

Learning from Experiences with Shaping Requirements through Decision Documentation

GI Fachgruppentreffen, 27.-28.11.2014, Dortmund

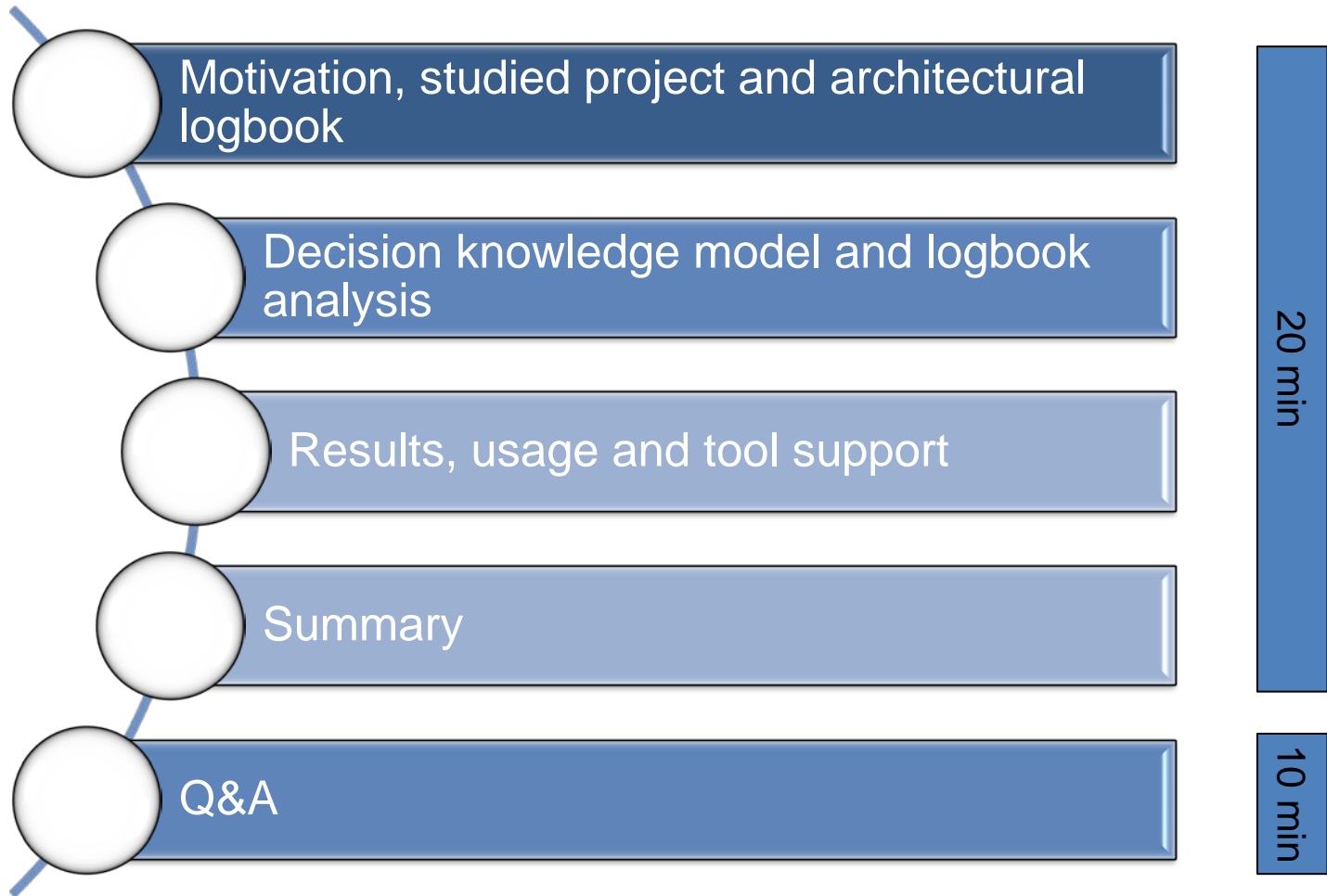
**Tom-Michael Hesse, Christian Kücherer,
Barbara Paech**

Institute of Computer Science
Chair of Software Engineering
Im Neuenheimer Feld 326
69120 Heidelberg, Germany

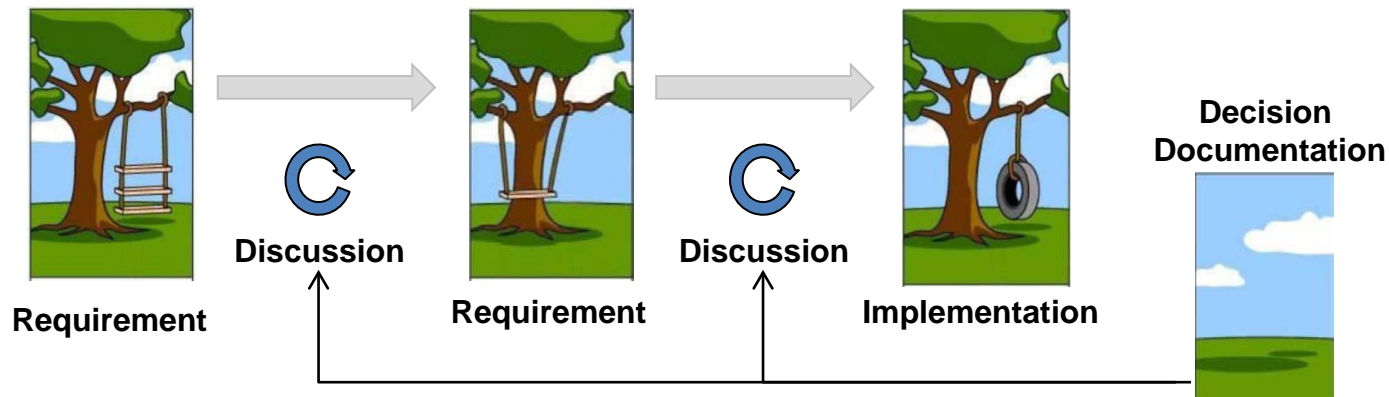
<http://se.ifi.uni-heidelberg.de>

{hesse,kuecherer,paech}@informatik.uni-heidelberg.de





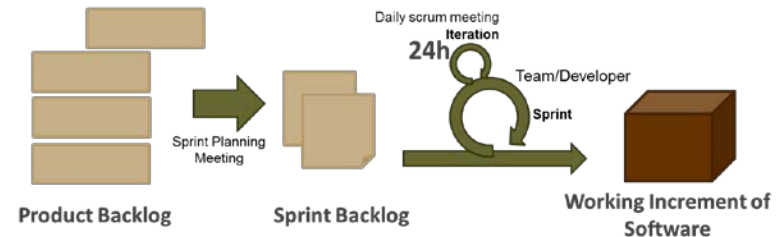
- Typical situation: All team members are responsible for requirements (e.g., in agile development teams)



- But: Decisions should be documented for future comprehension and exploitation
 - Reports on experiences with decision documentation in agile projects are rare

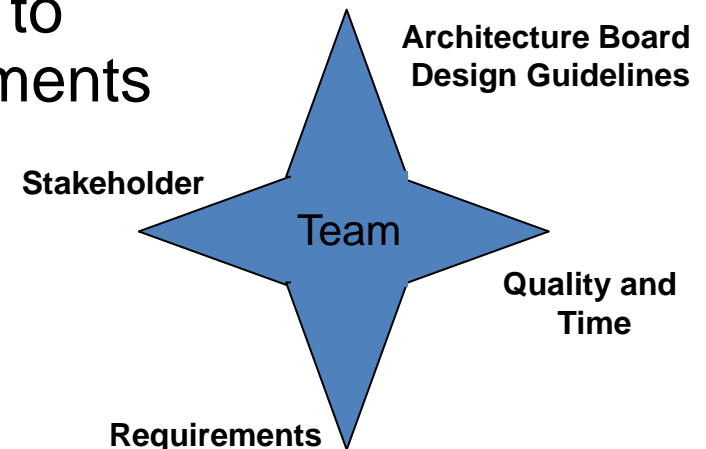
Background: The Studied Project

- Agile project using Scrum
- 3 years duration
- Team size: ~12 person, junior team
- Development of an interactive web-front-end (Ajax) for a content management system (CMS)



Experiences

- Discussions needed by developers to understand and implement requirements raised decisions, but:
 - Many **reoccurring** discussions
 - **No clear** or forgotten decisions
 - **No binding character** of decisions



Background: Architectural Logbook

- Documentation of
 - Decision and issues
 - Discussions with impact on team
- Decisions are **related** to and **motivated** by Requirements
- Discussion supported **understanding** of requirements and **shaped** them
- Team accessible Word document on fileserver
- Entries added by architect and sent by E-Mail

Project
Architecture-
Decision Logbook

Introduction

Decision 1
<details here>

....
Decision n
<details here>

Analysis of Metadata

- **15 pages** DIN A4
- **40** documented decisions and issues
- Artifacts: UML diagrams, code, tables, documents (presentations), figures, links to (development-)tasks

Title: *Removal of Duplicates in Hit-list*

Date: 22.11.2013, Product Owner Alice, Developer John, Architect Zoe

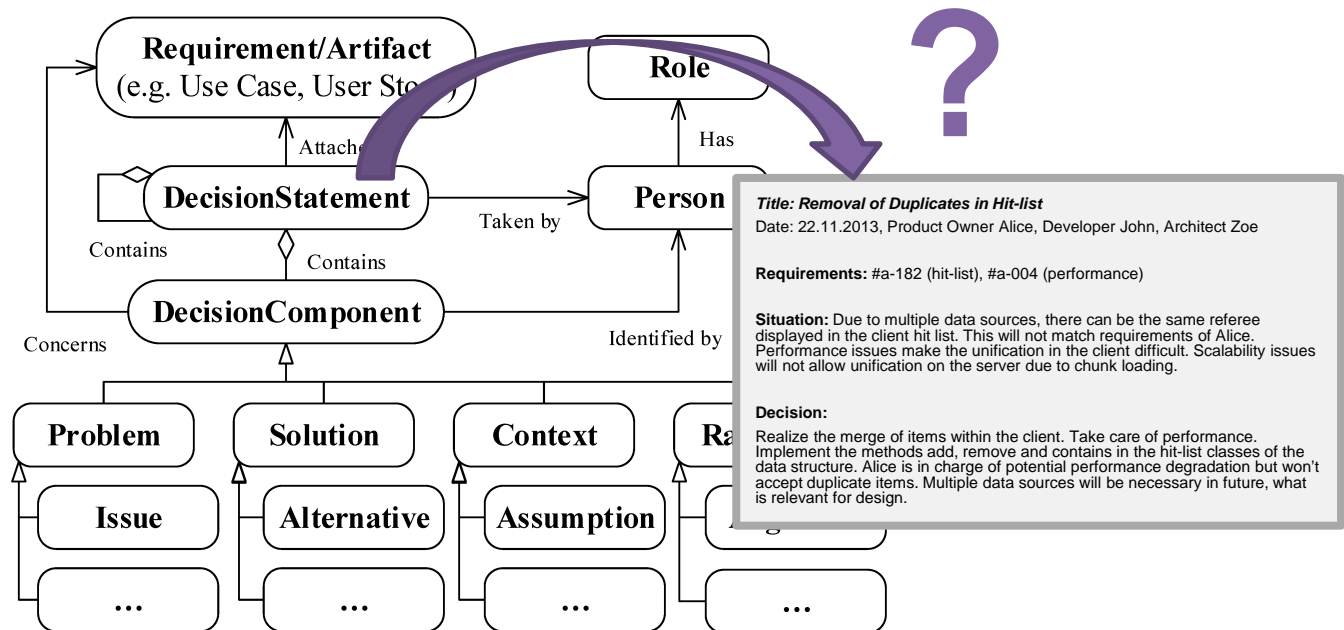
Requirements: #a-182 (hit-list), #a-004 (performance)

Situation: Due to multiple data sources, there can be the same referee displayed in the client hit list. This will not match requirements of Alice. Performance issues make the unification in the client difficult. Scalability issues will not allow unification on the server due to chunk loading.

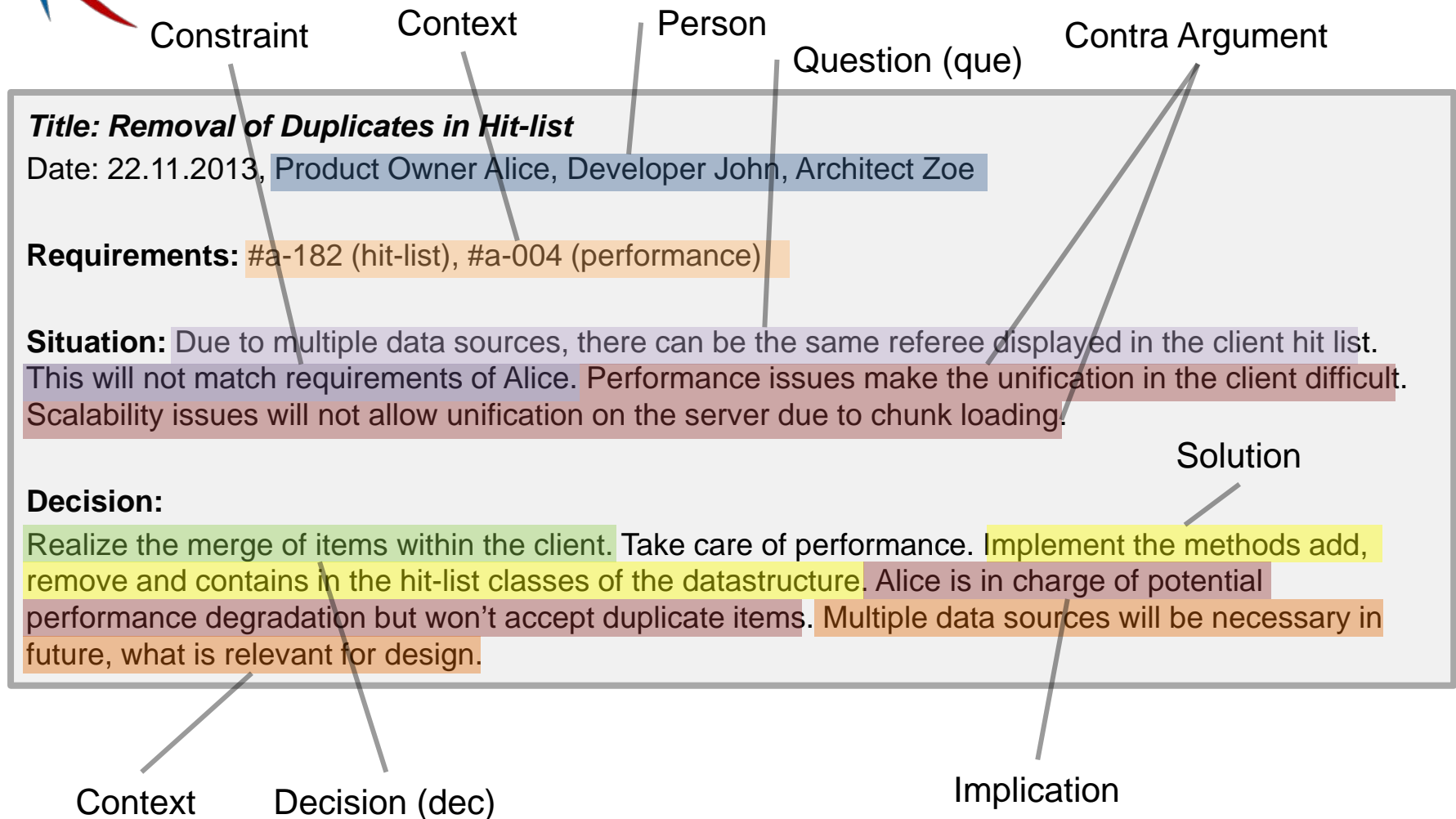
Decision:

Realize the merge of items within the client. Take care of performance. Implement the methods add, remove and contains in the hit-list classes of the data structure. Alice is in charge of potential performance degradation but won't accept duplicate items. Multiple data sources will be necessary in future, what is relevant for design.

- Strategies for decision-making:
 - **Rational** – Uncover and assess all alternatives before deciding
 - **Naturalistic** – Follow one solution until changes are required
- Decision Knowledge Model:



Study: Which Knowledge Was Documented?

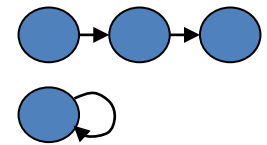


Results: Decision-related Findings

Entry Nr.	Person	Dec	Que	Goal	Sol	Claim	Cn	Asp	Cst	Imp	Argpro	Argcon	Argntrl	Sum
1		1	1		4	1	1		2	1	4	1	1	17
2	2	2		1	1		1	1	2	1	3			14
3	1	1	1		1		1		2					7
4	2	1	3		1		1							8

Findings

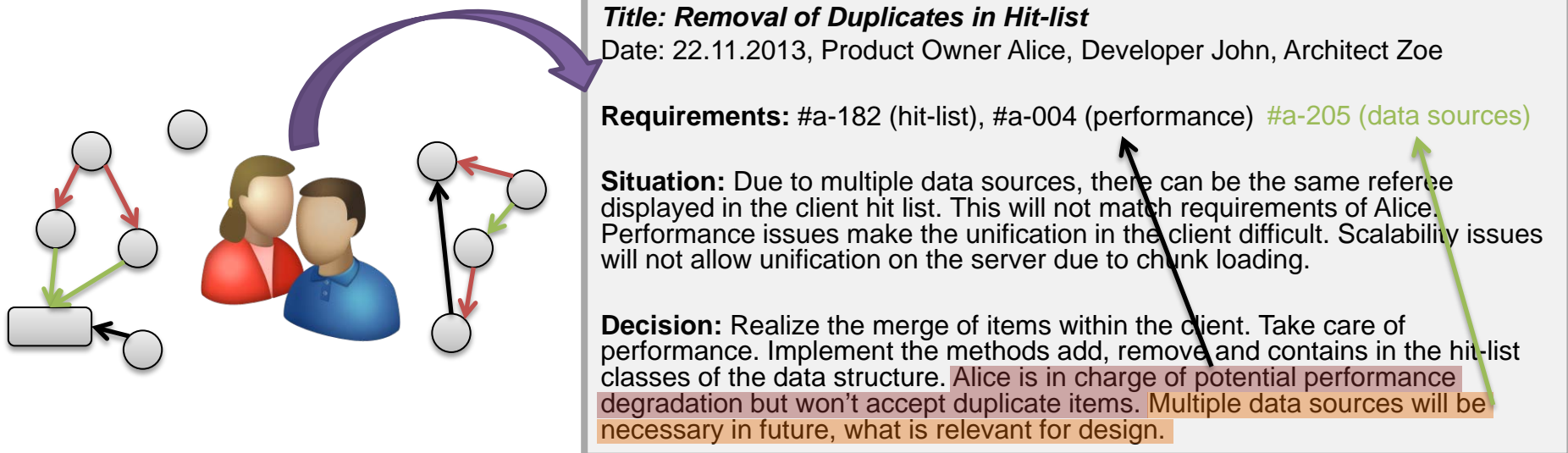
- *Decision* and *Solution* often described in the same sentence (19/40)
 - 4 decisions had follow ups
 - 1 decision had 3 recurrences
 - 3 decisions did not contain a clear decision statement
 - 6 decisions had no question, issue or goal
 - Only 1 decision has no solution
-
- 25 decisions contained one or more **arguments**
 - 18 decisions contained one or more **constraints**
- Observation: Influential elements in discussions



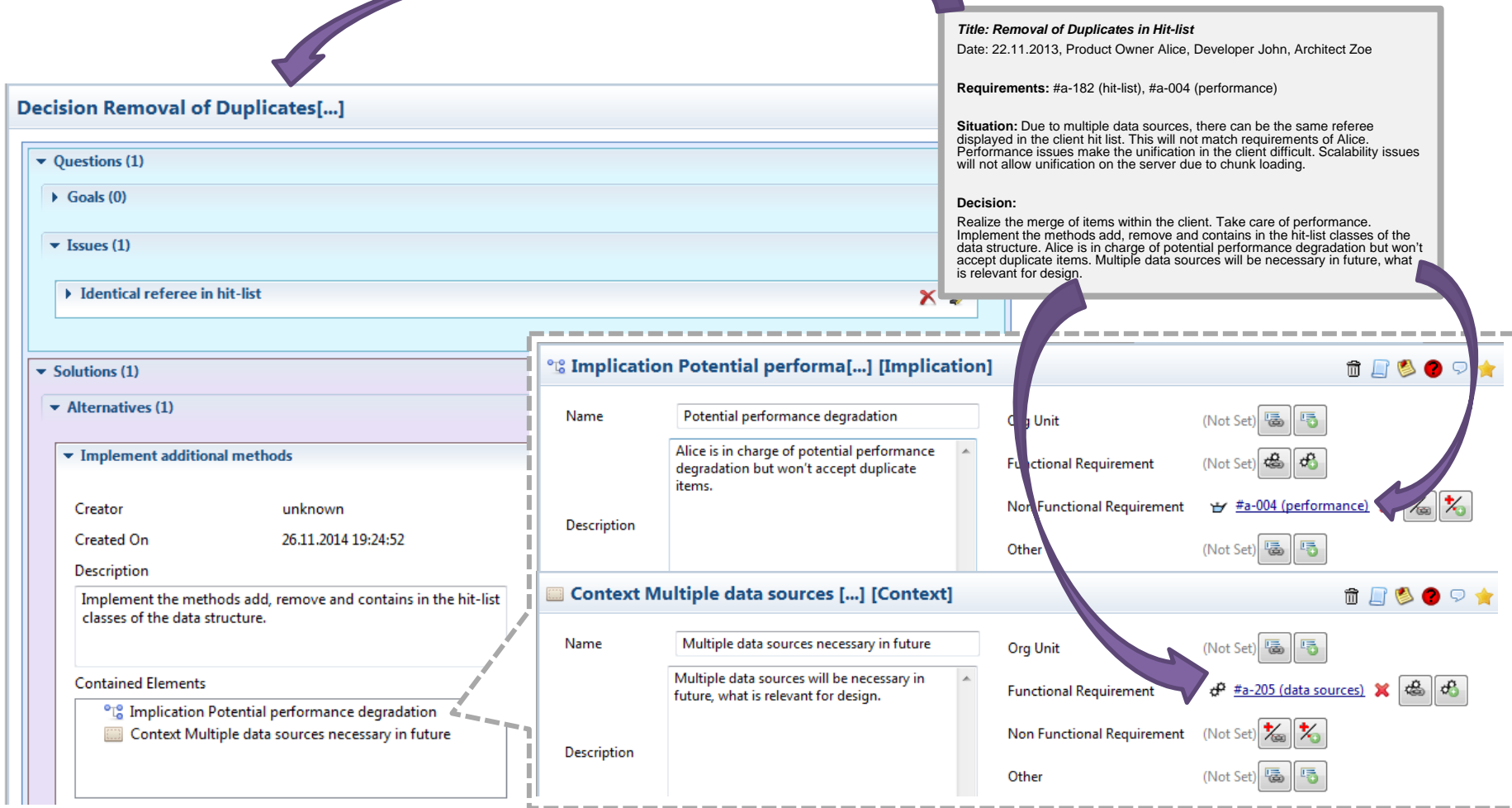
32	1	1												5
33	2	1			1		1							5
34														0
35	1	1	1		1		1				1			6
36	2		1		2		2		2		1			10
37	2	1	1		2		1		1			2		10
38	2	4			4				1	1				12
38a			1											1
39	2	1	1		3		2					2		11

Support Shaping Requirements through Documented Decisions

- Discussions indicate that **decisions were** made **naturalistic**
 - Focus on one solution, which might be challenged in future
 - Example: Decision with 4 solutions and 6 arguments is difficult to comprehend without explicit representation of links
 - Important to know: Which knowledge concerns which requirement?
→ Capture knowledge and links iteratively and fine-grained



Integration in Case Tool UNICASE



Decision Removal of Duplicates[...]

- Questions (1)
 - Goals (0)
 - Issues (1)
 - Identical referee in hit-list
- Solutions (1)
 - Alternatives (1)
 - Implement additional methods
 - Creator: unknown
 - Created On: 26.11.2014 19:24:52
 - Description: Implement the methods add, remove and contains in the hit-list classes of the data structure.
 - Contained Elements
 - Implication Potential performance degradation
 - Context Multiple data sources necessary in future

Implication Potential performa[...] [Implication]

Name	Potential performance degradation	Org Unit	(Not Set)
Description	Alice is in charge of potential performance degradation but won't accept duplicate items.	Functional Requirement	(Not Set)
		Non Functional Requirement	#a-004 (performance)
		Other	(Not Set)

Context Multiple data sources [...] [Context]

Name	Multiple data sources necessary in future	Org Unit	(Not Set)
Description	Multiple data sources will be necessary in future, what is relevant for design.	Functional Requirement	#a-205 (data sources)
		Non Functional Requirement	(Not Set)
		Other	(Not Set)

Callout Box:

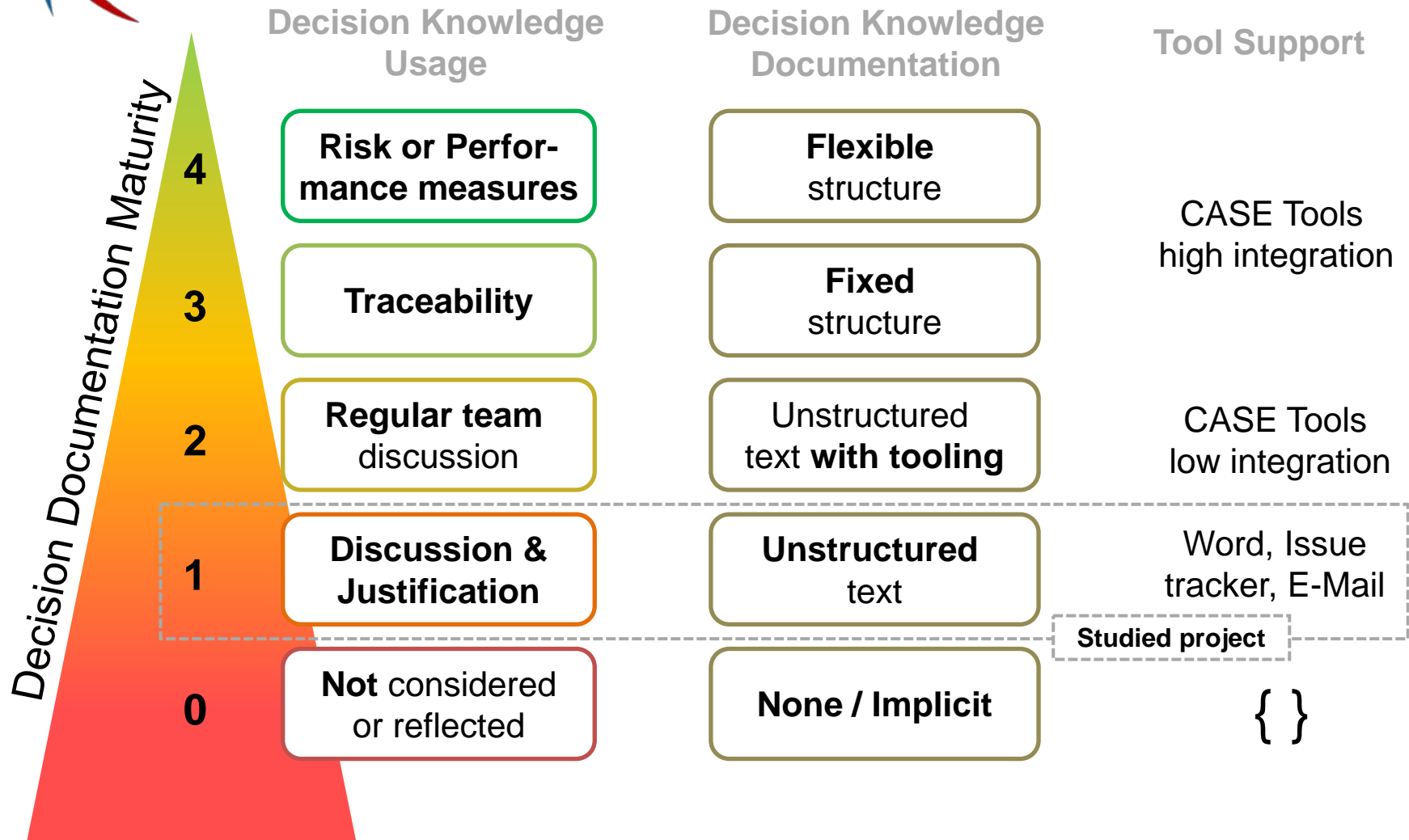
Title: Removal of Duplicates in Hit-list
Date: 22.11.2013, Product Owner Alice, Developer John, Architect Zoe

Requirements: #a-182 (hit-list), #a-004 (performance)

Situation: Due to multiple data sources, there can be the same referee displayed in the client hit list. This will not match requirements of Alice. Performance issues make the unification in the client difficult. Scalability issues will not allow unification on the server due to chunk loading.

Decision: Realize the merge of items within the client. Take care of performance. Implement the methods add, remove and contains in the hit-list classes of the data structure. Alice is in charge of potential performance degradation but won't accept duplicate items. Multiple data sources will be necessary in future, what is relevant for design.

Potential Levels of Usage and Tool Support



- Agile teams elicit and shape requirements in discussions – in particular, when making decisions

Analysis of an **architectural logbook** with 40 decisions and issues showed:

- Even when decisions are documented, **missing structures** and **tool integration** lower documentation value
- Many decisions are **solution-driven/naturalistic** and **will be challenged** in the future by new arguments
- To keep track with resulting changes in requirements, decisions should be **documented in an appropriate structure** and with **tool support**

<end>
Thank you.
Questions?
Discussion!
</end>



Tom-Michael Hesse, Christian Kücherer, Barbara Paech

Institute of Computer Science

Chair of Software Engineering

Im Neuenheimer Feld 326

69120 Heidelberg, Germany

<http://se.ifi.uni-heidelberg.de>

hesse|kuecherer|paech@informatik.uni-heidelberg.de



RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG
