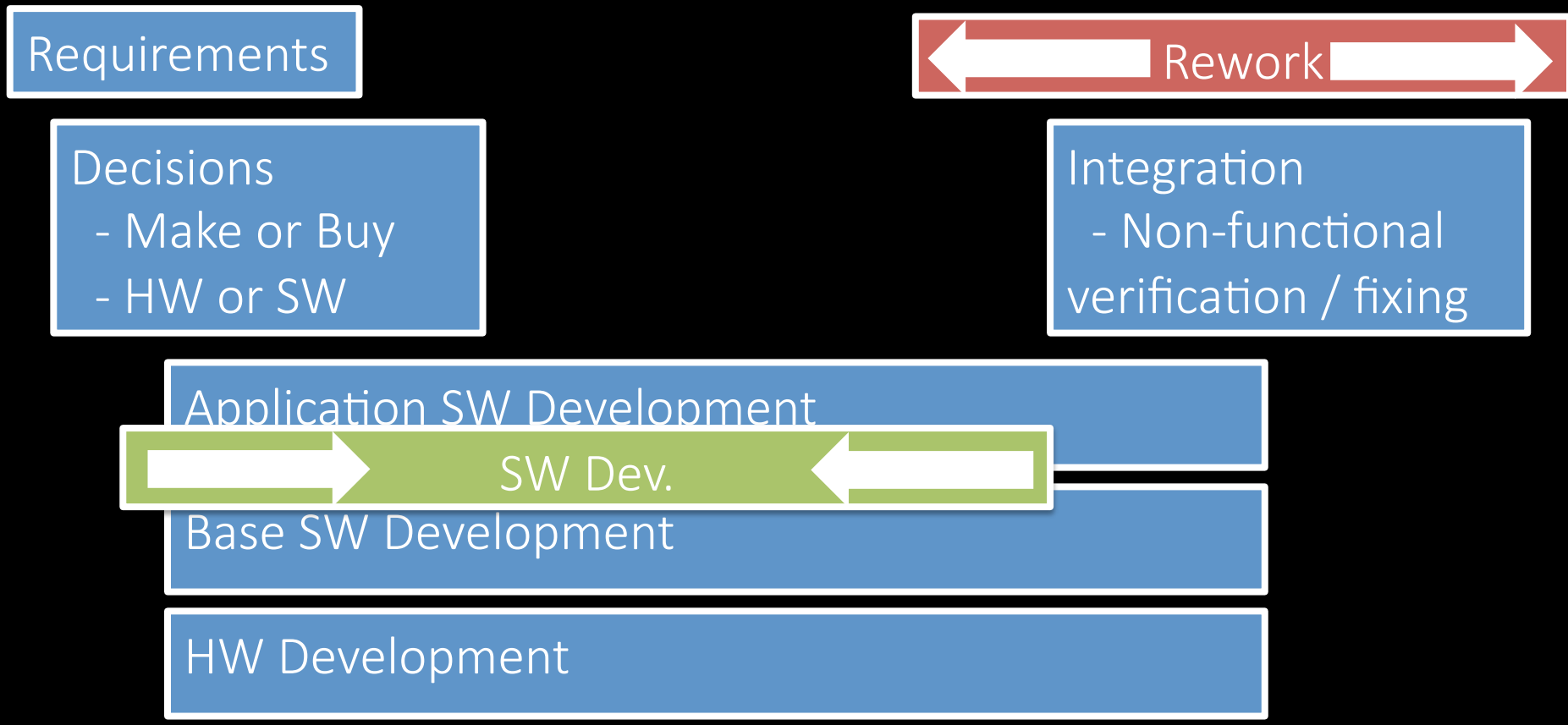
Some data on SEEC

- 93 participants (50 from industry)
  - From 19 organizations
  - 4 countries (Sweden, Denmark, Canada, France)

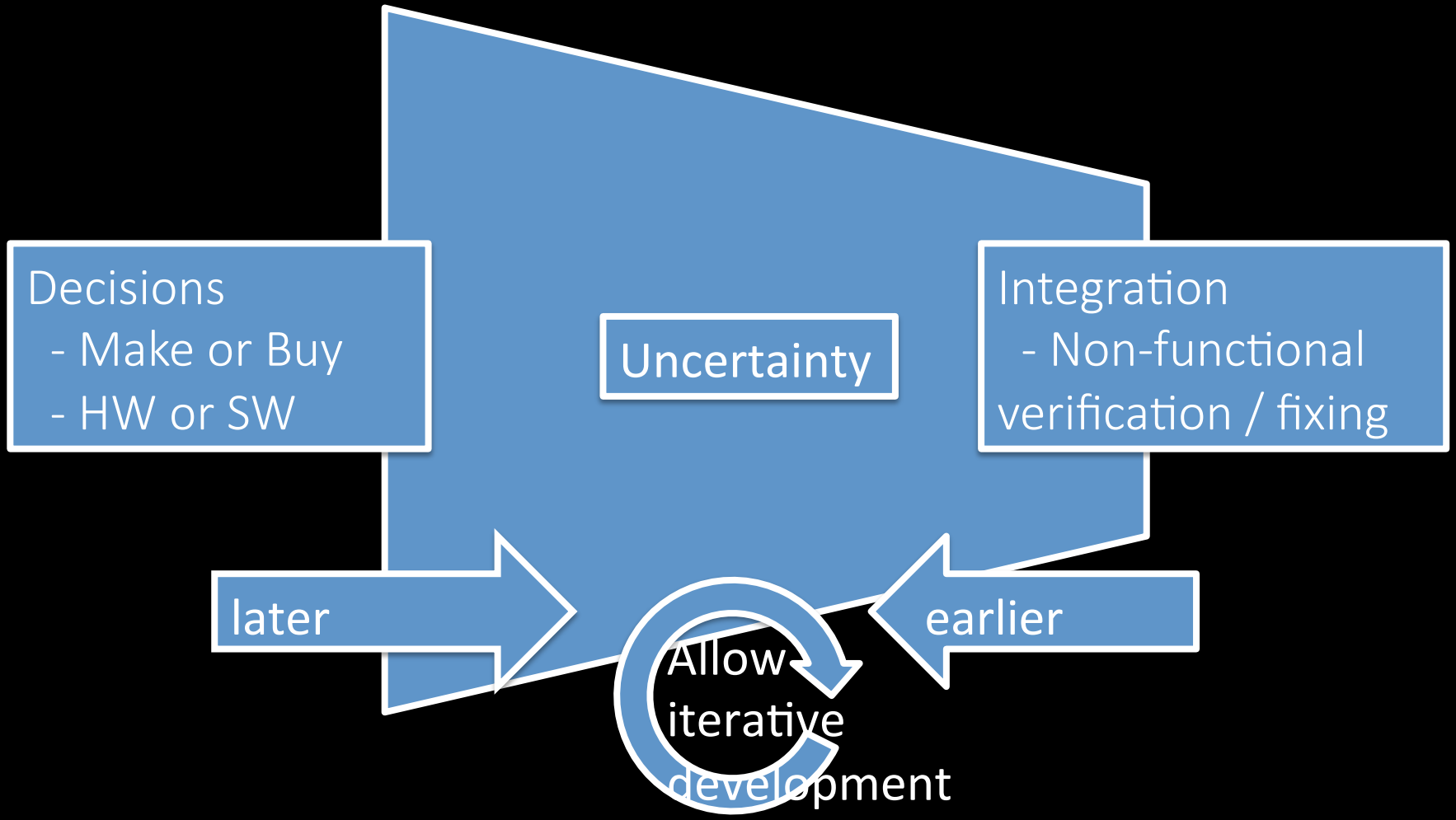
- Great stories
  - Connect people
  - Get good input into SE practice
  - Companies found great employees / Students found great jobs

  - What is yours?

Status quo: RE in a non-agile world



Vision


Decisions
- Make or Buy
- HW or SW
Uncertainty
Integration
- Non-functional verification / fixing
later
earlier
Allow iterative development

Background

- Agile Requirements Engineering
  - Breadth first
  - Just-in-time

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

State of research

Computers in Human Behavior xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Review

A systematic literature review on agile requirements engineering practices and challenges

Irum Inayat [a,*], Siti Salwah Salim [a], Sabrina Marczak [b], Maya Daneva [c], Shahaboddin Shamshirband [d,e]

[a] Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya (UM), 50603, Malaysia
[b] School of Computer Science, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) University, Rio de Janeiro, Brazil
[c] Information Science Research Group, University of Twente, Enschede, The Netherlands
[d] Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya (UM), 50603, Malaysia
[e] Department of Computer Science, Chalous Branch, Islamic Azad University (IAU), 46615-397 Chalous, Mazandaran, Iran

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Agile software development methods
Agile requirements engineering
Collaboration
Traditional requirements engineering
Systematic review

ABSTRACT

Unlike traditional software development methods, agile methods are marked by extensive collaboration, i.e. face-to-face communication. Although claimed to be beneficial, the software development community as a whole is still unfamiliar with the role of the requirements engineering practices in agile methods. The term "agile requirements engineering" is used to define the "agile way" of planning, executing and reasoning about requirements engineering activities. Moreover, not much is known about the challenges posed by collaboration-oriented agile way of dealing with requirements engineering activities. Our goal is to map the evidence available about requirements engineering practices adopted and challenges faced by agile teams in order to understand how traditional requirements engineering issues are resolved using agile requirements engineering. We conducted a systematic review of literature published between 2002 and June 2013 and identified 21 papers, that discuss agile requirements engineering. We formulated and applied specific inclusion and exclusion criteria in two distinct rounds to determine the most relevant studies for our research goal. The review identified 17 practices of agile requirements engineering, five challenges traceable to traditional requirements engineering that were overcome by agile requirements engineering, and eight challenges posed by the practice of agile requirements engineering. However, our findings suggest that agile requirements engineering as a research context needs additional attention and more empirical results are required to better understand the impact of agile requirements engineering practices e.g. dealing with non-functional requirements and self-organising teams.

Contents

* Corresponding author.
E-mail address: irum@siswa.um.edu.my (I. Inayat).

http://dx.doi.org/10.1016/j.chb.2014.10.046
0747-5632/

Please cite this article in press as: Inayat, I, et al. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.10.046

Practice

1. Face-to-face communication
2. Customer involvement
3. User stories
4. Iterative requirements
5. Requirements prioritisation

6. Change management
7. Cross-functional teams
8. Prototyping
9. Testing before coding
10. Requirements modelling
11. Requirements management
12. Review meetings and acceptance tests
13. Code refactoring
14. Shared conceptualisations
15. Pairing for requirements analysis
16. Retrospectives
17. Continuous planning

State of research:

How agile practices help

No.	Challenge	Practice	Description
1.	Communication issues (Bjarnason et al., 2011a; Carlson & Matuzic, 2010)	Frequent face-to-face meetings (Bang, 2007; Sillitti et al., 2005)	Agile RE promotes regular interaction with customer and among teams. It is the predominant method for eradicating communication gaps
		Collocated teams (Highsmith & Fowler, 2001)	Agile principles prefers collocated teams for better communication and collaboration
		Onsite customer (Cao & Ramesh, 2008; Lundh & Sandberg, 2002; Pichler et al., 2006)	Customer and development teams should be located in the same place to enhance informal communication, to enable timely feedback, to facilitate agreement, to develop ownership, and to create a sense of responsibility
		Alternate customer representations (Bjarnason et al., 2011a; Fraser, Mellon, Dunsmore, & Lundh, 2001; Hoda, Noble, & Marshall, 2011)	In industry, having business representatives to be present at the development site is expensive and impossible at times. RE alternatives such as proxy customers can serve the same purpose
		Cross-functional agile teams (Bjarnason et al., 2011a)	Cross-functional agile teams aid in the clarification and understanding of requirements
		Integrated RE process (Bjarnason et al., 2011a)	Locating RE process closer to development activities enhances developer's understanding and reduces communication lapses
2.	Overscoping (Bjarnason et al., 2011a)	One continuous scope flow (Bjarnason et al., 2011a)	In agile methods, developers receive a list of features that are constantly prioritised by the customer. Thus, the chance of having to repeat allocation in projects is reduced
		Gradual detailing (Bjarnason et al., 2011a)	Gradual detailing of requirements helps to reduce overscoping and contributes to a feasible scope
		Cross-functional teams (Bjarnason et al., 2011a)	The team can focus more on important features when sharing responsibilities and working closely together in a cross-functional structure
3.	Requirements validation (Carlson & Matuzic, 2010)	Requirements prioritisation (Racheva et al., 2010)	Customer continues with prioritisation requirements in every iteration; thus, less important requirements remain on hold
		Prototyping (Cao & Ramesh, 2008; Ramesh et al., 2010)	Prototyping helps in providing the customer with a blueprint of the product, and therefore helps in validating the requirements
4.	Requirements documentation (Bjarnason et al., 2011a)	User stories (Bjarnason et al., 2011a; Carlson & Matuzic, 2010)	User stories are precise and provide to-the-point explanation of user demands, prevent the need for maintaining long SRS documents as well as constant updating and traceability
		Face-to-face communication (Cao & Ramesh, 2008; Ramesh et al., 2010)	More face-to-face communication reduces ambiguities and the need for maintaining long documents
5.	Rare customer involvement (Carlson & Matuzic, 2010)	Requirements prioritisation by the customer (Racheva et al., 20	Prioritisation of the requirements for all iterations ensures, to a large extent that the customer goals will be met

Go to page 14

State of research:

Challenges in agile RE

Challenge	Description	Impact	Solutions
Minimal documentation (Cao & Ramesh, 2008)	User stories and product backlogs are the only documents in agile methods (Zhu, 2009)	Traceability issues (Zhu, 2009)	
Customer availability (Ramesh et al., 2010)	Availability of customer for requirements negotiation, clarification and feedback	Increase in rework	Surrogate customers (Ramesh et al., 2010)
Inappropriate architecture (Ramesh et al., 2010)	Inadequate infrastructure can cause problems during later project stages	Increase in cost	Code refactoring (Berry, 2002)
Budget and time estimation (Cao & Ramesh, 2008)	Initial estimates of time and cost are changed substantially by a change in requirements in subsequent stages	Project delays	Frequent communication
		Over-budgeting	Accurate modelling of user story
Neglecting non-functional requirements (NRFs)	User stories only satisfy system/product features	System security, usability, performance at stake	NRF modelling approach (Farid & Mitropoulos, 2012b). The NORMATIC tool (Farid & Mitropoulos, 2012a)
Customer inability and agreement (Daneva et al., 2013; Ramesh et al., 2010)	Incomplete domain knowledge and in consensus among customer groups	Increase in rework	Creation of delivery stories to accompany user stories (Daneva et al., 2013)
		Increase in cost	Frequent communication Iterative RE (Ramesh et al., 2010)
Contractual limitations (Cao & Ramesh, 2008)	Fixed-price contracts do not allow changes	Increase in cost	
Requirements change and its evaluation	To find the consequences of requirements change	Increase in work delay	RE-KOMBINE framework (Ernst et al., 2013)

Requirements flow in the automotive ecosystem

Experiences with Transitioning to Lean Requirements Engineering at Volvo Car Group

Ulf Eliasson*, Rogardt Heldal†, Eric Knauss†, Patrizio Pelliccione†

Challenge	Practices	Recommendations
Balancing under-specification and over-specification of requirements	• *Personal network* • *Oral communication* • *Assumptions*	• Networking • On demand / Just-in-time RE • Infrastructure for communication and feedback
Synchronizing cross-function and cross-system requirements	• Personal network	• Continuous integration and deployment
Friction for changing requirements increases over time	• Workarounds	• Defer commitment
Avoid premature commitment	• Workarounds • Oral communication	• On demand / Just-in-time RE • Rationale as by-product

Feedback on Reqts in the automotive ecosystem

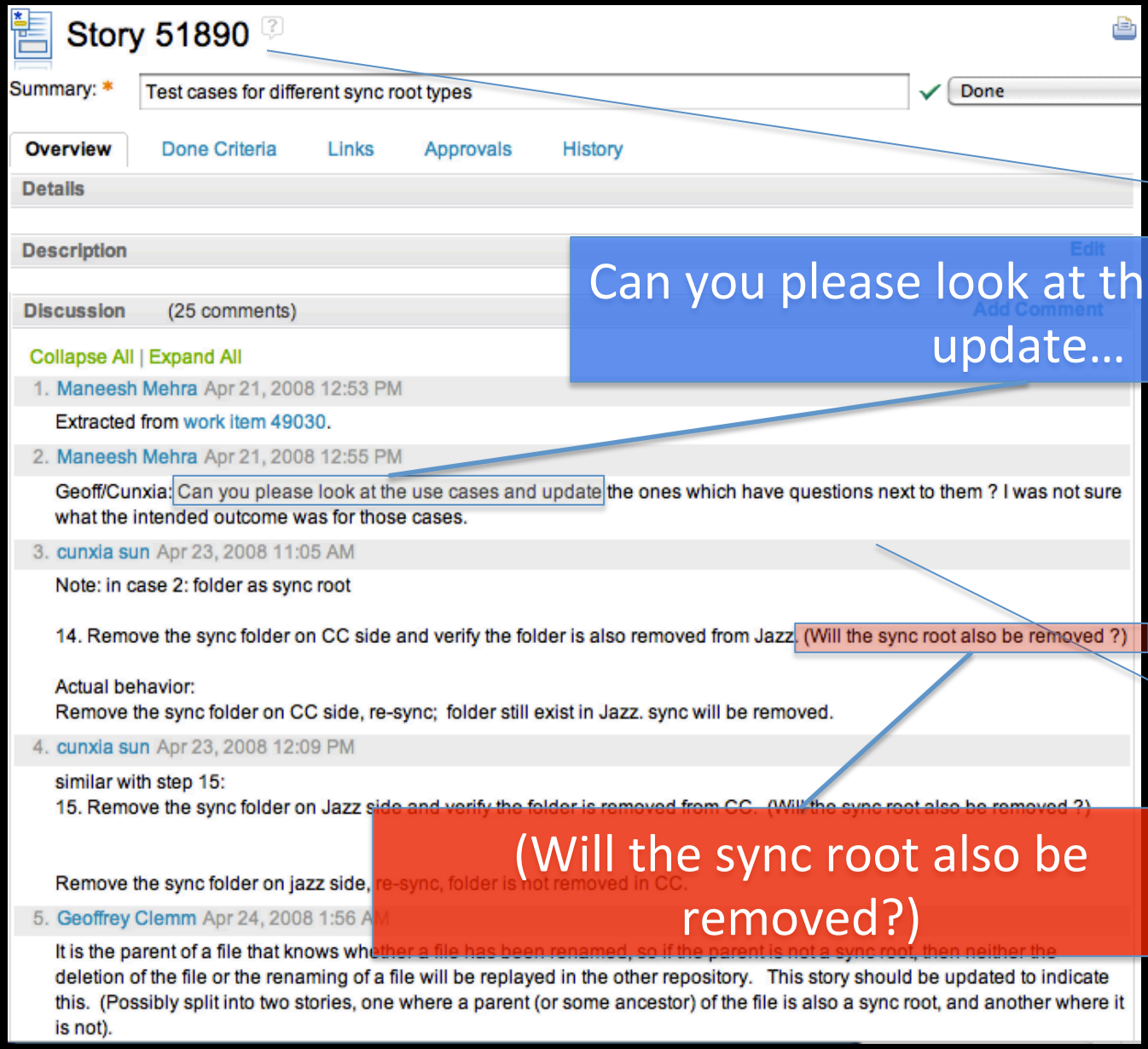
Experiences with Transitioning to Lean Requirements Engineering at Volvo Car Group

Ulf Eliasson*, Rogardt Heldal†, Eric Knauss†, Patrizio Pelliccione†

Challenge	Practices	Recommendations
Slow feedback cycle on requirements	• *Personal network* • *Oral communication*	• Short iterations / agile • Virtual verification early
Balancing the need for oral communication and thorough documentation of requirements	• *Personal network* • *Oral communication*	• On demand / Just-in-time RE • Rationale as by-product
Find the right person for getting or giving feedback or information	• Personal network • Expert seeking	• Facilitate networking within company and supply-chain
Sufficiently fast supplier interaction	• Personal network	• New business cases / opportunities for suppliers

Example: User stories in IBM RTC

Type	Id	Summary	Status	Priority	Story Points
	249878	Develop RPP prototype to work with RQM server proxy example	Implementing		Medium (5)
	213426	Create ClearQuest-CLM Integration Cook book	Ready to Implement		Unassigned
	261347	SCM needs resources for compiling native binaries to be included in the RTC build	Implementing		Unassigned
	255301	Prototype end to end gap scenario in Visual Studio client	Implementing		Extra Large (13)
	261375	Write a preliminary front end client native launcher for AIX platform	Ready to Test		Medium (5)
	253416	Create IES 4.2 / P2 based RTC client offering that shell-shares with Rxx products	Implementing		Unassigned
	259647	Test Install of IES clients (4.2.2 and 3.6.2) from CLM and RTC launchpads	Testing		Small (3)
	250410	Write a preliminary front end client native launcher for Windows, Mac & some Unix platforms	Ready to Test		Large (8)
	244355	As a user, I want to be able to resolve port related conflicts	Ready to Implement		Unassigned
	246735	HPI: Request build from Hudson/Jenkins and report back to RTC build	Implementing		Medium (5)
	164045	history import should import all file/folder versions for base ClearCase	Invalid		Gigantic (200)
	251965	As a user, I want to be able to deal with conflicts created by an accept while porting	Ready to Implement		Unassigned
	236727	Support for in-place script editing for Script Provider Aspectlets	Ready to Test		Unassigned



Story 51890

Summary: * Test cases for different sync root types

Done

Overview | Done Criteria | Links | Approvals | History

Details

Description

Discussion (25 comments)

Collapse All | Expand All

1. Maneesh Mehra Apr 21, 2008 12:53 PM

Extracted from work item 49030.

2. Maneesh Mehra Apr 21, 2008 12:55 PM

Geoff/Cunxia: Can you please look at the use cases and update the ones which have questions next to them ? I was not sure what the intended outcome was for those cases.

3. cunxia sun Apr 23, 2008 11:05 AM

Note: in case 2: folder as sync root

14. Remove the sync folder on CC side and verify the folder is also removed from Jazz. (Will the sync root also be removed ?)

Actual behavior:
Remove the sync folder on CC side, re-sync; folder still exist in Jazz. sync will be removed.

4. cunxia sun Apr 23, 2008 12:09 PM

similar with step 15:
15. Remove the sync folder on Jazz side and verify the folder is removed from CC. (Will the sync root also be removed ?)

Remove the sync folder on jazz side, re-sync, folder is not removed in CC.

5. Geoffrey Clemm Apr 24, 2008 1:56 AM

It is the parent of a file that knows whether a file has been renamed, so if the parent is not a sync root, then neither the deletion of the file or the renaming of a file will be replayed in the other repository. This story should be updated to indicate this. (Possibly split into two stories, one where a parent (or some ancestor) of the file is also a sync root, and another where it is not).

Look at user stories

Can you please look at the use case and update…

Analyze discussion

(Will the sync root also be removed?)

Story 51890

Summary: * Test cases for different sync root types — Done

Overview | Done Criteria | Links | Approvals | History

Details

Description

Discussion (25 comments)

Collapse All | Expand All

Extracted from work item 4903

2. Maneesh Mehra Apr 21, 2008

Geoff/Cunxia: Can you please ... what the intended outcome wa...

3. cunxia sun Apr 23, 2008 11:05

Note: in case 2: folder as sync ...

14. Remove the sync folder on ...

Actual behavior:
Remove the sync folder on CC side, re-sync; folder still exist in Jazz. sync will be removed.

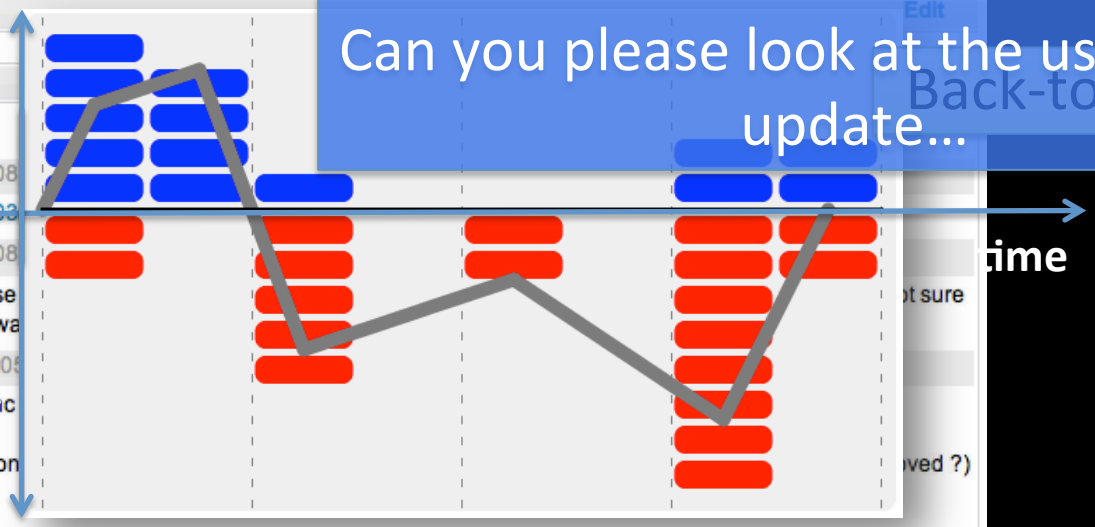
4. cunxia sun Apr 23, 2008 12:09 PM

similar with step 15:
15. Remove the sync folder on Jazz side and verify the folder is removed from CC. (Will the sync root also be removed ?)

Remove the sync folder on jazz side, re-sync, folder is not removed in CC.

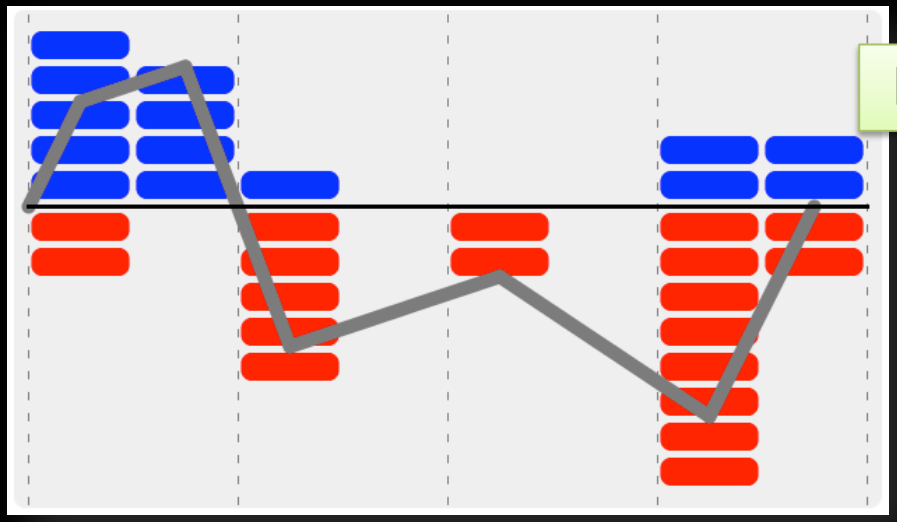
5. Geoffrey Clemm Apr 24, 2008 1:56 AM

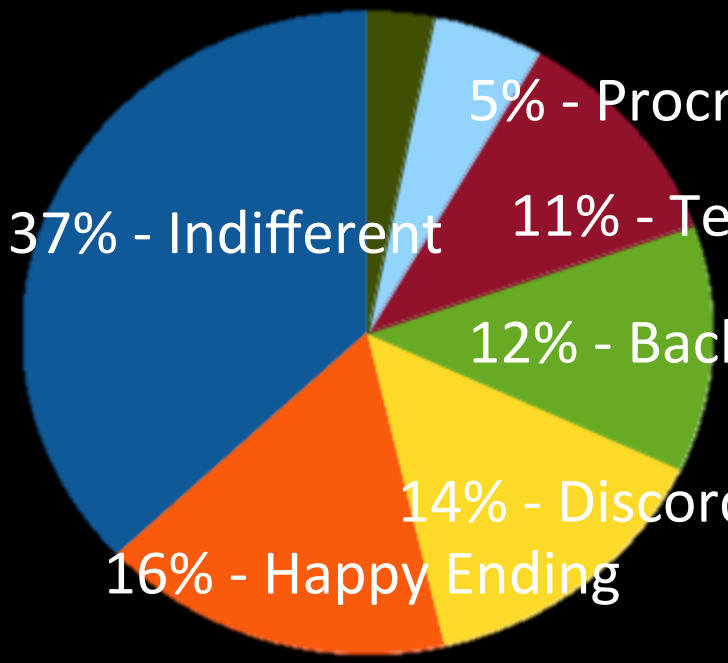
It is the parent of a file that knows whether a file has been renamed, so if the parent is not a sync root, then neither the deletion of the file or the renaming of a file will be replayed in the other repository. This story should be updated to indicate this. (Possibly split into two stories, one where a parent (or some ancestor) of the file is also a sync root, and another where it is not).



Can you please look at the use case and update…

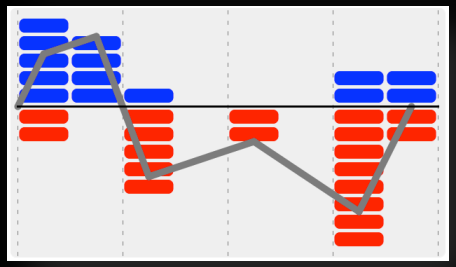
(Will the sync root also be removed?)


Back-to-draft

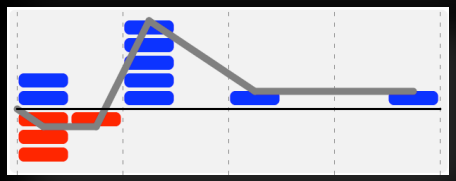


Indifferent

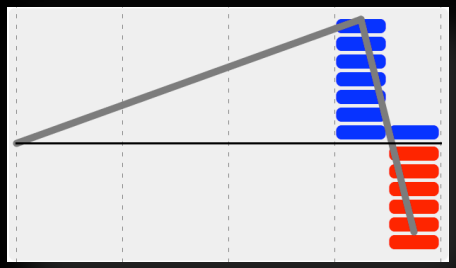
Back-to-draft



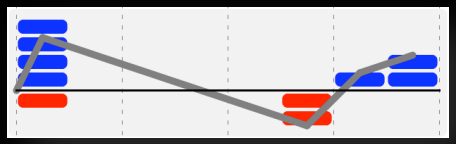
Textbook-example



Procrastination

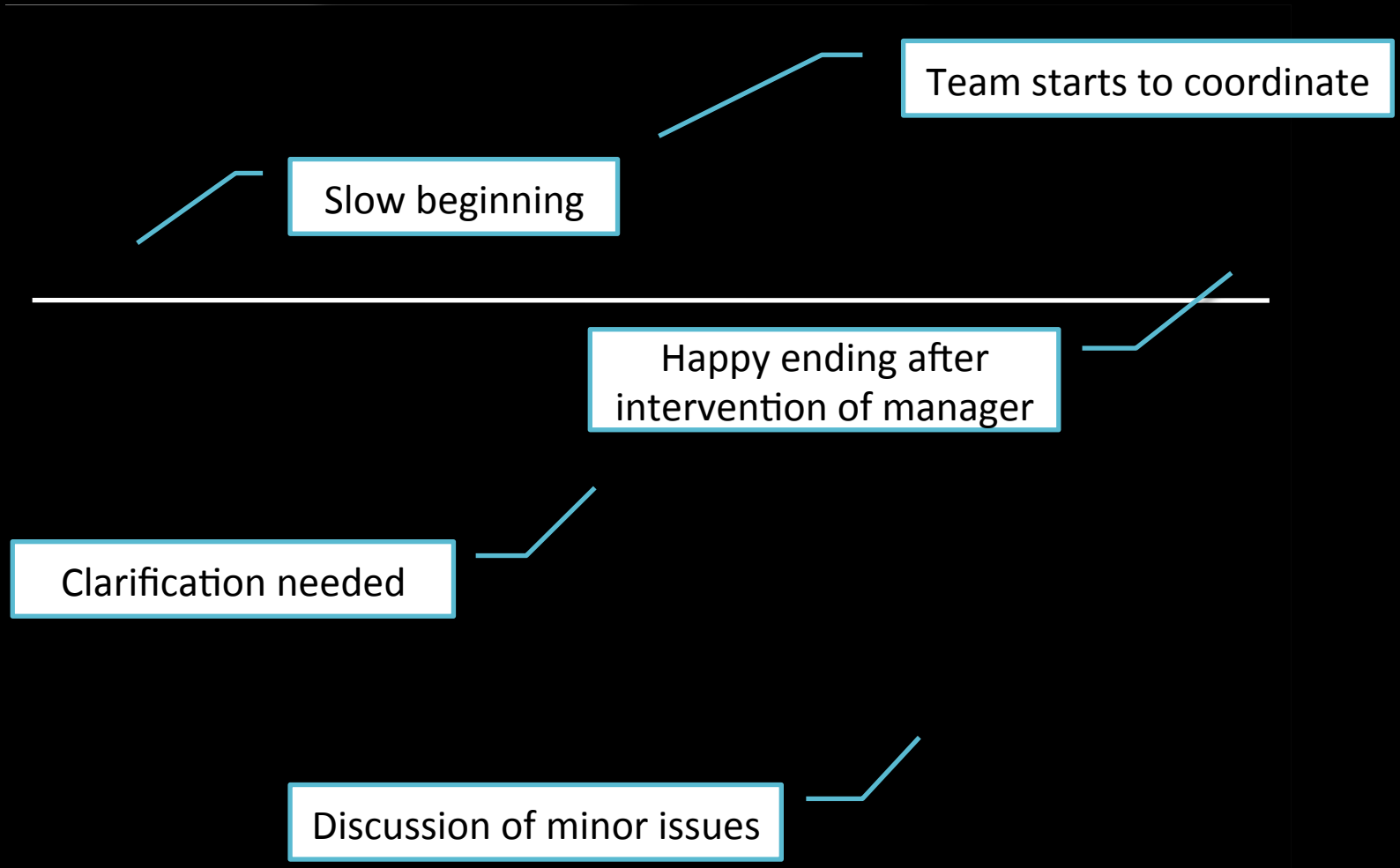


Happy-ending

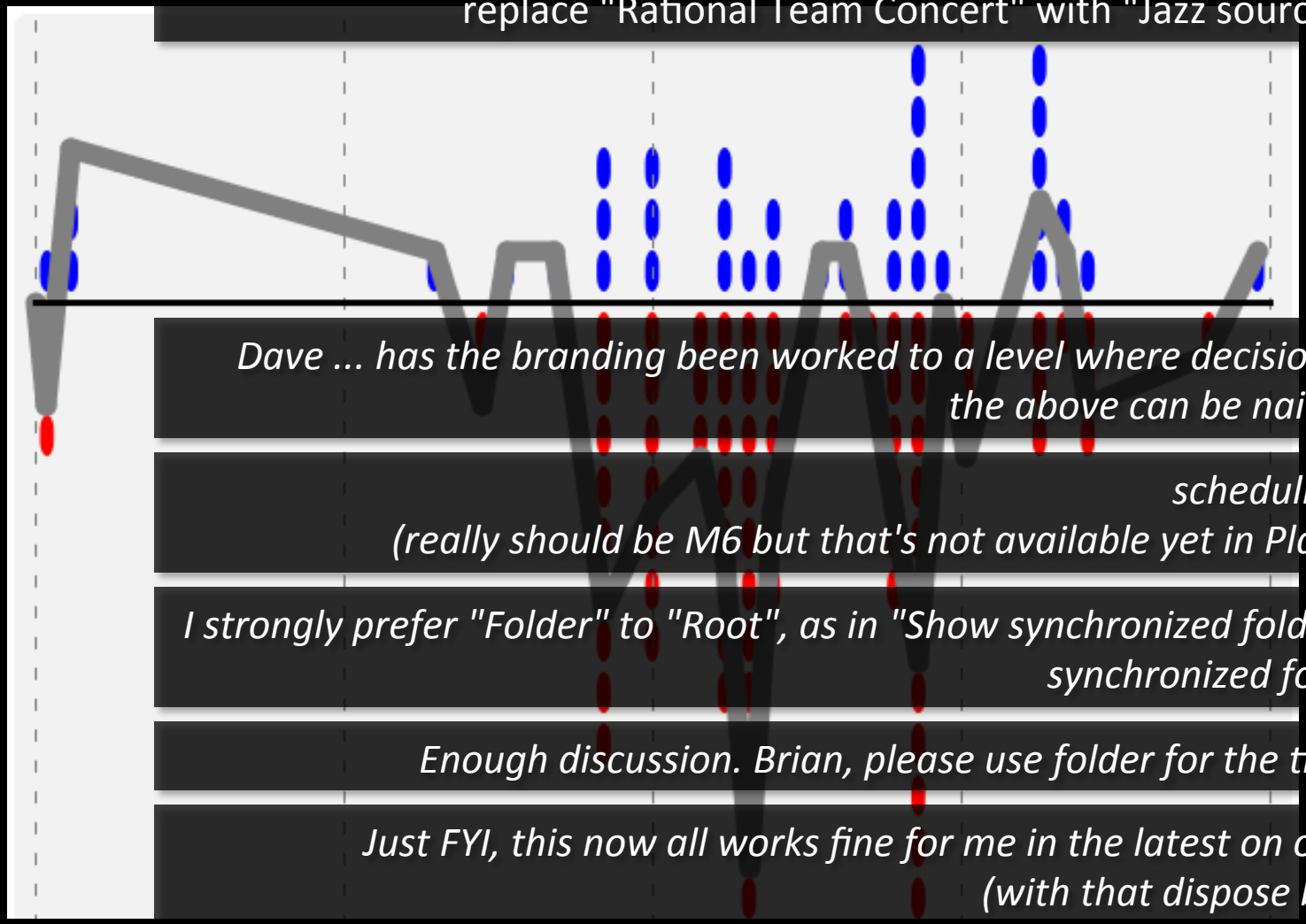


Discordant

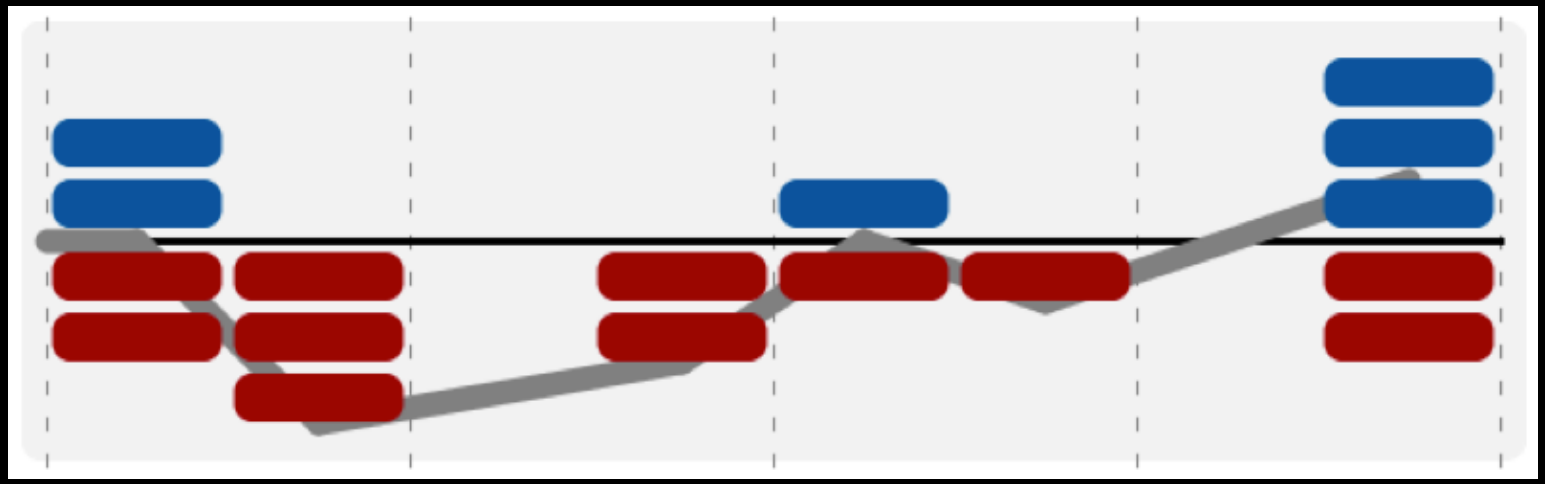
Example


Team starts to coordinate
Slow beginning
Happy ending after intervention of manager
Clarification needed
Discussion of minor issues

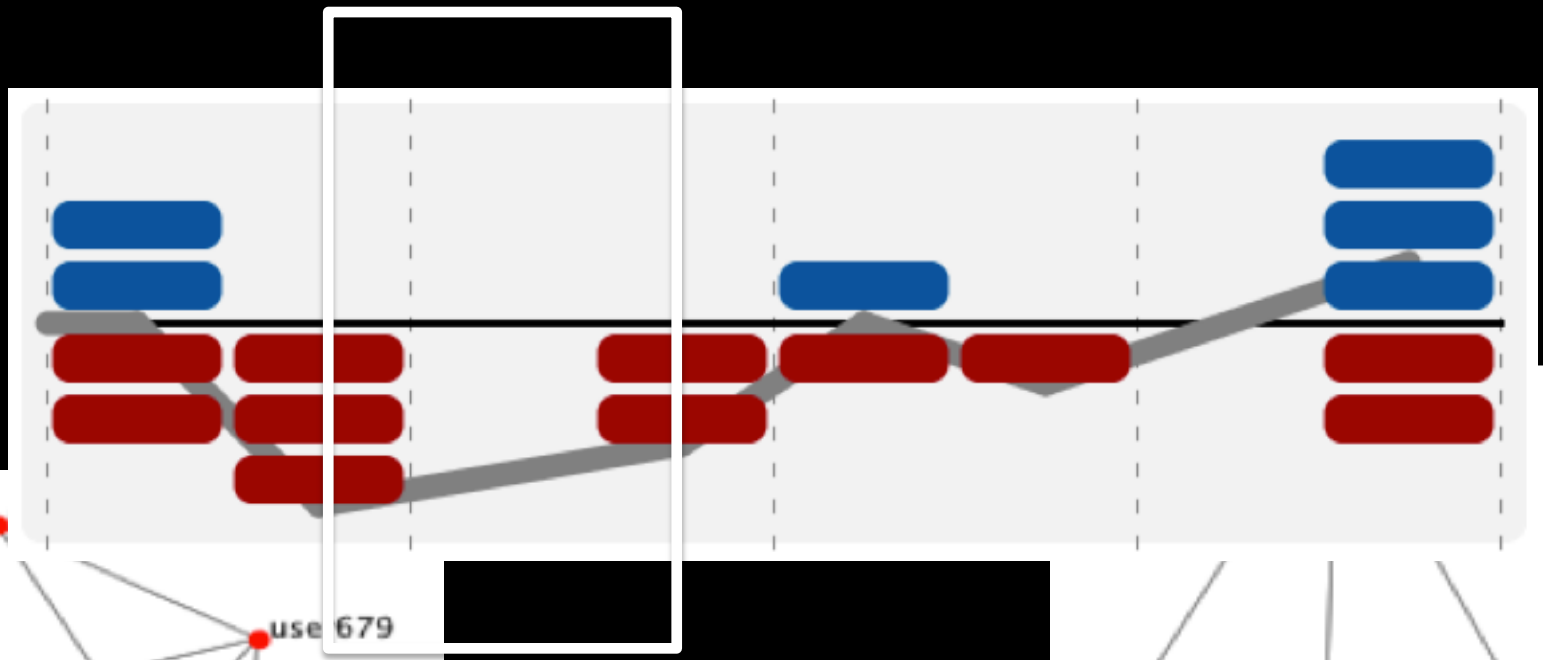
Example

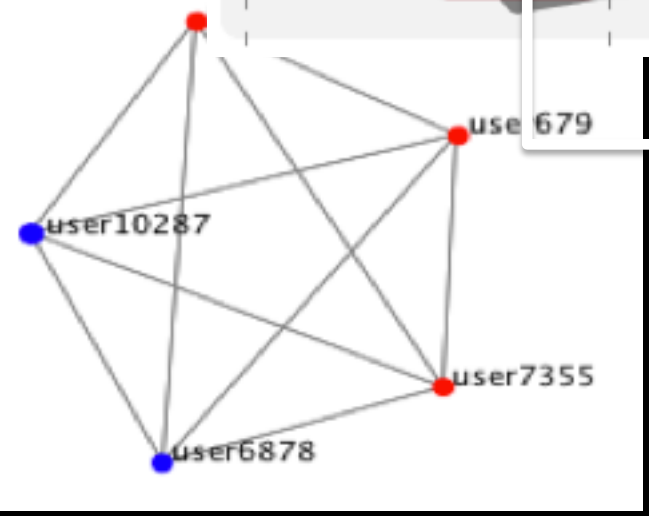

replace "Rational Team Concert" with "Jazz source control"
Dave ... has the branding been worked to a level where decisions such as the above can be nailed down?
scheduling for M5
(really should be M6 but that's not available yet in Planned For)
I strongly prefer "Folder" to "Root", as in "Show synchronized folders", "Add synchronized folder", etc.
Enough discussion. Brian, please use folder for the time being.
Just FYI, this now all works fine for me in the latest on our stream (with that dispose bug fixed).

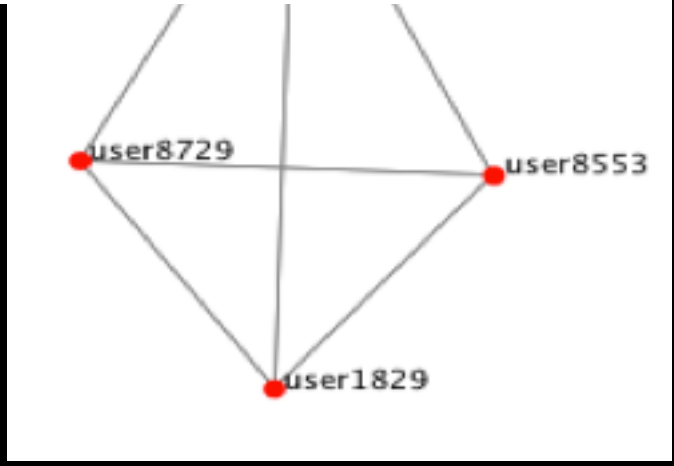
Are there problematic requirements?



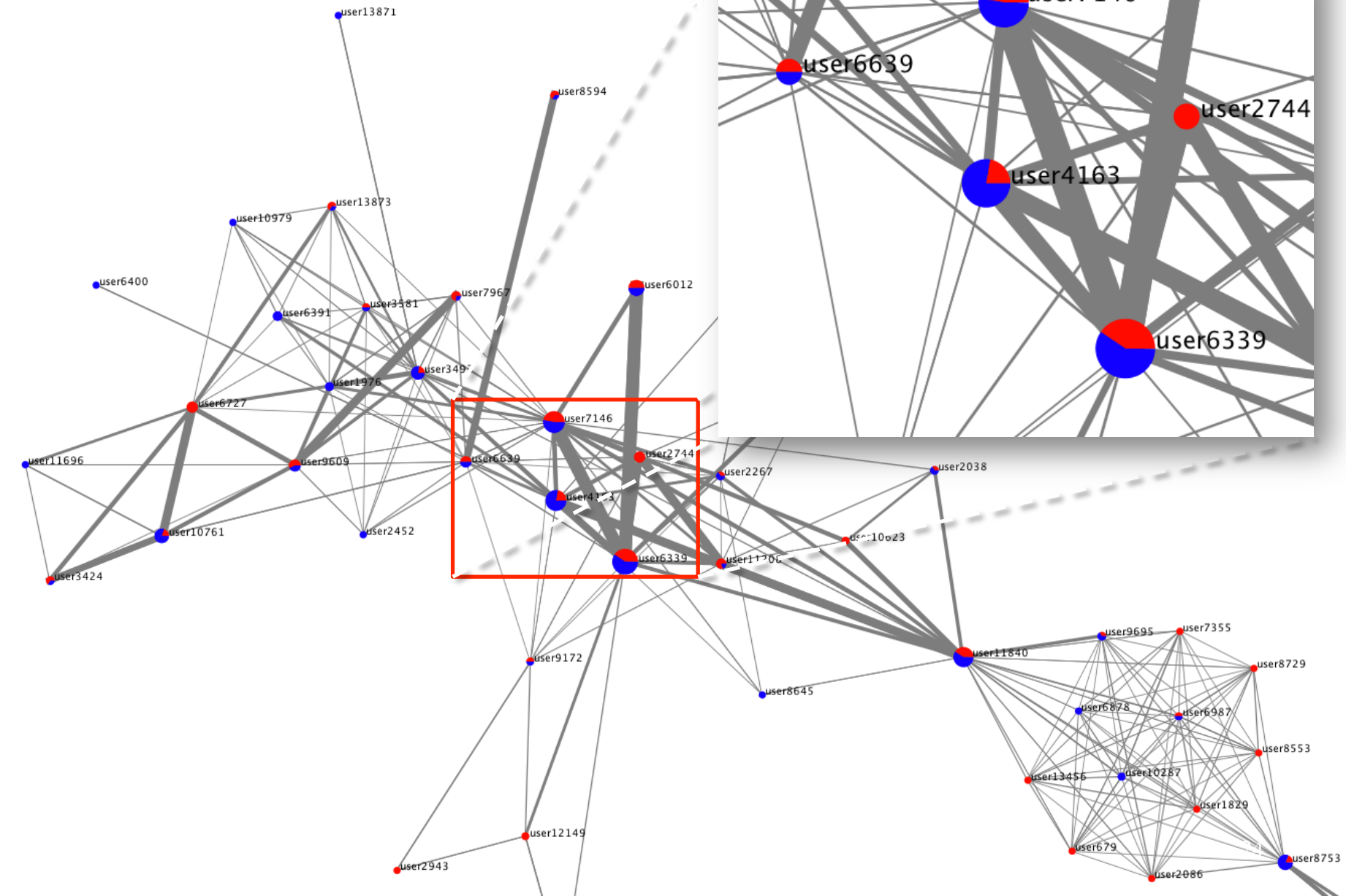
Are there comm. Breakdowns?







Who is knowledgea



CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Baldvin

SUCCESSFUL WITHOUT REQUIREMENTS

http://seecgot.wordpress.com

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

All

GROUP DISCUSSION

Discussion Topics

Aspects

- Agile RE practices
- Oral/informal reqts.
- Interplay of RE and Tests
- Upfront analysis and prioritization

Challenges

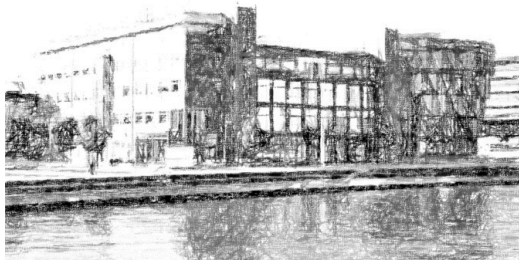
- Waterfall context (e.g. because of HW dev.)
- Complex value-chain (e.g. reqts. to suppliers
- Safety regulations

1. Problems/ Drawbacks?

2. (Promised) Benefit?

3. How should/could it work?

Workshop Closing



SEEC | Eric Knauss
http://seecgot.wordpress.com

- Quick round
  - One thing you liked
  - One thing you wish for
  - One thing you want to see in SEEC next year

Workshop Closing

Gothenburg, Sweden, March 14-17, 2016

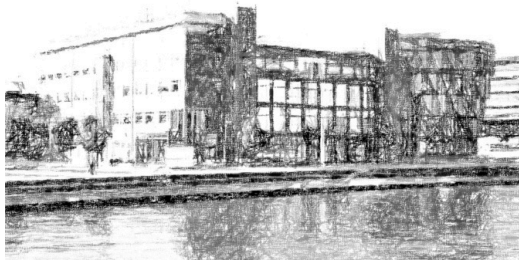
REFSQ 2016

22nd Intl. Working Conference on Requirements Engineering: Foundation for Software Quality

The **European elite in Requirements Engineering** will come to Sweden and Gothenburg in March of 2016. More than **100 people**, both from industry and academia, will get together for four days to **exchange experiences, improve how to work with requirements, business analysis and product management and have a great time**.

- One thing that you like
  - good discussion
  - time for discussion
  - Breadth of participants (students, academics, industry)
  - Conflicting ideas
  - Bringing together this mix
  - 3h timeslot
- One thing that you wish for
  - Waterfall not opposite of agile
  - Initiate collaborations
  - More small companies in academic focus
  - Even more breadth
  - More quality assurance to relate agile and modeling
  - More focused discussion, come to results (+1)
  - Fika too short
- Next
  - tool interoperabilty
  - open source tooling
  - **RE and agile**
  - **Alignment (e.g. between RE & Testing**, models & dev.)
  - Qualities (beyond code quality)
  - Interoperability (tooling, assumptions on gaps
  - Productive Large-Scale works
  - Present from collaborative projects
  - From customer needs to models
  - Metrics in industry

Workshop Closing



SEEC | Eric Knauss
http://seecgot.wordpress.com

- Tack så mycket | Thank you | Vielen, vielen Dank!
- Feedback: welcome!
- New topics: welcome!
- You: welcome!

Thanks!
Eric Knauss
eric.knauss@cse.gu.se
@oerich
oerich.wordpress.com

For example to the next SEEC Workshop!

CHALMERS

UNIVERSITY OF TECHNOLOGY

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Software Engineering Experience Circle

16th Mar 2015, RE in agile context

brought to you by:

Division of Software Engineering, Lindholmen Science Park, and Software Center