Experiences with Empirical PhD Work

Prof. Dr. Barbara Paech
Institut für Informatik
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
paech@informatik.uni-heidelberg.de

software engineering heidelberg

Software Engineering@Uni HD

- Prof. Dr. Barbara Paech
  - Since 18 years in HD
  - before Fh IESE, Kaiserlautern
  - 15 finished PhD students
  - 5 ongoing PhD students

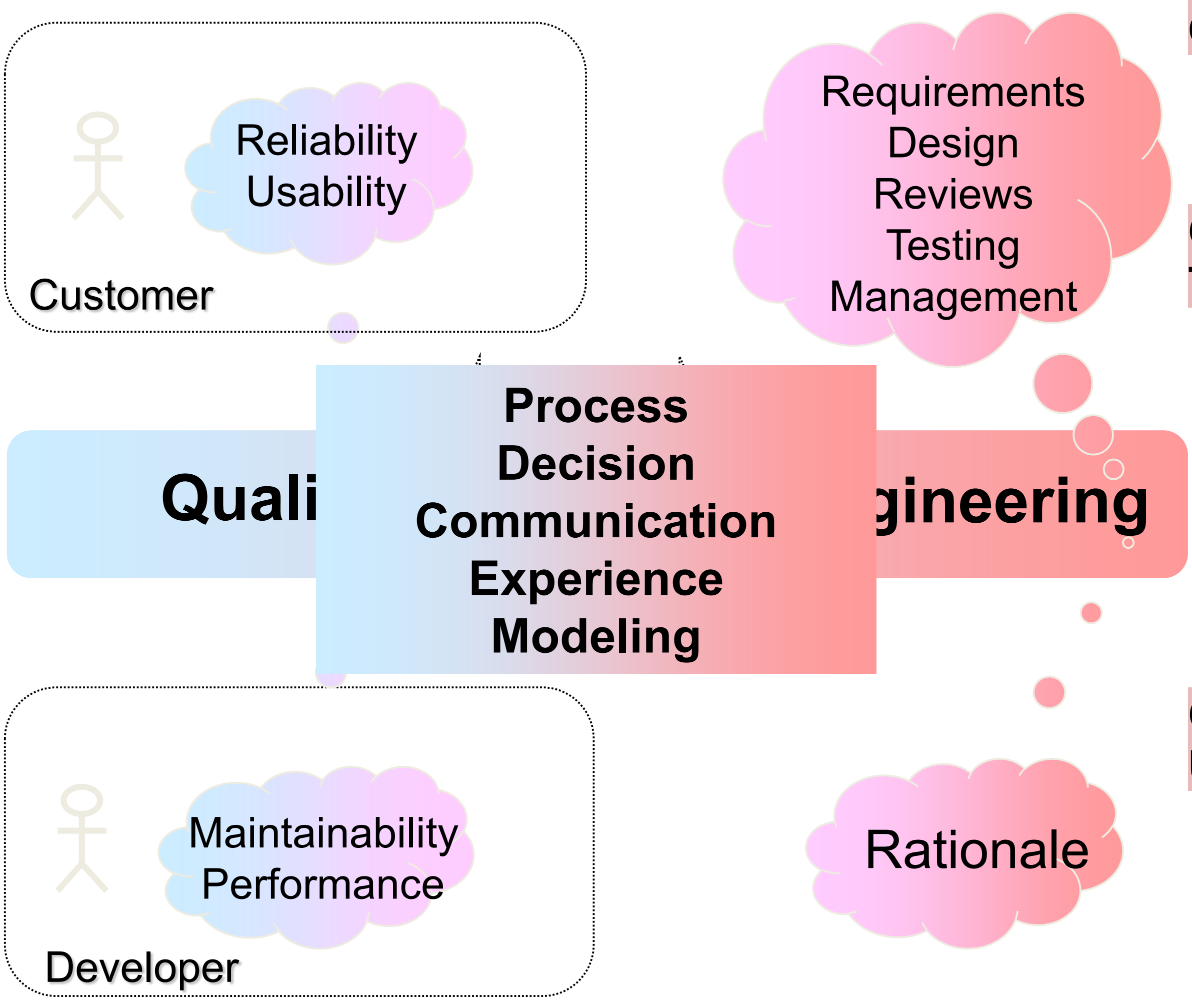
- **Profile Quality Engineering through Software Engineering Intelligence**

- **Products**
  - SE teaching and consulting
  - Requirements Engineering Method TORE
  - Rationale Management Tool (with TU München)

Our Topics

Misuse-Oriented Quality RE

Aligning BP and IS Quality



Task-Oriented RE

Feature Management

Continuous Traceability

Empirical Test Foci Definition

Scientific Software Testing

CASE-Tool UNICASE

Mgt–Tool ConDec

software engineering heidelberg

Important Principles

- **Humans are important**
  - TORE: base requirements on the tasks of the users
  - Finished PhD: Predicting user satisfaction
  - Finished PhD: Improve communication of decisions between users and developers
- **Decision making is important**
  - Capture rationale to improve quality, communication, maintenance
  - Current PhD: Continuous decision making

  - Finishd PhD: Continuous trace capture between requirements and code
  - Finished PhD: RE for decision support systems
- **Empirical Research is important**
  - Take problem from industry, evaluate solution in industry
  - Finished PhD: Empirical test-foci definition: base future test focus on empirical evaluation of system and process data
  - Finished PhD: Mining feature descriptions

Agenda

- Motivation: Ideal SE research
- Existing approaches
- Our PhD approach
- Example PhD
- Open Questions

Hevner et al: Design Science Research

[Hevner et al 2004]

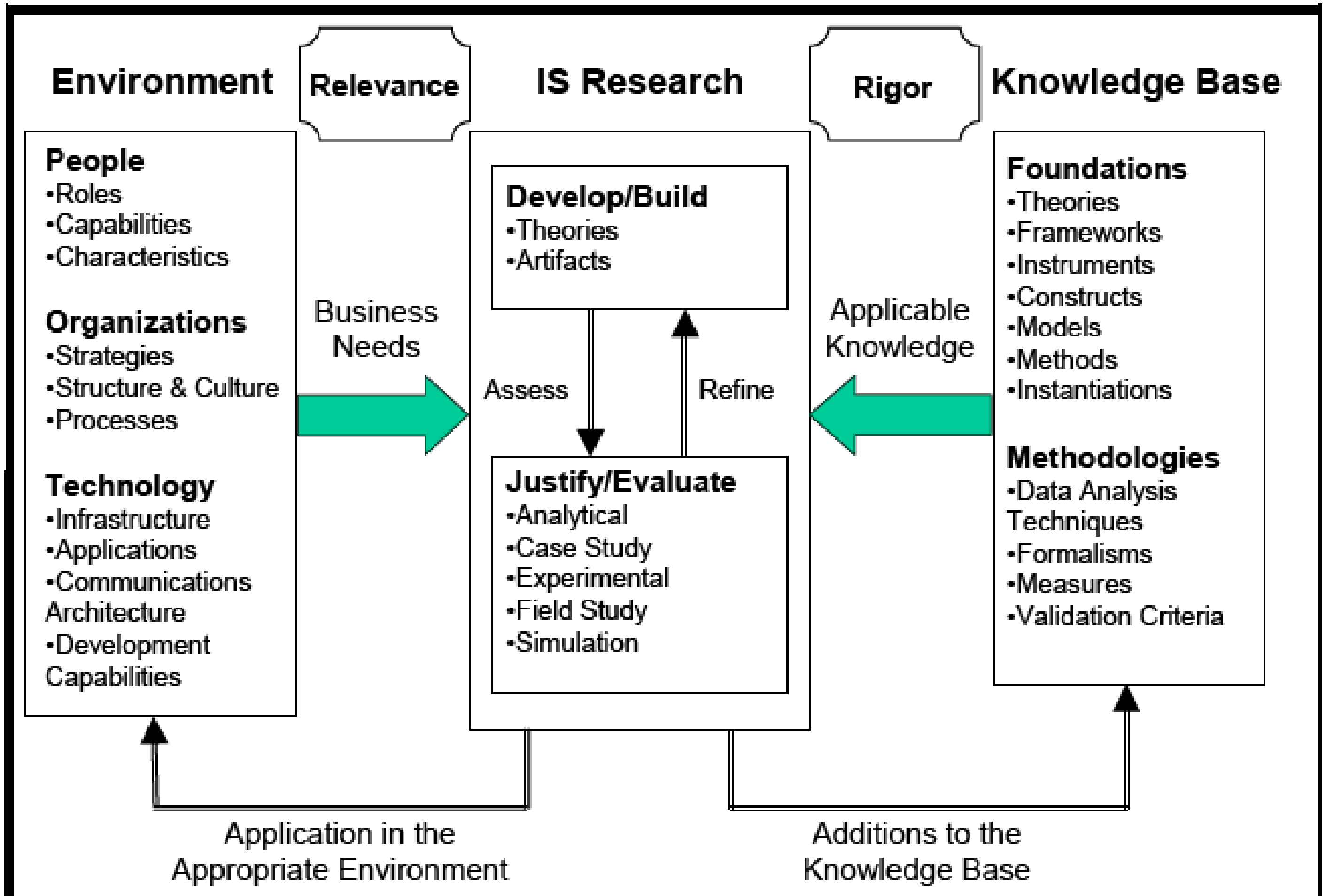


Figure 2. Information Systems Research Framework

Ideal Software Engineering Research

Exp1: Establishing a problem can be a PhD on its own.

- Observe SE Practice (to identify relevant problems)
  - Create a justified theory for practice problems
  - Create a justified theory for the solution idea

Establish Problem

- Design solution (Method/Tool)

Design Solution

Exp2: Designing the solution is often the simpler part. Validation must be considered right from the beginning.

- Validate solution
  - First in academia, then in practice
  - Create a justified theory for the solution (to learn for the next problem)

Validate Solution

Similarity to ideal SE practice

Research

- Observe SE Practice (to identify relevant problems)
  - Theory for practice problems
  - Theory for the solution idea
- Design solution (Method/Tool)
- Validate solution
  - First in academia, then in practice
  - Create a justified theory for the solution (to learn for the next problem)

Practice

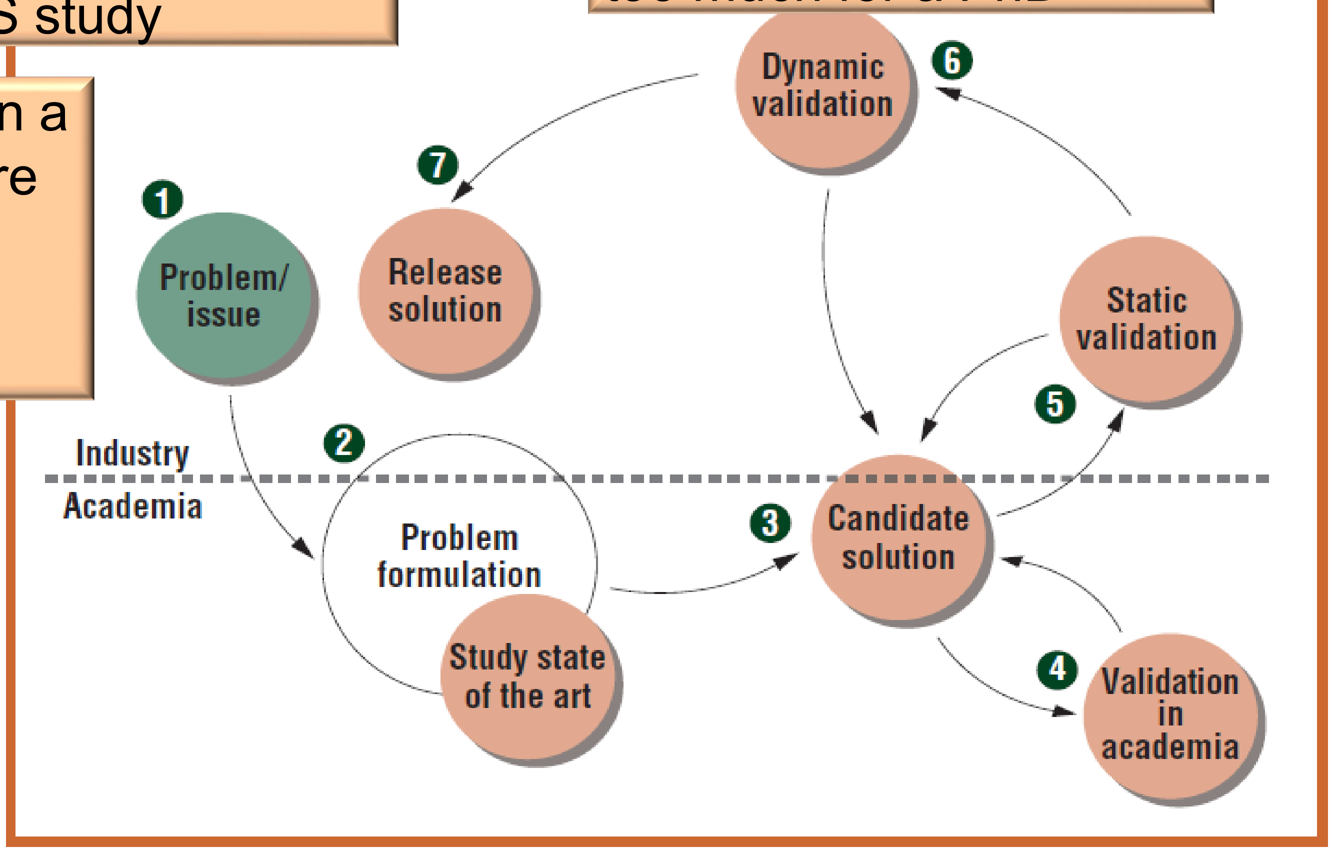
- Observe business practice (software usage)
  - Theory for problems (business case)
  - Theory for solution (software specification)
- Build software
- Prototype, Test
- Operation in production environment
- Observe benefits and effects to learn for next release

Gorschek et al: Technology Transfer

Exp3: Clients do not like to spend much time on AS-IS study

Exp5: Solution release is too much for a PhD

Exp4: There is often a problem idea, before there is a client. Finding the right client is difficult.

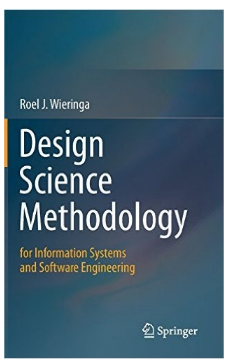


[Gorschek et al 2006]

Main goal: to help client
Distinguish validation in academia and industry in several stages

Wieringa: Technical Action Research

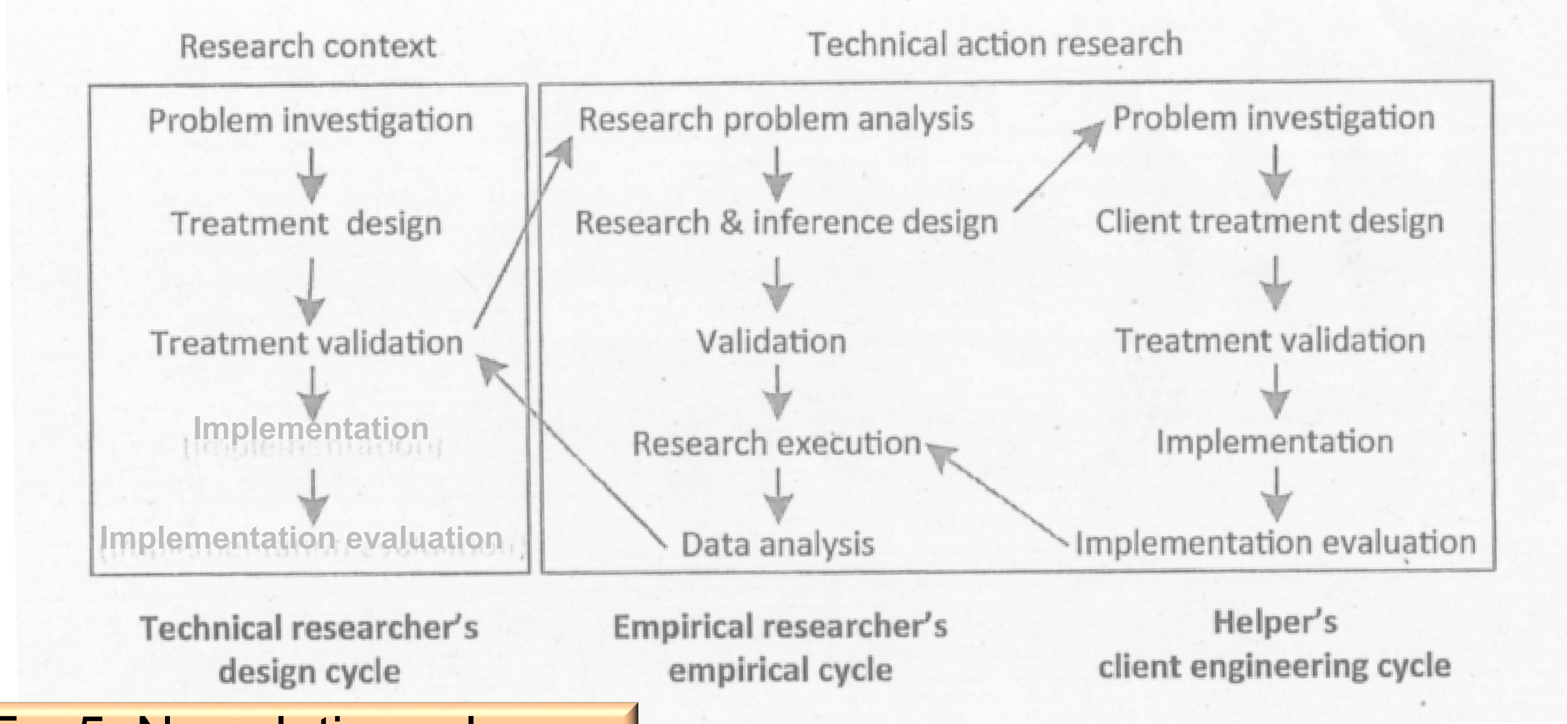
[Wieringa,Morali 2012]



[Wieringa 2014]

Exp4: Problem idea

Exp6: It is difficult to balance the clients interest and the empirical research goal

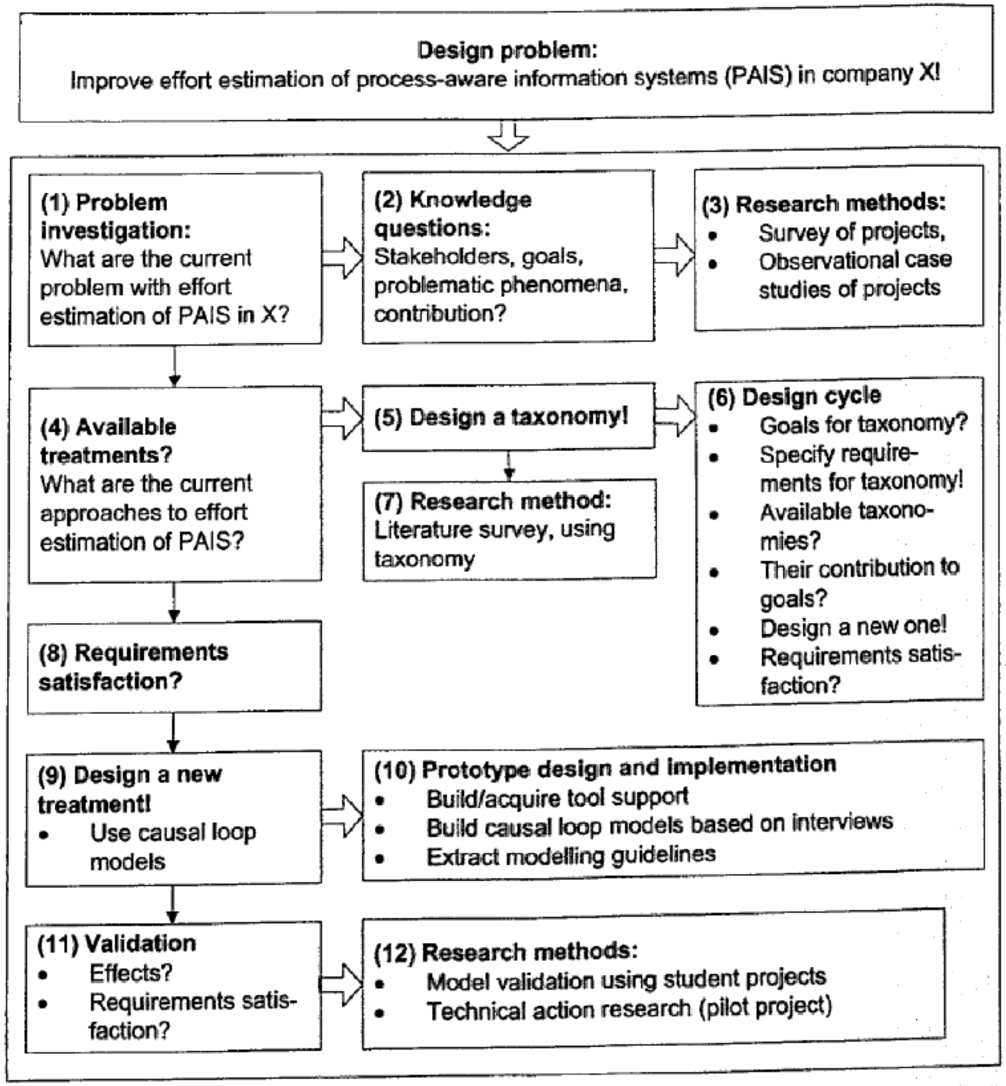


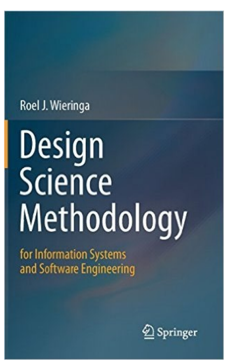
Exp5: No solution release

Several goals:
Distinguish overall research, validation research and improvement for client

Wieringa: PhD example

Exp7: Several small studies easier than one big study, possibly with several clients

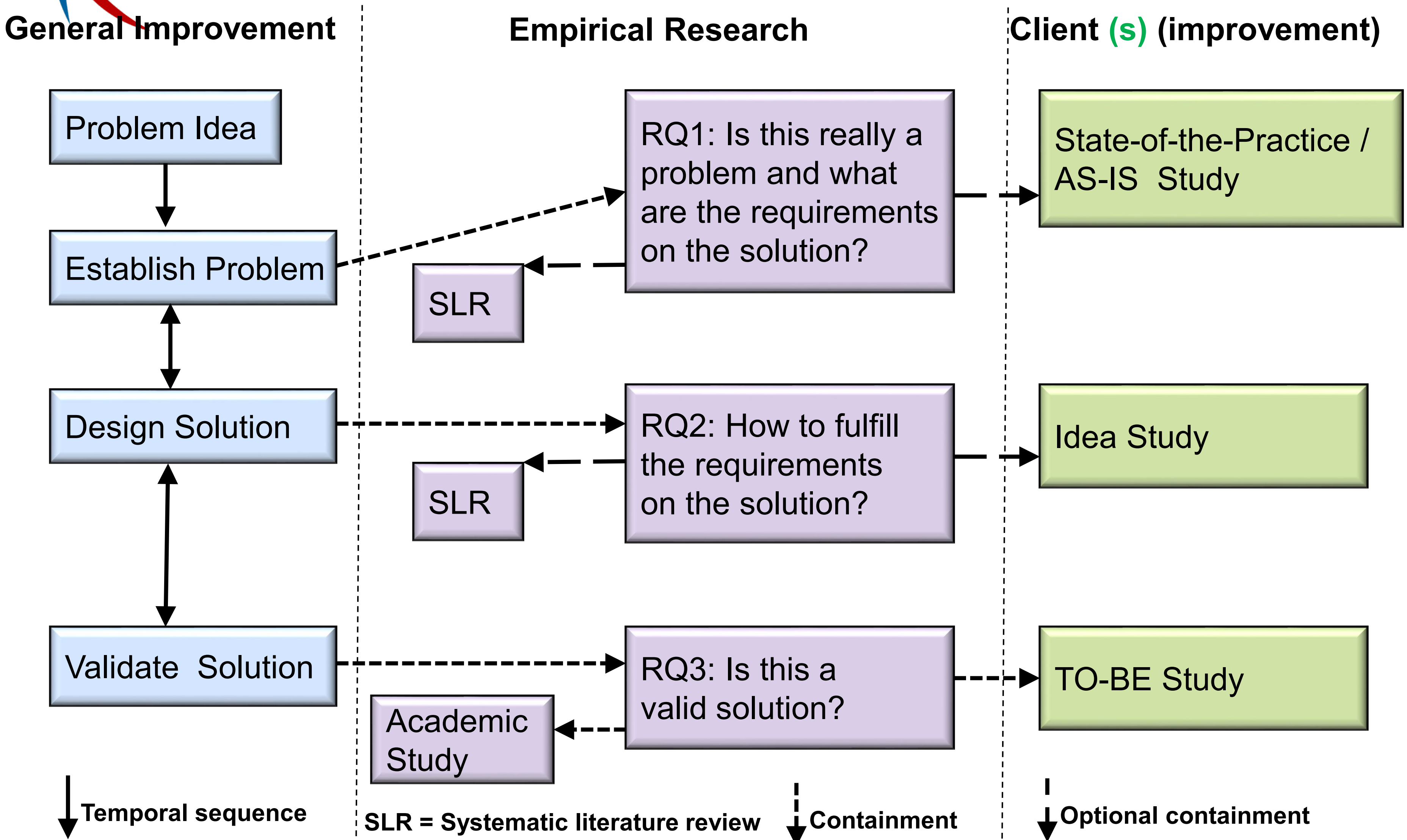




[Wieringa 2014]

Exp8: Full pilot project is often difficult

Our PhD approach: Combination of Small Studies



TO-BE Study

- The TO-BE study should apply the solution to a practice project.
- It involves static and dynamic analysis of the solution.
- If it is not possible that the client applies the solution in an ongoing project (moderated by the researcher), the researcher applies the solution
  - In an ongoing project OR
  - Retrospectively on past project data OR
  - In a simulation extrapolating the ongoing project

Exp9: Documented project data often not sufficient for retrospective validation, especially for a method with many human activities

Simulated method application

- 3 steps of the simulated method application
  - AS-IS study of the actual project
    - Understand the status wrt. the problem (how urgent is the problem)
    - Understand the status wrt potential solution (how easy is it to apply the solution)
  - Sketch the method application on the actual project data (changing the actual project as little as possible)
  - Discuss the simulation with the project participants

Exp10: Application based on an ongoing project is more convincing than on „old“ project data.

software engineering heidelberg

Example PhD Ulrike Abelein

User-Developer Communication in Large Scale IT Projects

- Published in ICSE Chase, REFSQ and Empirical Software Engineering Journal

- Problem from own experience in industry
- Solution is a method

Establish Problem and First Design Ideas

Part II - Problem Investigation

Chapter 3 - Understanding the Influence of UPI System Success (State-of-the-Art)

RQ1: Does increased user participation and involvement (UPI) leads to increased system success? (Knowledge Problem)

Results: Meta-analysis on empirical evidence on the effect of UPI on system success

SLR

RQ2: What are the characteristics of methods aiming to increase UPI in software development? (Knowledge Problem)

Results: Analysis of existing methods

Chapter 4 – User-Developer Communication in Large-Scale IT Projects

RQ3: How and how well is user-developer communication supported in large-scale IT projects (with a focus on the decisions which are made in the design and implementation phase and their rationale)? (Knowledge Problem)

Results: State-of-practice of UDC in large-scale IT projects

State-of-the Practice Study

Chapter 5 – A Descriptive Classification for End User-Relevant Decisions of Large-Scale IT Projects

RQ4: What are user-relevant decisions in the design and implementation phase? (Knowledge Problem)

Results: Descriptive classification of user-relevant decisions

Design Solution

Part III - Treatment Design

Chapter 6 – Requirement for the UDC-LSI Method

Results: Conceptual framework and requirements for the UDC - LSI Method

Chapter 7 – The UDC-LSI Method to Enhance User-Developer Communication in Large-Scale IT Projects

Results: UDC - LSI Method for large-scale IT projects using traditional methods in customer-specific software development to increase system success

First Design Validation and Implementation Validation

Part IV - Design Validation & Implementation Evaluation

Chapter 8 - Expert Assessment of UDC-LSI Method - Results of an Interview Series with Practice Experts

Idea Study

RQ5: What is the potential of the UDC-LSI method to improve system success? (Knowledge Problem)

Results: Design validation incl. benefits and obstacle for implementation of UDC-LSI method

Chapter 9 - Evaluation of the UDC-LSI method – the iPeople Case Study

TO-BE Study

RQ6: What effects with regards to usability and utility has the UDC-LSI in large-scale IT project? In particular:? (Knowledge Problem)

Results: Confirmation of feasibility, effectiveness, efficiency, and acceptance, of UDC-LSI method

Simulated application

Many further questions

- Wieringas book gives very good advice on how to do the empirical work, however…
- How to scope the SLR?
  - balance research question, search terms and amount of papers
- How to do the AS-IS study, if client has no time?
  - Similar to problems in requirements elicitation for software….
- How to describe a method in detail?
  - Similar to problems in requirements specification and validation
    - How to get judgement of future users before they can use the software
- Which criteria describe the validity of the solution?
  - checklist
- How to consider which threats to validity?
  - checklist

Quality of the solution

- Many different terms: utility, usability, acceptance,…
- We use
  - Feasibility: can the solution really be applied in practice (by other people)?
  - Effectiveness: does the solution application lead to the required effects?
  - Efficiency: is the overhead by the solution application worth the effect?
  - Acceptance: do the practitioners accept the solution?
    - E.g. using Technology Acceptance Model (TAM)
      - Perceived ease of use, perceived usefulness, attitude towards using,behavioral intention towards using

Summary

- Design Science research is important for an SE PhD
- Complete technology transfer often not possible
- Distinguish improvement and research
- Combine different small studies for different purposes
- If unavoidable, validate solution partially (e.g. through simulation)

- It is difficult to generalize from individual PhDs….