Naming the Pain in Requirements Engineering (NaPiRE) – 2014

Results from Germany

Establishment of an open and generalisable survey basis for RE
Expectations and status quo in RE
Problems and needs in RE

www.re-survey.org

Dr. Daniel Méndez, TU München
Prof. Dr. Stefan Wagner, Uni Stuttgart

In cooperation with
Ahmet Tuzcu, TU München
Prof. Dr. André Schekelmann, Hochschule Niederrhein
Agenda

1. Background and Goals of NaPiRE
2. NaPiRE 2014 Germany – Study Population
3. Status Quo …
   - … in Requirements Engineering
   - … in Requirements Engineering Process Standards
   - … in Requirements Engineering Improvement
4. Contemporary Problems in Requirements Engineering
Evidence-based RE Research: Current State and Implications

Current State

- Available surveys beyond RE are not reliable / reproducible / detailed enough (e.g., Chaos report)
- Most investigations in RE remain isolated
  - Generalizations difficult as no empirical survey basis available

Implication

- Practically relevant problems and goals not sufficiently understood – this makes e.g. the evaluation of the new RE methods tricky as we do not know exactly what to check
- Research in RE is not (industry) problem-driven enough, consequence is weak transfer of (practical and basic) RE research results to practice
NaPiRE: Naming the Pain in Requirements Engineering

Objectives, Benefits, Approach, Principles

Objectives
• Establishing an open and generalizable survey basis for RE
• Survey on status quo and problems and needs in Requirements Engineering

Benefits
• Long-term investigations of current state of RE practices
• Support to problem-driven RE research

Approach
• Globally distributed, bi-yearly replicated family of surveys
• Under the umbrella of the International Software Engineering Research Network

Principles
• Openness and transparency
• Anonymity, but closed
• Based on theory and expectations

www.re-survey.org
The (current) NaPiRE-World

Countries that conducted surveys in 2014 (or are on their way)

Austria, Brazil, Canada, China, Estonia, Finland, Germany, Ireland, Luxembourg, Netherlands, Norway, Spain, Sweden, USA
The (current) NaPiRE World

The NaPiRE Team

NaPiRE Austria
Michael Felderer, University of Innsbruck

NaPiRE Brazil
Marcos Kalinowski, Universidade Federal Fluminense (UFF)
Rodrigo Spinola, UNIFACS/Fraunhofer Project Center at UFBA
Rafael Prikladnicki, Pontifícia Universidade Católica do Rio Grande do Sul (PUC-RS)
Tayana Conte, Universidade Federal do Amazonas (UFAM)

NaPiRE Canada
Günther Ruhe, University of Calgary

NaPiRE China
Qing Wang, Institute of Software Chinese Academy of Sciences

NaPiRE Estonia
Dietmar Pfahl, University of Tartu

NaPiRE Finland
Markku Oivo, University of Oulu
Tomi Männistö, University of Helsinki
Casper Lassenius, Aalto University

NaPiRE Germany
Daniel Méndez Fernández, Technische Universität München
Stefan Wagner, University of Stuttgart

NaPiRE Ireland
Desmond Greer, Queen’s University Belfast

NaPiRE Luxembourg
Mehrdad Sabetzadeh (temp.), University of Luxembourg

NaPiRE Netherlands
Maya Daneva, University of Twente
Roel Wieringa, University of Twente

NaPiRE Norway
Sagar Sen (temp.), Simula Research Laboratory

NaPiRE Spain
Oscar Pastor, Universidad Politécnica de Valencia
Nelly Condori-Fernandez, PROS Research Center

NaPiRE Sweden
Marie-Therese Christiansson, Karlstads Universitet
André Schekelmann, Hochschule Niederrhein

NaPiRE USA
Birgit Penzenstadler, University of California, Irvine
NaPiRE focusses on five main research questions

This report presents the results from Germany

Research questions and methods

1. What is the status quo in RE?!
2. What is the status quo in RE process standards?!
3. What is the status quo in RE improvement?!
4. What contemporary problems exist in RE and how do they manifest themselves?!
5. Are there observable patterns of expectations, status quo, and problems in RE?!

„Theory of expectation“
„Grounded Theory“

Version: 1.4 (Changes in questionnaire from 2013 from feedback rounds)
Agenda

1. Background and Goals of NaPiRE

2. NaPiRE 2014 Germany – Study Population

3. Status Quo …
   - … in Requirements Engineering
   - … in Requirements Engineering Process Standards
   - … in Requirements Engineering Improvement

4. Contemporary problems in Requirements Engineering
Study Population

The study is mainly based on experienced people working in big companies

- German companies only
- 125 companies were invited, 54 participated, 42 almost completed questionnaires
- Only fully completed questionnaires (30) are considered in the results here

General characteristics of companies

- Most respondents work in big companies (>2000 employees) (50%), rest equally distributed over companies of different size, 50% of the companies work in globally distributed settings
- Industries: Consulting & Software Development (40%), Banking & Insurance (17%), Logistics & Transportation (13%)
- 50% of the companies work in product development, 30% as customer, 26% as contractor (more than one role possible)
- Non-agile process models are (still) widely used (Waterfall: 23%, V-Model XT: 12%, RUP: 20% / Scrum: 31%, XP 2% / other: 20%) (more than one model possible)

General characteristics of participants

- Respondents work in different roles: project lead (26%), requirements engineer (20%), or other
- 90% have more than 3 years experience in their role
1. Background and Goals of NaPiRE
2. NaPiRE 2014 Germany – Study Population
3. Status Quo …
   ... in Requirements Engineering
   ... in Requirements Engineering Process Standards
   ... in Requirements Engineering Improvement
4. Contemporary problems in Requirements Engineering
Status Quo in RE – How do you elicit requirements?

Requirements elicitation is „People Business“

- Meetings / Workshops: 86.67%
- Interviews: 86.67%
- Scenarios: 50.00%
- Prototyping: 43.33%
- Observation: 26.67%
- Other: 16.67%
### Status Quo in RE – *How do you document requirements?*

*Requirements are written, Models are (often) drawn*

#### Functional Requirements

<table>
<thead>
<tr>
<th></th>
<th>Business Process Models</th>
<th>Use Case Models</th>
<th>Goal Models</th>
<th>Data Models</th>
<th>Structured Requirements Lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free textual form</td>
<td></td>
<td>33%</td>
<td></td>
<td>3%</td>
<td>17%</td>
</tr>
<tr>
<td>Textual with constraints</td>
<td></td>
<td>10%</td>
<td>33%</td>
<td>13%</td>
<td>7%</td>
</tr>
<tr>
<td>Semi-formal (UML)</td>
<td></td>
<td>43%</td>
<td>57%</td>
<td>7%</td>
<td>43%</td>
</tr>
<tr>
<td>Formal</td>
<td></td>
<td>17%</td>
<td>3%</td>
<td>3%</td>
<td>27%</td>
</tr>
</tbody>
</table>

#### Non-Functional Requirements

- 52% We use quantified textual requirements
- 33% We use non-quantified textual requirements
- 15% Other
Status Quo in RE – *Which traces do you explicitly manage?*

*Traces are managed according to specific (individual) needs*

- Traces between requirements and design documents: 40%
- Traces between requirements and code: 23%
- None: 3%
- Other: 33%

- Traces between requirements, underlying needs and context of use
- Traces between requirements and feature specifications
- Traces between requirements (different req. levels);
- Traces between change requests and requirements
- Request - Requirements - tech-Specs - Code
- Theoretically a lot
Status Quo in RE – *How do you analyze the effect of changes to requirements?*

*Concerning the use of traces, there is room for improvement*

- Impact Analysis on the code: 20%
- Impact Analysis between requirements: 57%
- No analysis of the effect of changes to requirements: 30%
- Other: 13%

- Between requirements by review (often with developers)
- Impact from requirements to test cases
- Rerun test suites
- We use monitoring
### Status Quo in RE – How do you align the software test with the requirements?

**Requirements are tested – full coverage is not reached**

<table>
<thead>
<tr>
<th>Action</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>We define acceptance criteria for requirements</td>
<td>67%</td>
</tr>
<tr>
<td>We check the coverage of requirements with tests.</td>
<td>53%</td>
</tr>
<tr>
<td>Testers participate in requirements reviews.</td>
<td>40%</td>
</tr>
<tr>
<td>We derive tests from system models</td>
<td>27%</td>
</tr>
<tr>
<td>Other</td>
<td>7%</td>
</tr>
<tr>
<td>We do not align test and requirements</td>
<td>7%</td>
</tr>
</tbody>
</table>
**Status Quo in RE – How do you deal with changing requirements after the initial release?**

**Requirements are often not explicitly changed after an initial release**

“We write a system documentation based on requ. spec.; this documentation is updated after each release based on CRs or new requirement specs.”

- 17%: We regularly change the requirements specification.
- 30%: We update our product backlog.
- 40%: We only work with change requests.
- Other 13%
Status Quo in RE Standards – *What RE company standard have you established in your company?*

*Standards exist – mostly, however, people use company-specific adaptations*

- An own standard that defines artefacts and offers document templates (53%)
- An own standard that defines the process including roles and responsibilities (43%)
- An own standard that defines the coarse process with deliverables (40%)
- A standard that is predefined by the development process (e.g., RUP, Scrum) (37%)
- A standard that is predefined according to a regulation (e.g., ITIL) (10%)
- Other (7%)
- None (7%)
Status Quo in RE Standards – *Is the RE standard mandatory and practiced?*

In 2/3 of the companies, the company standard is practiced
### Status Quo in RE Standards – *Which reasons do you agree with as an argument to define a company standard?*

*The main reason is to make explicit the interconnected concepts of the domain*

<table>
<thead>
<tr>
<th>Reason</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better quality assurance of the artefacts (e.g., within quality gates)</td>
<td>47%</td>
<td>7%</td>
<td>0%</td>
</tr>
<tr>
<td>Seamless development by integrating RE into the development process</td>
<td>43%</td>
<td>13%</td>
<td>3%</td>
</tr>
<tr>
<td>Higher efficiency</td>
<td>40%</td>
<td>10%</td>
<td>0%</td>
</tr>
<tr>
<td>Knowledge transfer</td>
<td>40%</td>
<td>20%</td>
<td>7%</td>
</tr>
<tr>
<td>Support of project management and planning</td>
<td>37%</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td>Better support of progress control</td>
<td>33%</td>
<td>20%</td>
<td>3%</td>
</tr>
<tr>
<td>Support of distributed development</td>
<td>27%</td>
<td>30%</td>
<td>7%</td>
</tr>
<tr>
<td>Better tool support</td>
<td>23%</td>
<td>37%</td>
<td>7%</td>
</tr>
<tr>
<td>Compliance to regulations and standards (like CMMI)</td>
<td>17%</td>
<td>23%</td>
<td>23%</td>
</tr>
<tr>
<td>Support of benchmarks and / or comparison of different projects</td>
<td>7%</td>
<td>43%</td>
<td>17%</td>
</tr>
<tr>
<td>Formal prerequisite for project acquisition in our domain</td>
<td>3%</td>
<td>40%</td>
<td>13%</td>
</tr>
</tbody>
</table>
Status Quo in RE Standards – *Which barriers do you see for defining RE standard in your company?*

*Main barrier is missing willingness to change*

<table>
<thead>
<tr>
<th></th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing willingness for changes</td>
<td>43%</td>
<td>3%</td>
<td>10%</td>
</tr>
<tr>
<td>Higher process complexity</td>
<td>30%</td>
<td>10%</td>
<td>13%</td>
</tr>
<tr>
<td>Higher demand for communication</td>
<td>13%</td>
<td>20%</td>
<td>13%</td>
</tr>
<tr>
<td>Lower efficiency</td>
<td>10%</td>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>Missing possibilities of standardisation</td>
<td>10%</td>
<td>33%</td>
<td>7%</td>
</tr>
</tbody>
</table>
Status Quo in RE Standards – How do you check the application of your requirements engineering standard?

The application of the standard is checked – in different ways

- "Tool-based."
- "Practices are implemented in RM tool, which supports as well as enforces compliance to process; additional means: coaching by RE team, extensive training and other knowledge resources"

- via project assessments (47%)
- via constructive quality assurance, e.g. via checklists or templates (40%)
- via analytic quality assurance, e.g. as part of quality gates (40%)
- it is not checked (17%)
- other (10%)
Status Quo in RE Standards – How is your requirements engineering standard applied (tailored) in your regular projects?

Tailoring is mostly done on an experience-based manner

- At the beginning of a project, the project lead / requirements engineer tailors the standard based on experiences - 47%
- We have defined a tailoring approach that continuously guides the application of the standard in our project - 27%
- We have tool support for tailoring our Requirements Engineering standard - 23%
- We do not consider a particular tailoring approach - 20%
- Other - 3%
Status Quo in RE Improvement – *Is your RE engineering continuously improved?*

*Companies always try to get better*

- **43%** Yes, we improve our requirements engineering via an own business unit / role.
- **43%** Yes, we improve our requirements engineering via external consultants.
- **7%** Yes, our project teams improve requirements engineering.
- **7%** No
Status Quo in RE Improvement – *Why do you continuously improve your requirements engineering?*

There is a strong intrinsic motivation to improve RE

It helps us to determine our strengths and weaknesses and act accordingly. 67%

Other 23%

An improvement is demanded by a regulation (e.g., CMMI, Cobit, or ITIL). 7%

An improvement is expected by our customers. 7%

We conduct the improvement to obtain a certain certification. 0%

- "Because RE is what we sell to our customers, and therefore we have to be at a high level of knowledge and quality"
- "Part of overall improvement of SE practices"
- "Only by continuous improvement and evolution, the RM practices can be kept functioning and effective in a huge global product development organization"
- "It helps us to meet our customers demands better and stabilizes our product quality"
- "To accelerate projects and avoid mistakes made in the past"
- "We want to show how important good RE is"
Status Quo in RE Improvement – *Do you use a normative, external standard for your improvement?*

*Normative improvement approaches a la CMMI are essentially neglected*

- Yes, we use an external standard for assessing RE (e.g., CMMI for RE)
- No, we use an internally defined (company-specific) standard for improving RE
Agenda

1. Background and Goals of NaPiRE
2. NaPiRE 2014 Germany – Study Population
3. Status Quo …
   ... in Requirements Engineering
   ... in Requirements Engineering Process Standards
   ... in Requirements Engineering Improvement
4. Contemporary problems in Requirements Engineering
How do you rate these statements concerning you RE standard?

**RE standard seems to miss clear guidance concerning artifacts and procedures**

<table>
<thead>
<tr>
<th>RE standard ...</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>... is hard to understand</td>
<td>3%</td>
<td>23%</td>
<td>33%</td>
</tr>
<tr>
<td>... is too complex</td>
<td>3%</td>
<td>17%</td>
<td>33%</td>
</tr>
<tr>
<td>... is too abstract</td>
<td>7%</td>
<td>17%</td>
<td>40%</td>
</tr>
<tr>
<td>... does not support the specification of precise requirements</td>
<td>3%</td>
<td>10%</td>
<td>43%</td>
</tr>
<tr>
<td>... does not scale to our projects' high complexity</td>
<td>3%</td>
<td>13%</td>
<td>40%</td>
</tr>
<tr>
<td>... is too heavy weight for our projects (e.g. does not support agility)</td>
<td>3%</td>
<td>13%</td>
<td>33%</td>
</tr>
<tr>
<td>... is not flexible enough (e.g. it offers no means to tackle moving targets / chance intensive requirements)</td>
<td>7%</td>
<td>7%</td>
<td>30%</td>
</tr>
<tr>
<td>... does not sufficiently define a clear terminology</td>
<td>10%</td>
<td>13%</td>
<td>33%</td>
</tr>
<tr>
<td>... gives no guidance on how to create the specification documents</td>
<td>13%</td>
<td>3%</td>
<td>40%</td>
</tr>
<tr>
<td>... does not sufficiently allow for deviation according to our project circumstances that cannot be formalized</td>
<td>3%</td>
<td>17%</td>
<td>33%</td>
</tr>
<tr>
<td>... does not fit to the variety of our projects (e.g. size or technical domain)</td>
<td>3%</td>
<td>10%</td>
<td>27%</td>
</tr>
<tr>
<td>... does not sufficiently define roles and responsibilities</td>
<td>13%</td>
<td>20%</td>
<td>30%</td>
</tr>
</tbody>
</table>
What are the problems experienced by the respondents in the projects?
Question 28: Considering your personal experiences, how do the following (more general) problems in requirements engineering apply to your projects?

<table>
<thead>
<tr>
<th>Problem</th>
<th>I disagree</th>
<th>Neutral</th>
<th>I agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication flaws within the project</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>development team</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication flaws between developers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and the customer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminological problems</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question 29: Considering your personally experienced problems (stated in the previous question), which ones would you classify as the five most critical ones (ordered by their relevance)

- Problem #1 (most critical one): Please make a selection
- Problem #2: Please make a selection
- Problem #3: Communication flaws between us and the customer
- Problem #4: Terminological problems
- Problem #5: Unclear responsibilities
- Incomplete and/or hidden requirements

Question 33: Considering your personally experienced most critical problems (selected in the previous question), which would you classify as a major cause for project failure (if at all)?

- Terminological problems
- Unclear responsibilities
- Incomplete and/or hidden requirements
- Insufficient support by project lead
- Insufficient support by customer
Question 29: Considering your personally experienced problems (stated in the previous question), which ones would you classify as the five most critical ones (ordered by their relevance)

### Analysis of most critical RE problems for causes and consequences

<table>
<thead>
<tr>
<th>Problem #1 (most critical one)</th>
<th>Ursache zu Problem #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem #2</td>
<td>Ursache zu Problem #2</td>
</tr>
<tr>
<td>Problem #3</td>
<td>Ursache zu Problem #3</td>
</tr>
<tr>
<td>Problem #4</td>
<td>Ursache zu Problem #4</td>
</tr>
<tr>
<td>Problem #5</td>
<td>Ursache zu Problem #5</td>
</tr>
</tbody>
</table>

### Ursachen der RE Probleme

<table>
<thead>
<tr>
<th>Problem</th>
<th>Ursache zu Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem #1</td>
<td>Ursache zu Problem #1</td>
</tr>
<tr>
<td>Problem #2</td>
<td>Ursache zu Problem #2</td>
</tr>
<tr>
<td>Problem #3</td>
<td>Ursache zu Problem #3</td>
</tr>
<tr>
<td>Problem #4</td>
<td>Ursache zu Problem #4</td>
</tr>
<tr>
<td>Problem #5</td>
<td>Ursache zu Problem #5</td>
</tr>
</tbody>
</table>

### Auswirkungen der RE Probleme

<table>
<thead>
<tr>
<th>Problem</th>
<th>Auswirkung zu Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem #1</td>
<td>Auswirkung zu Problem #1</td>
</tr>
<tr>
<td>Problem #2</td>
<td>Auswirkung zu Problem #2</td>
</tr>
<tr>
<td>Problem #3</td>
<td>Auswirkung zu Problem #3</td>
</tr>
<tr>
<td>Problem #4</td>
<td>Auswirkung zu Problem #4</td>
</tr>
<tr>
<td>Problem #5</td>
<td>Auswirkung zu Problem #5</td>
</tr>
</tbody>
</table>

Approach: Analysis of causes and consequences
How do the following (more general) problems in requirements engineering apply to your projects?

Requirements Engineering remains inherently complex and volatile

<table>
<thead>
<tr>
<th>Problem</th>
<th>Agree</th>
<th>Neutral</th>
<th>Diagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit requirements not made explicit</td>
<td>37%</td>
<td>10%</td>
<td>3%</td>
</tr>
<tr>
<td>Stakeholders with difficulties in separating requirements from previously known solution designs</td>
<td>37%</td>
<td>23%</td>
<td>3%</td>
</tr>
<tr>
<td>Moving targets (changing goals, business processes and/or requirements)</td>
<td>30%</td>
<td>20%</td>
<td>13%</td>
</tr>
<tr>
<td>Time boxing / Not enough time in general</td>
<td>27%</td>
<td>17%</td>
<td>17%</td>
</tr>
<tr>
<td>Incomplete</td>
<td>23%</td>
<td>17%</td>
<td>7%</td>
</tr>
<tr>
<td>Communication flaws between developers and the customer</td>
<td>20%</td>
<td>23%</td>
<td>3%</td>
</tr>
<tr>
<td>Insufficient support by customer</td>
<td>13%</td>
<td>33%</td>
<td>7%</td>
</tr>
<tr>
<td>Weak access to customer needs and/or (internal) business information</td>
<td>13%</td>
<td>23%</td>
<td>13%</td>
</tr>
<tr>
<td>Inconsistent requirements</td>
<td>10%</td>
<td>17%</td>
<td>13%</td>
</tr>
<tr>
<td>Discrepancy between high degree of innovation and need for formal acceptance of (potentially wrong/incomplete/unknown) requirements</td>
<td>10%</td>
<td>23%</td>
<td>17%</td>
</tr>
<tr>
<td>Unclear/unmeasurable non-functional requirements</td>
<td>10%</td>
<td>27%</td>
<td>13%</td>
</tr>
<tr>
<td>Volatile customer’s business domain regarding, e.g., changing points of contact, business processes or requirements</td>
<td>10%</td>
<td>30%</td>
<td>20%</td>
</tr>
<tr>
<td>Terminological problems</td>
<td>7%</td>
<td>37%</td>
<td>10%</td>
</tr>
<tr>
<td>Unclear responsibilities</td>
<td>7%</td>
<td>27%</td>
<td>10%</td>
</tr>
<tr>
<td>Insufficient support by project lead</td>
<td>7%</td>
<td>37%</td>
<td>20%</td>
</tr>
<tr>
<td>Missing traceability</td>
<td>7%</td>
<td>30%</td>
<td>17%</td>
</tr>
<tr>
<td>&quot;Gold plating&quot; (implementation of features without corresponding requirements)</td>
<td>7%</td>
<td>30%</td>
<td>17%</td>
</tr>
<tr>
<td>Underspecified requirements that are too abstract and allow for various interpretations</td>
<td>7%</td>
<td>30%</td>
<td>20%</td>
</tr>
<tr>
<td>Communication flaws within the project development team</td>
<td>3%</td>
<td>23%</td>
<td>7%</td>
</tr>
<tr>
<td>Weak knowledge of customer's application domain</td>
<td>3%</td>
<td>17%</td>
<td>33%</td>
</tr>
<tr>
<td>Weak relationship between customer and project lead</td>
<td>0%</td>
<td>33%</td>
<td>27%</td>
</tr>
<tr>
<td>Technically unfeasible requirements</td>
<td>0%</td>
<td>30%</td>
<td>23%</td>
</tr>
</tbody>
</table>
**Which one are the most critical ones? Which can even lead to project failure?**

*The problems lie in the heart of RE: Decide what you want!*

<table>
<thead>
<tr>
<th>Issue</th>
<th>Prio 1</th>
<th>Prio 2</th>
<th>Prio 3</th>
<th>Prio 4</th>
<th>Prio 5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete and / or hidden requirements</td>
<td>10%</td>
<td>20%</td>
<td>7%</td>
<td>13%</td>
<td>3%</td>
<td>53%</td>
</tr>
<tr>
<td>Moving targets (changing goals, business processes and / or requirements)</td>
<td>20%</td>
<td>13%</td>
<td>0%</td>
<td>7%</td>
<td>7%</td>
<td>47%</td>
</tr>
<tr>
<td>Stakeholders with difficulties in separating requirements from previously known solution designs</td>
<td>10%</td>
<td>13%</td>
<td>13%</td>
<td>0%</td>
<td>0%</td>
<td>37%</td>
</tr>
<tr>
<td>Time boxing / Not enough time in general</td>
<td>17%</td>
<td>10%</td>
<td>0%</td>
<td>7%</td>
<td>0%</td>
<td>33%</td>
</tr>
<tr>
<td>Discrepancy between high degree of innovation and need for formal acceptance of (potentially wrong / incomplete / unknown) requirements</td>
<td>0%</td>
<td>0%</td>
<td>7%</td>
<td>10%</td>
<td>13%</td>
<td>30%</td>
</tr>
<tr>
<td>Communication flaws between us and the customer</td>
<td>7%</td>
<td>3%</td>
<td>10%</td>
<td>3%</td>
<td>7%</td>
<td>30%</td>
</tr>
<tr>
<td>Communication flaws within the project team</td>
<td>17%</td>
<td>7%</td>
<td>3%</td>
<td>0%</td>
<td>0%</td>
<td>27%</td>
</tr>
<tr>
<td>Underspecified requirements that are too abstract and allow for various interpretations</td>
<td>0%</td>
<td>0%</td>
<td>10%</td>
<td>7%</td>
<td>7%</td>
<td>23%</td>
</tr>
<tr>
<td>Insufficient support by customer</td>
<td>3%</td>
<td>7%</td>
<td>3%</td>
<td>0%</td>
<td>7%</td>
<td>20%</td>
</tr>
</tbody>
</table>

Top 5 problems that can lead to project failure
Which implications do the most critical problems have?
(based on grounded theory)

Problems in RE have severe consequences for both product and project.

- Late delivery increased time pressure: 54%
- Poor product quality: 46%
- Increased costs: 39%
- Increased use of resources/effort: 36%
- Increased time for clarification/general misunderstanding: 29%
- Increased number of change requests: 29%
- Requirements get unclear/poor quality of requirements: 18%
- General demotivation/frustration: 18%
- Decreased user acceptance: 14%
- Increased number of failures/bugs: 11%
- Decreased efficiency/ performance: 11%
- Need for rework/post implementation: 11%
- Customer dissatisfaction: 11%
- Miscommunication: 7%
- Incorrect/poor implementation: 7%
What are the causes for the most critical problems? (based on grounded theory)

Problems arise from missing skills and lack of support, but often it’s due to an uncontrollable uncertainty in a project (e.g. the customer engagement).
What are the causes as they have been experienced by the respondents?
Causes – as stated by the respondents

Causes mentioned by those respondents that also mentioned the Top 5 problems

Missing Awareness of RE
• Availability and understanding of importance of RE.
• Insufficient understanding of RE by stakeholders.
• Lack of willingness to change; underestimating the effort for requirements engineering;

Deficiencies in (Use of) Requirements Engineering Methods
• Missing company wide standard
• No concrete examples for requirements (no specification by example)
• Sometimes the wrong persons are asked for the requirements
• Unstructured elicitation
• Lack of measurements implemented in business environment do not allow to specify precisely measurable NFRs
• Bad RE knowledge / Method is only mandatory in IT
• Missing competence in RE
• Qualification in RE of team members
• Most Requirements are standard. If there are innovative solutions affordable people are not trained to adequate elicitate requirements
Causes – as stated by the respondents
Causes mentioned by those respondents that also mentioned the Top 5 problems

Too much Solution Orientation

- Real business needs are not transparent to delivery teams. they just get the solutions to implement
- Thinking in solutions leads to disregarding the understanding of the problem and the requirements from all stakeholders
- It's easier to talk in solutions
- Stakeholders always bring the solutions as requirements, and don’t express the real requirements behind them
- Requirement wrongly specified in terms of a solution (that is not feasible)
- Obvious: many people tend to think in known solutions
- Product owner is presentig the solution and not the problem he is trying to solve
- The focus is on what is new/challenging/exciting and never on completeness. Combined with time boxing this leads to missing requirements
- Long term employees have good knowledge of the systems and therefore don't concentrate on the requirements but on the solution
- Missing concentration on business processes, teams are thinking in issues
- Customer doesn't really know what he wants
- Customer do not know what he is really needing
Causes – as stated by the respondents

Causes mentioned by those respondents that also mentioned the Top 5 problems

Deficiencies in Customer Collaboration

- Communication between customer and consultant as well as consultant and developer
- Complicated customer
- Lack of interest in domain of business of the customer
- Little information about each other
- There are boundaries between delivery teams and the end customers. No direct communication
- Problems with matrix organization in business
- The customer is king and hardly someone is allowed to tell them about freezes and the need of formal change requests
- Customer do not have enough time
- Unclear responsibilities on customer side
- Weak management on customer side
- Not enough customer involvement in defining requirements
- Unclear business vision and understanding by stakeholders.
Causes – as stated by the respondents

Causes mentioned by those respondents that also mentioned the Top 5 problems

Project are challenging from a business and technical perspective

• Big system/project size
• Amount of requirements (> 100.000)
• technical complexity of the domain
• underestimation of the value of domain knowledge
• Missing Knowledge in the domain (customer and developer)
• Lack of standard terminology in the industry; local variances; no company standard
• Missing of a global view onto the system
• Missing understanding for the daily business on the one hand and the lack of knowledge how software is going to be developed
Causes – as stated by the respondents

Causes mentioned by those respondents that also mentioned the Top 5 problems

Deficiencies in Project Organization

• Process model is not precise enough
• On-accuracy of use of defined process
• Hierarchical structure, no self-empowered team (no responsibility taken) leads to an unbalanced communication
• Bad project management / bad project management skill off the staff
• Oversized Portfolio planning
• All channeled through PM, not much knowledge transfer
• Project members are not co-located. Different project roles have different homes in the line-organization (matrix organization).
• Under estimation of communication efforts
• People working in IT/development never got training on communication - they are not even aware of their missing skills
• Weak role of req engineers, too much focus on dev. team
• 2 different languages
• No introduction and explanation to developer
Causes – as stated by the respondents

Causes mentioned by those respondents that also mentioned the Top 5 problems

Not enough time, not enough budget

• Lack of time to discuss
• Limited time
• Time to market vs. technical complexity
• Pressure to meet budget and plan without considering the scope and effort;
• Political dates published in the newspapers
• Finalise for certain release date
• Focus on money, on time to market, on business - constraints given by management/controlling department
So, where does it hurt now?
Summary – Where does it hurt?

- … In **awareness** for RE in **management and business**
  (People working in RE are aware of problems in RE and their consequences!)

- … In the **development** of (concrete) **company specific standards**
  (Standards for RE play an important role – they are used, their motivations are seen
  – and they are company-specific, prescriptions to not help)

- … In **methods** and procedures that help to make clear **distinctions** between
  **requirements** and **solutions**

- … In **education** and training – with a special focus on the **business** side,
  not only on the **IT** side

- … In **communication** and **collaboration**

- Apart of that, it hurts at topics that lie **outside** of the area of influence of RE:
  - project organization
  - time pressure
What now?
There’s still a lot to do … NaPiRE 2014 and beyond

• **Further** analysis of the data from global replications is ongoing at the moment
• **Communication** and **Dissemination** of the results to academia and industry

→ Stay tuned and visit our website http://re-survey.org

Work is not done, though: Please (continue to) support us … e.g. in NaPiRE 2016 😊