Design Science Research Methods

Prof. Dr. Roel Wieringa
University of Twente,
The Netherlands
www.cs.utwente.nl/~roelw

Research methodology accross the disciplines

- Do these disciplines have the same methodology?
  - Technical science: Build cool stuff; test it; iterate
  - Social science: Observe people, interpret what they do or say; or select a sample, do a lot of statistics; iterate.
    - *For social scientists, engineers are slightly autistic tinkerers*
    - *For technical scientists, social scientists are chatterboxes*
  - Physical science: Build instruments, create phenomena, analyze data, create theories; iterate.
    - *For physicists, other sciences are like stamp collecting*
  - Mathematics: Read, think, write, think; iterate.
    - *Mathematicians think that they provide the foundations of civilization*

Our approach

- All research in all disciplines is **problem-solving**
- The problems in design science research are **design problems**
  - Goal is to design something useful
  - Research method is the design cycle
- The problems in empirical research are **knowledge questions**
  - Goal is to acquire theoretical knowledge
  - Research method is the empirical cycle

- Wieringa, R.J. (2014) *Design science methodology for information systems and software engineering.* Springer Verlag

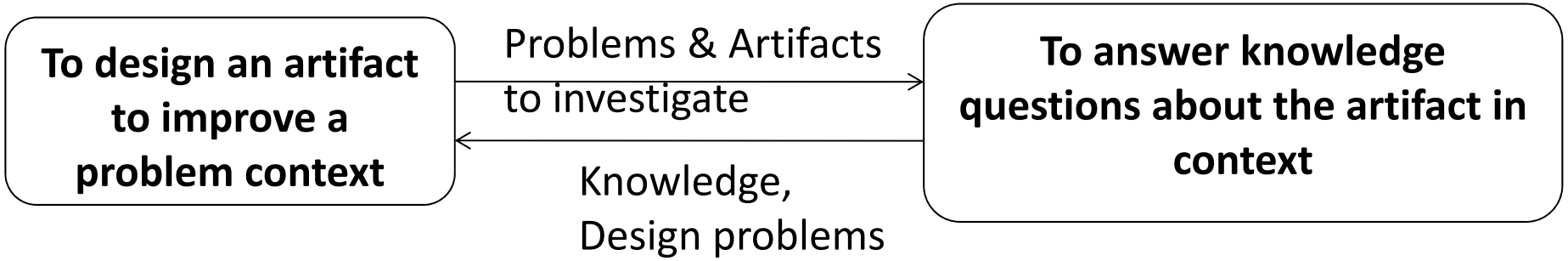
Outline

1. What is design science?
2. Research goals and problems
3. The design and engineering cycles
4. The empirical cycle

What is design science?

- Design science is the **design** and **investigation** of artifacts in context

Two kinds of research problems in design science



- *Design software to estimate Direction of Arrival of plane waves, to be used in satellite TV receivers in cars*
- *Design a Multi-Agent Route Planning system to be used for aircraft taxi route planning*
- *Design a data location regulation auditing method*

Is the artifact **useful**?

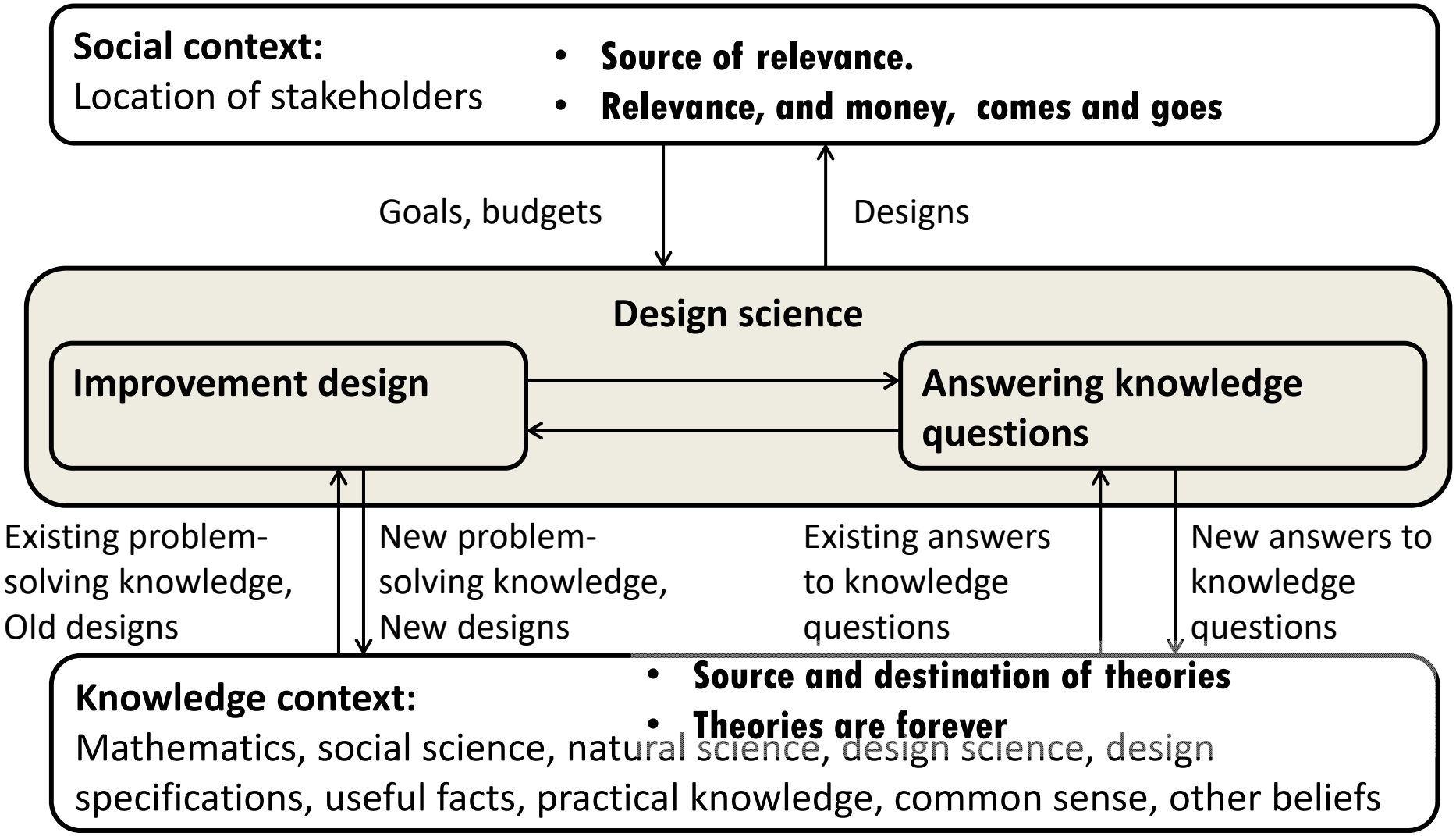
- *Is the DoA estimation accurate enough`in this context?*
- *Is it fast enough?*
- *Is this routing algorithm deadlock-free on airports?*
- *How much delay does it produce?*
- *Is the method usable and useful for consultants?*

Is the answer **true**?

Reality check

- What research problem(s) are you investigating?
  - Artifact and context
- NB
  - The title of your thesis is the shortest summary of your research project.
  - Often, it mentions the artifact and the context.

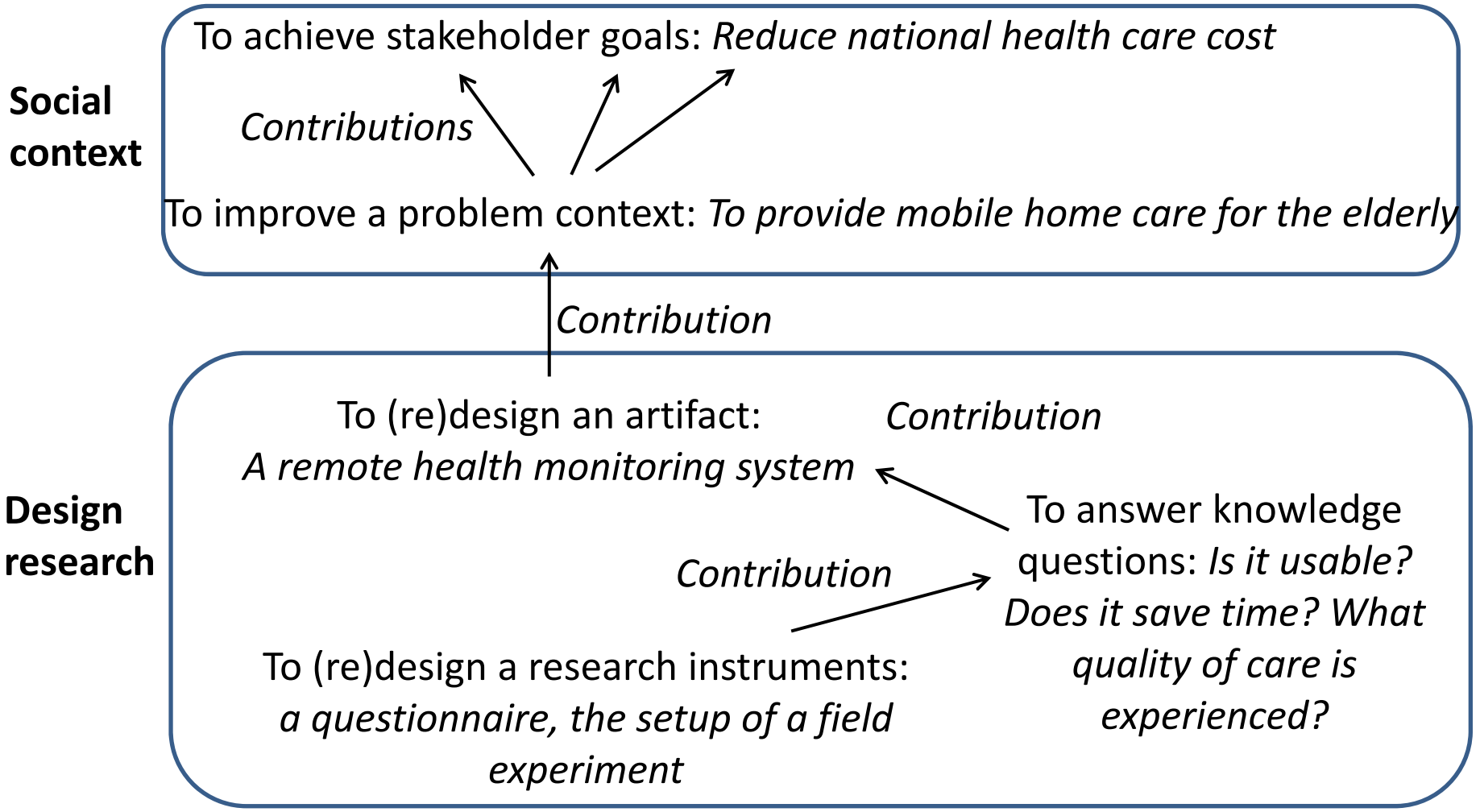
Framework for design science



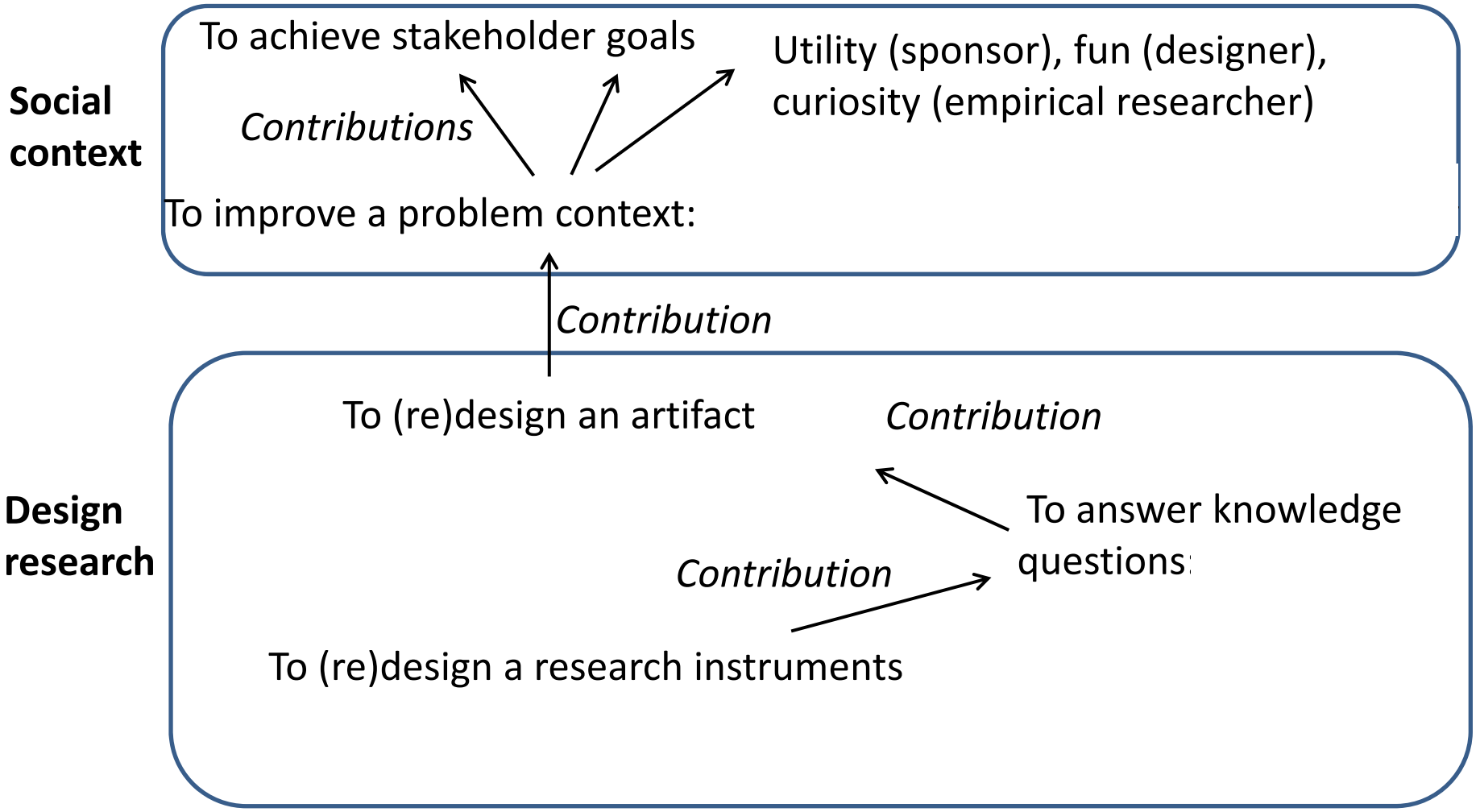
Outline

1. What is design science?
2. **Research goals and problems**
3. The design and engineering cycles
4. The empirical cycle

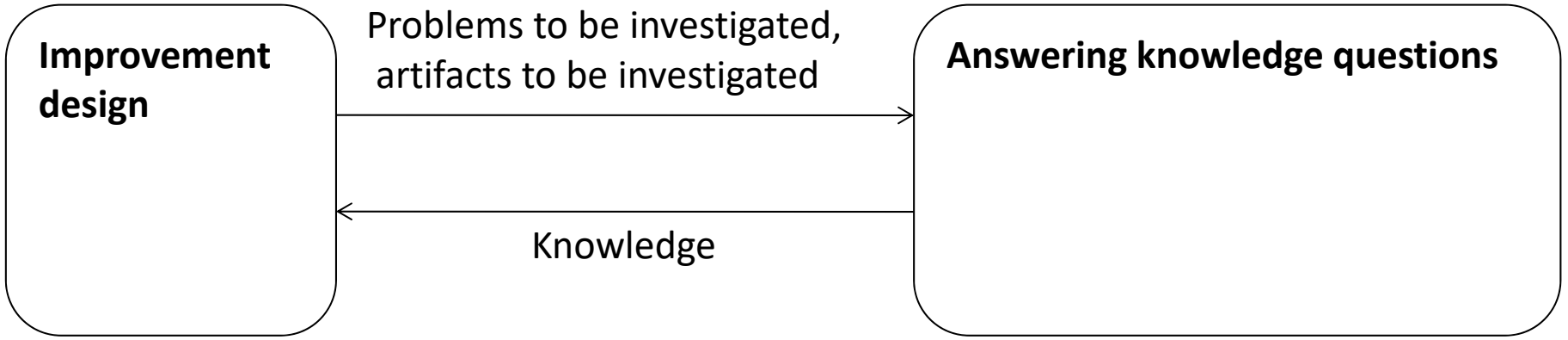
Goal structure: example



Goal structure: example



Three kinds of design research questions



1. **Design research problems** (a.k.a. *technical research questions*)
   - To improve some kind of artifact in some kind of context.
2. **Empirical knowledge questions**
   - To ask questions about the real world.
3. **Analytical knowledge questions**
   - To ask questions about the logical consequences of definitions

Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- *Reduce my headache*
- *by taking a medicine*
- *that reduces pain fast and is safe*
- *in order for me to get back to work*

Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- ***Reduce my headache***
- *by taking a medicine*
- *that reduces pain fast and is safe*
- ***in order for me to get back to work***

Problem context and stakeholder goals.

Stakeholder language

Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- *Reduce my headache*
- ***by taking a medicine***
- ***that reduces pain fast and is safe***
- *in order for me to get back to work*

Artifact and its desired properties.

Technical language

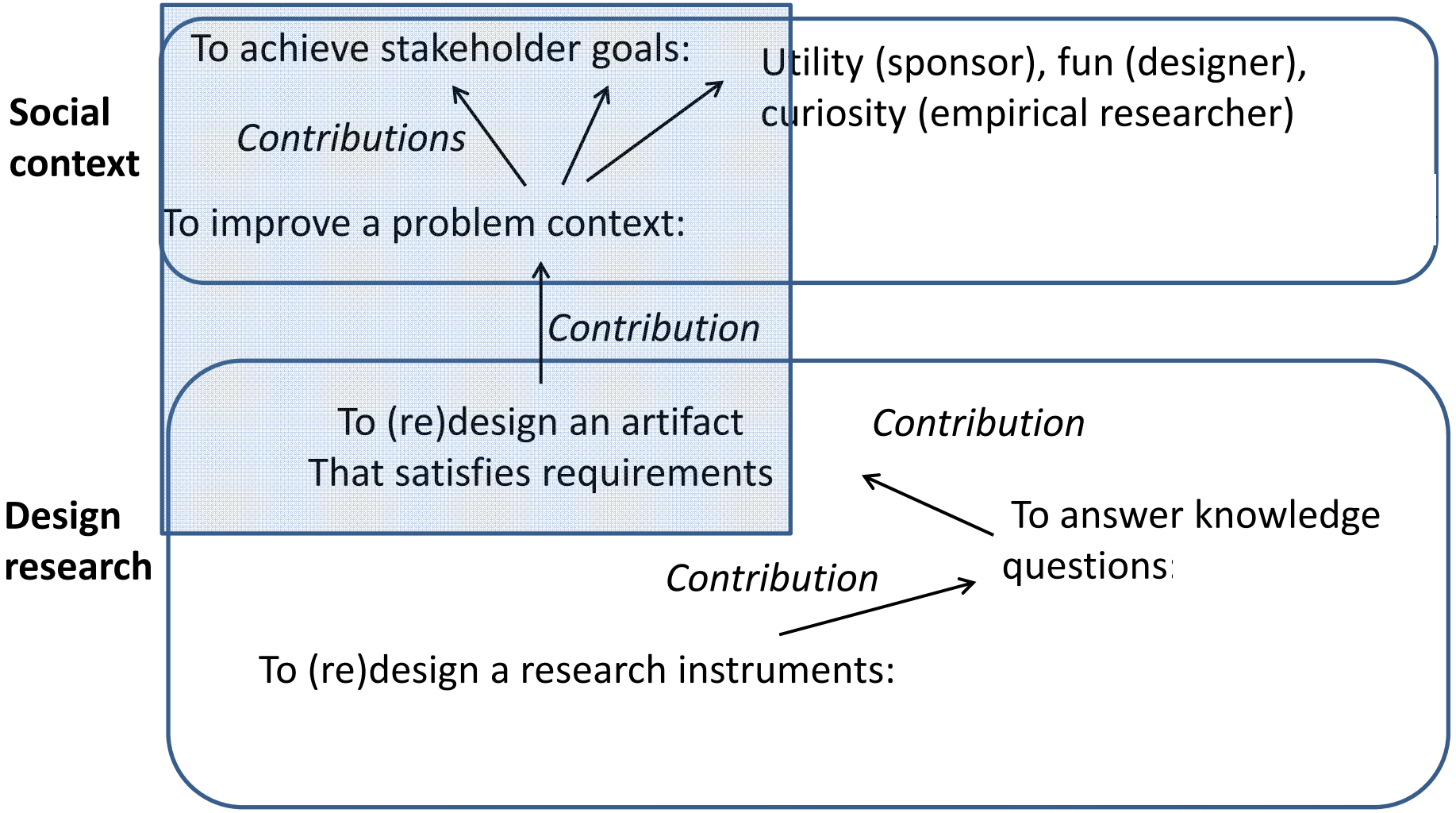
Also works for **research** problems rather than individual practical problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- *Reduce patients' headaches*
- *by treating it with a medicine*
- *that reduces pain fast and is safe*
- *in order for them to function as they wish*

The problem is now to design an artifact that helps a **class** of stakeholders achieve a **class** of goals.

- The design problem template relates the artifact to the problem context and stakeholder goals, and adds requirements



Discussion

- Who are the stakeholders of your project?
  - Real or hypothetical: Stakeholders may not know they are stakeholders
- What is/are your top-level design problem(s), using our template?
  - **Improve <problem context>**
  - **by <treating it with a (re)designed artifact>**
  - **such that <artifact requirements>**
  - **in order to <stakeholder goals>**

- NB some parts may be *currently* uncertain, fuzzy, or unknown.
- But surely, some parts are currently known!

There is no single “correct”problem statement

- A good problem statement forces the reader to think focussed about the artifact while remaining aware of the intended problem context

- Next two examples extracted from two M.Sc theses
  - http://essay.utwente.nl/67945/
  - http://essay.utwente.nl/69399/

- *BPMN Plus : a modelling language for unstructured business processes.* ← Artifact / ← Context
- *The objective of this study is*
  - *To investigate the way through which unstructured business processes can be modelled and managed without limiting their run-time flexibility.*
- *Research questions*
  - *Q1 What are the differences between structured and unstructured business processes?*
  - *Q2 What are the differences between Business Process Management and Case Management in dealing with unstructured business processes?*
  - *Q3 What are the capabilities of existing modelling notations to deal with unstructured business processes?*
  - *Q4 How to model an unstructured business process while providing run-time flexibility?*

- **Improve <problem context in which unstructured business process is to be modelled>**
- **by <introducing a modeling language for unstructured business processes>**
- **such that <requirements such as run-time flexibility, and ... learnability etc?>**
- **in order to <stakeholder goals, e.g. provide better process improvement advice to clients>**

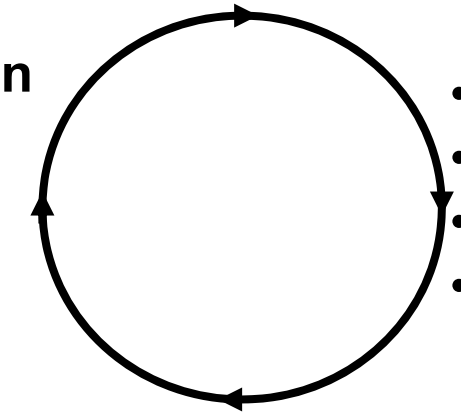
Outline

1. What is design science?
2. Research goals and problems
3. **The design and engineering cycles**
4. The empirical cycle

Engineering cycle

This is a checklist. See appendix A in the book & on my web site

! = Action
? = Knowledge question



Implementation evaluation = Problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

Design implementation

Treatment validation

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

Implementation is introducing the treatment in the intended problem context

- If problem context is a **real-world** context…. implementation of a solution is **technology transfer to the real world**.
  - Not part of a research project

- If the problem is to learn about the performance of a design ... Implementation of a solution is the **construction of a prototype and test environment**.
  - Part of a research project

Nesting of cycles

Research project: **design cycle**

Problem investigation	
Treatment design	
Treatment validation	Problem investigation (How to do the validation?)
	Experiment design & validation (design and validate a prototype & test environment)
	Implementation (construction of prototype & test environment, lab or field)
	Evaluation (analyze results)
Implementation (tech transfer)	
Implementation evaluation (in the field)	

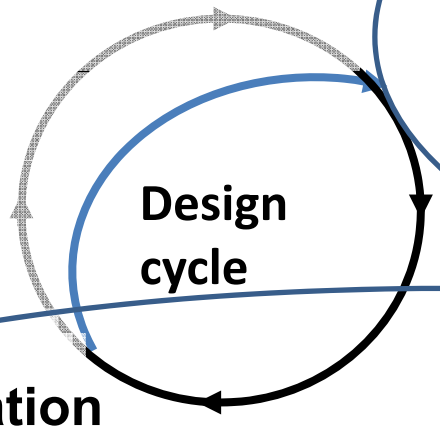
This is a very special engineering cycle, called the **empirical cycle**.

Questions?

Design cycle

Real-world problem-oriented research or evaluation research

Real-world design implementation



Real-world implementation evaluation = Real-world problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

Treatment validation

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

Solution-oriented research

Two kinds of design science research projects

- Problem-oriented research and evaluation research
  - Investigate the **real world** to learn about artifacts and how they are used by stakeholders
    - *How is the UML used in small and medium sized companies?*
    - *What is the cause if large SE projects being late?*
    - *How is RE done in large-scale agile projects?*
- Solution-oriened: technical research
  - Design an artifact, and validate it by simulation
    - *Design & validate a multi-agent system for autonomous route planning*
    - *Design & validate a system for remote health monitoring for the elderly*
    - *Design & validate a requirements engineering technique for agile global software engineering projects*

Example, missing question added

- *BPMN Plus : a modelling language for unstructured business processes.*
- *The objective of this study is*
  - *To investigate the way through which the unstructured business processes can be modelled and managed without limiting their run-time flexibility.*
- *Research questions*
  - *Q1 What are the differences between structured and unstructured business processes?*
  - *Q2 What are the differences between Business Process Management and Case Management in dealing with unstructured business processes?*
  - *Q3 What are the capabilities of existing modelling notations to deal with unstructured business processes?*
  - *Q4 How to model an unstructured business process while providing run-time flexibility?*
- *"The practical usefulness of newly proposed modelling notation is investigated by demonstrating it with the help of an example.*
- *Moreover, the proposed modelling notation is validated by conducting interviews with experienced practitioners."*

Problem

- *Stakeholders? Goals? : BiZZDesign consultants. To provide high-quality consultancy.*
- *Conceptual problem framework? Business process modelling, structured & unstructured. See Q1.*
- *Phenomena? Causes, mechanisms, reasons? BPMN does not allow for modelling flexible business processes; but case-management systems almost impose no constraints. Simple explanation: the languages lack facilities. See Q2.*
- *Effects? Positive/negative goal contribution? Limits to consultancy advice.*

Treatment

- *Specify requirements! Omitted research question. May be part of Q2.*
- *Requirements contribute to goals? Omitted too.*
- *Available treatments? See Q3.*
- *Design new ones! See Q4.*

Validation *Omitted questions, but done by means of interviews.*

- *Context & Artifact → Effects? Does it work?*
- *Effects satisfy Requirements? Does it work as desired?*
- *Trade-offs for different artifacts? Performance of different languages on similar cases?*
- *Sensitivity for different Contexts? Performance the designed language in different cases?*

Research questions reformulated (and renumbered)

Problem investigation

- *Q1 Who are the stakeholders, what are their goals, and what problems do they encounter when modeling unstructured business processes?*
- *Q2 How to define structured and unstructured business processes?*
- *Q3 What are the capabilities of BPM and CM systems to deal with unstructured processes?*

Treatment design

- *Q4 What are the requirements of the language? E.g., usability, utility?*
- *Q5 What are the capabilities of existing business process modelling notations to deal with unstructured business processes? How do they score on the requirements?*
- *Q6 Design a language to model unstructured business processes*

Treatment validation

- *Q7 Can the language model known and expected unstructured business processes?*
- *Q8 Does it satisfy the requirements? How does that compare the other available languages*

Sequence of design cycles to reduce uncertainty & manage cost and risk

- Design the product idea
  - Sketch the problem – design the principle of operation – analytical validation of soundness of the idea
- Sketch the product
  - Describe problem – sketch product architecture – provide argument that this exhibits the necessary mechanisms to produce desired behavior
- Feasibility study
  - Same, but now validate by building small prototype in test environment
- Specify the product
  - Describe problem mechanisms and goals – Specify product requirements and structure – validate analytically and empirically
- Etc.

Recap

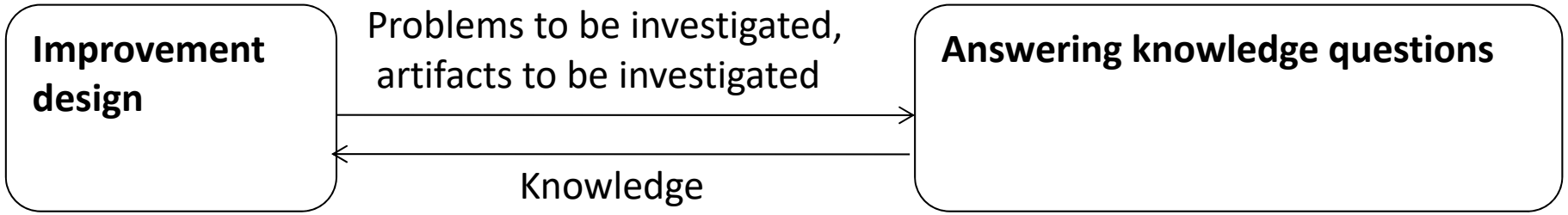
- Design science designs and investigates artifacts in context
  - Design problems versus knowledge questions
- Engineering cycle:
  - problem – design – validation – implementation – evaluation
- Design cycle:
  - problem – design – validation
  - Nesting of design cycles to solve subproblems
  - Sequence of design cycles to refine global design

Questions?

Outline

1. What is design science?
2. Research goals and problems
3. The design and engineering cycles
4. **The empirical cycle**

Research problems in design science



Design research problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>.

Design cycle

- Problem investigation
- Treatment design
- Treatment validation

2. Empirical knowledge questions

– To ask questions about the real world: about the problem or about the artifact in context.

3. Analytical knowledge questions

– Yields definitions, assumptions, theorems.

Empirical knowledge questions

- **Descriptive** knowledge questions:
  - What happened?
  - How much? How often?
  - When? Where?
  - What components were involved?
  - Who was involved?
  - Etc. etc.

Journalistic questions.
Yield **facts**.

- **Explanatory** knowledge questions:
  - Why?
    1. What has **caused** the phenomena?
    2. Which **mechanisms** produced the phenomena?
    3. For what **reasons** did people do this?

Beyond the facts.
Yields **theories**.

Three kinds of explanations: Example

- *Descriptive question: Is the light on?*
  - *Based on observation: Yes.*
  - *When? Now.*
  - *Where? Here.*
- *Explanatory question: Why is it on?*
  1. ***Cause:*** *because someone turned the light switch, it is on (and not off).* Explains difference with off-state.
  2. *Why does this cause the light to switch on?* ***Mechanism:*** *because the switch and light bulbs are connected by wires to an electricity source, in this architecture …, and these components have these capabilities …..* Explains how on-state is produced.
  3. *By why did someone turn the light on?* ***Reasons:*** *Because we wanted sufficient light to be able to read, and it was too dark to read.* Explains which stakeholder goal is contributed to.

Another example: software

- *Descriptive question: What is the performance of this program?*
  - *Execution time for different classes of inputs?*
  - *Memory usage?*
  - *Accuracy?*
  - *Etc. etc.*
- *Explanatory question: Why does this program have this performance (compared to others)?*
  1. ***Cause:*** *Variation in execution time is caused by variation in input; etc.*
  2. ***Mechanism:*** *Execution time varies this way because it has this architecture with these components*
  3. ***Reasons:*** *Observed execution time varies this way because users want to be on-line all the time, and therefore provide these inputs*

Another example: method

- *Descriptive question: What is the performance of this method for developing software?*
  - *Understandability for practioners*
  - *Learnability*
  - *Quality of the result*
  - *Perceived utility*
  - *Etc. etc.*
- *Explanatory question: Why does this method have this performance?*
  1. ***Cause:*** *Difference in understanding of methods by software engineers is attributed to differences in the methods.*
  2. ***Mechanism:*** *These differences are explained by the structure of the method and/or the structure of cognition.*
  3. ***Reasons:*** *No explanation in terms of reasons here.*

Research questions reformulated again

Problem investigation

- *Q1 Who are the stakeholders, what are their goals, and what problems do they encounter when modeling unstructured business processes?*
- *Q2 How to define structured and unstructured business processes?*
- *Q3 What are the capabilities of BPM and CM systems to deal with unstructured processes?*

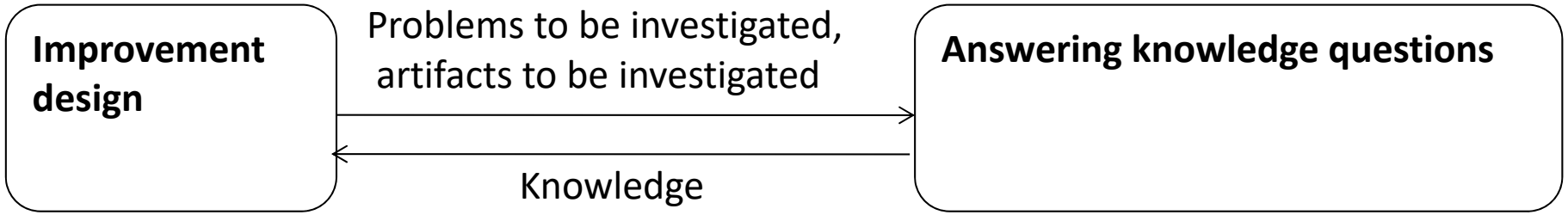
Treatment design

- *Q4 What are the requirements of the language?* ***Why?***
- *Q5 What are the capabilities of existing business process modelling notations to deal with unstructured business processes? How do they score on the requirements?*
- *Q6 Design a language to model unstructured business processes*

Treatment validation

- *Q7 Can the language model known and expected unstructured business processes?* ***Why (not)?***
- *Q8 Does it satisfy the requirements? How does that compare the other available languages*

Research problems in design science



Design research problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>.

Design cycle

- Problem investigation
- Treatment design
- Treatment validation

2. Empirical knowledge questions

- Descriptive: what, how, when, where, who, etc.→ **Facts**
- Explanatory: Why → **Theories**

3. Analytical knowledge questions

- Yields definitions, assumptions, theorems.

We want to develop theories of problems and of designs

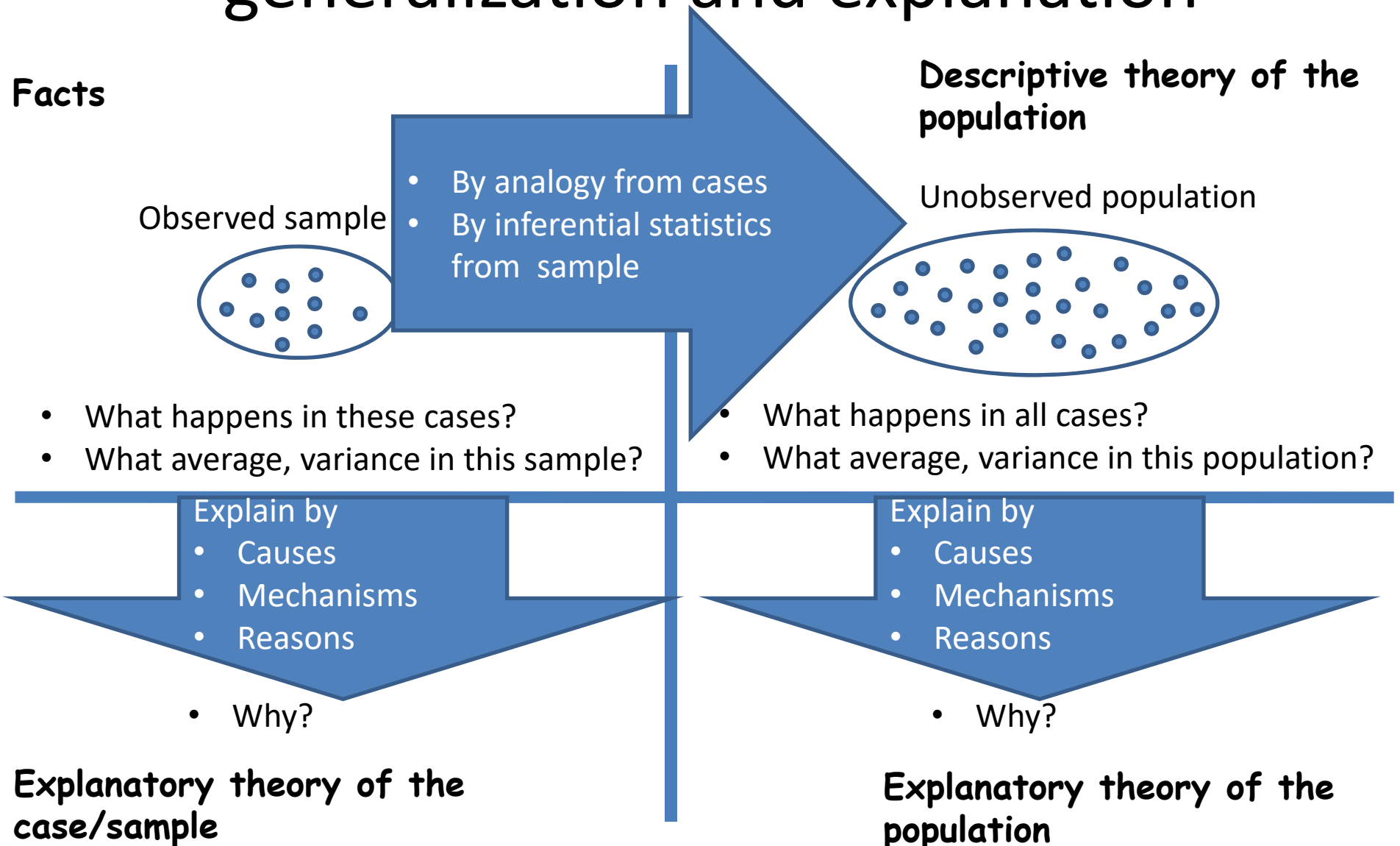
Example of a problem theory:

- *A theory of modeling of unstructured business processes*
  - *Scope of such a theory: the population of all cases in which unstructured business processes are modeled.*

Example of a design theory:

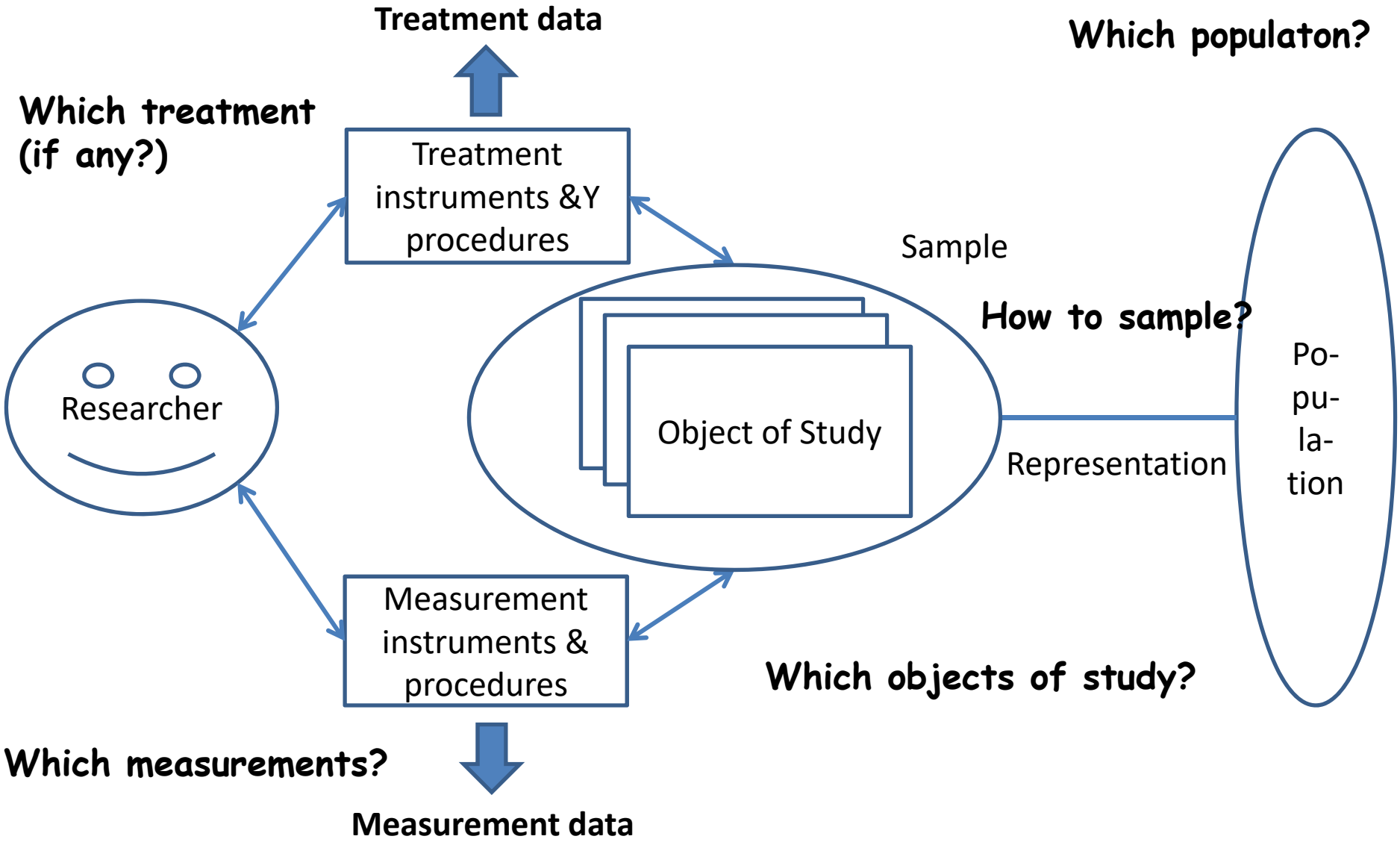
- *A theory of a particular notation for modeling unstructured business processes*
  - *Scope of such a theory: the population of all cases in which this notation is used to model an unstructured business process*

Two way to go beyond facts: generalization and explanation



To support generalization and explanation,
we need sound empirical research design

Design decisions for research setup



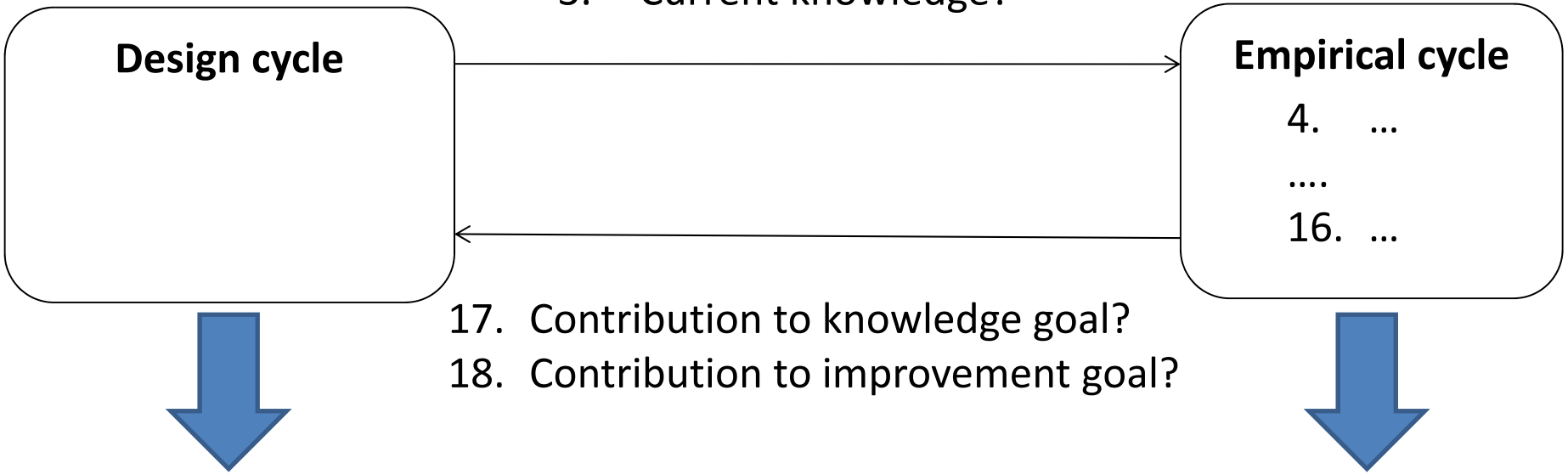
Research designs

	Observational study (no treatment)	Experimental study (treatment)
Case-based: investigate single cases, look at architecture and mechanisms	**Observational case study**	• **Expert opinion** (mental simulation by experts), • **Mechanism experiments** (simulations, prototyping), • **Technical action research** (experimental use of the artifact in the real world)
Sample-based: investigate samples drawn from a population, look at averages and variation	**Survey**	• **Statistical difference-making experiment** (treatment group – control group experiments)

Next two slides: Single checklist for all of these research designs

Checklist to establish context

1. Improvement goal?
2. Knowledge goal?
3. Current knowledge?



Data analysis

12. Data?
13. Observations?
14. Explanations?
15. Generalizations?
16. Answers?

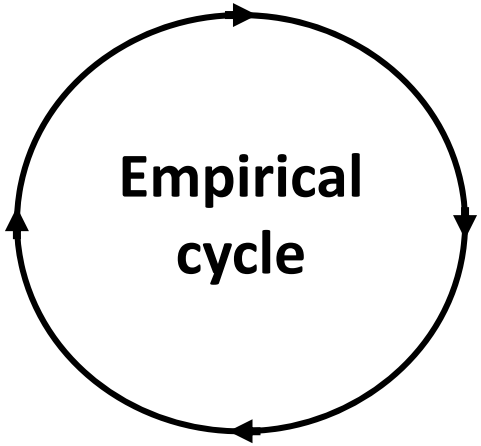
This is a checklist for

- **research design,**
- **research reporting,**
- **reading a report.**

App. B in my book & my web site

Research execution

11. What happened?

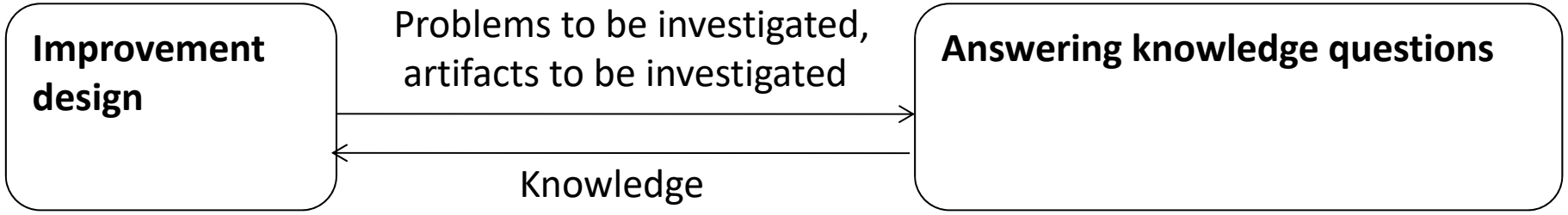


Research problem analysis

4. Conceptual framework?
5. Knowledge questions?
6. Population?

Design validation	**Research & inference design**	
7. Object of study validity?	7. Object of study?	Research setup
8. Treatment specification validity?	8. Treatment specification?	
9. Measurement specification validity?	9. Measurement specification?	
10. Inference validity?	10. Inference?	Inference

Summary



Design research problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>.

Design cycle

- Problem investigation
- Treatment design
- Treatment validation

Artifacts → Design cycle → Artefacts

Empirical knowledge questions

- Descriptive: what, how, when, where, who, etc.→ **Facts**
- Explanatory: Why → **Explanations**

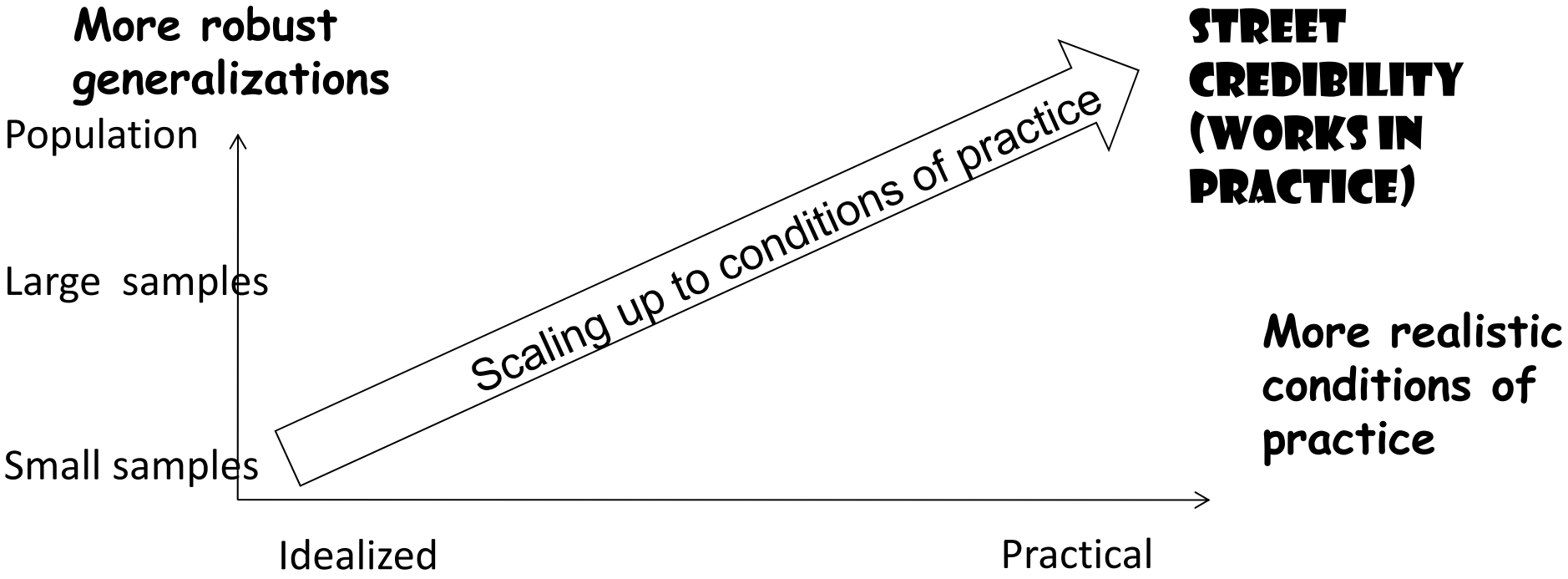
Empirical cycle

- Research problem analysis
- Research design & validation
- Research execution
- Data analysis

Theories → Empirical cycle → Theories

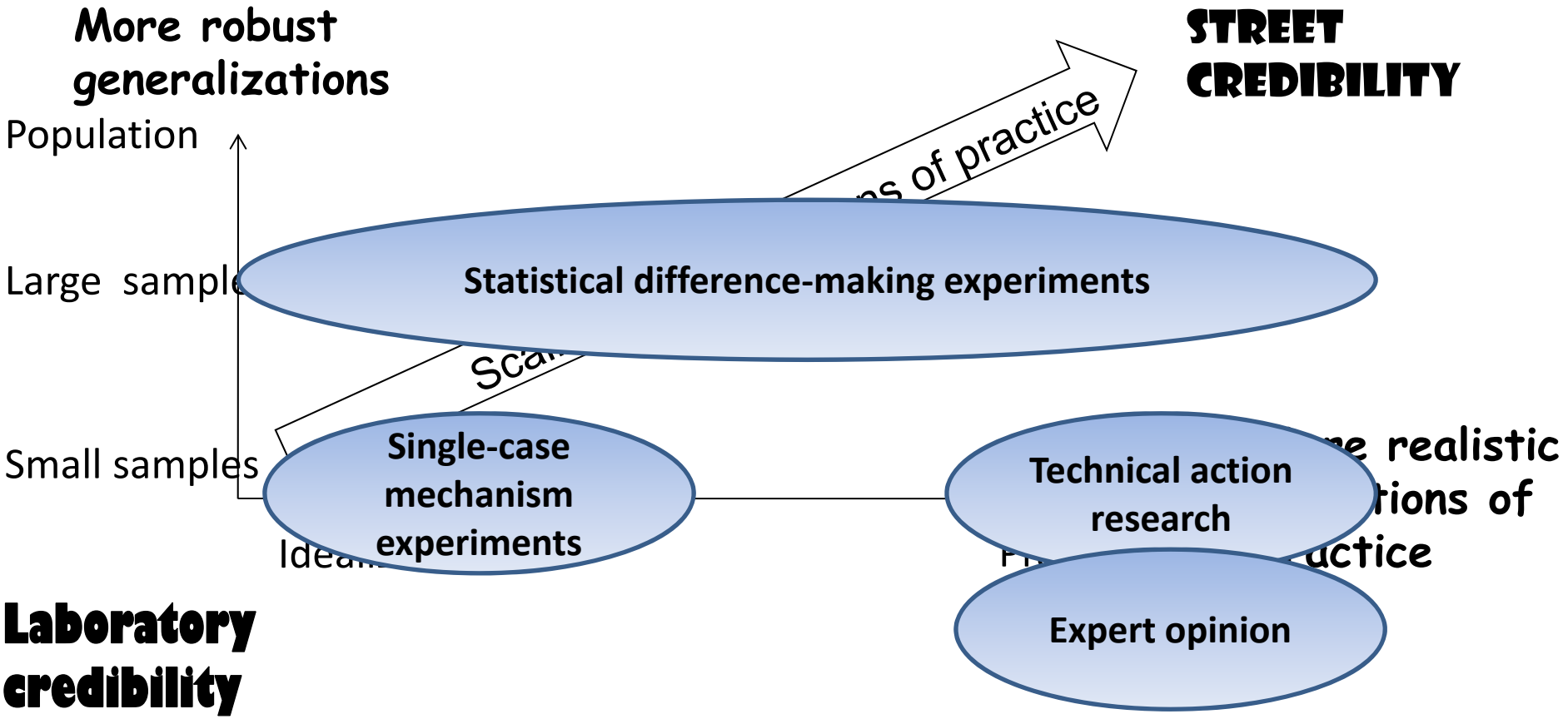
Analytical knowledge questions

Design science research strategy



Laboratory credibility (works in theory)

- Just like New Drug Research

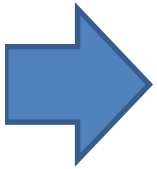


- Scaling up:
  - Single-case mechanism experiment (laboratory simulation)
  - Expert opinion
  - Single-case mechanism experiment (field simulation)
  - TAR (apply technique in a real-world project)

Take-home

- Design science designs and investigates artifacts in context
  - Design problems versus knowledge questions
- Solve design problems with design cycle:
  - Problem investigation – treatment design – treatment validation
  - Nesting and sequencing of design cycles
  - → Useful artifacts for a context
- Answer empirical knowledge questions with the empirical cycle
  - Research problem investigation – research design – validation – execution – analysis
  - Case-based or sample-based designs, observational or experimental designs
  - → Theories about artifact in context
- Research strategy: Scaling up from lab to practice

- Wieringa, R.J. and Daneva, M. (2015) *Six strategies for generalizing software engineering theories.* Science of computer programming, 101. pp. 136-152.



Wieringa, R.J. (2014) *Design science methodology for information systems and software engineering.* Springer Verlag

- Wieringa, R.J. (2014) *Empirical research methods for technology validation: Scaling up to practice.* Journal of systems and software, 95. pp. 19-31.

- Wieringa, R.J. and Morali, A. (2012) *Technical Action Research as a Validation Method in Information Systems Design Science.* In: *Design Science Research in Information Systems. Advances in Theory and Practice 7th International Conference*, DESRIST 2012, 14-15 May 2012, Las Vegas, USA. pp. 220-238. Lecture Notes in Computer Science 7286. Springer.

- Wieringa, R.J. (2010) *Relevance and problem choice in design science.* In: *Global Perspectives on Design Science Research (DESRIST). 5th International Conference*, 4-5 June, 2010, St. Gallen. pp. 61-76. Lecture Notes in Computer Science 6105. Springer.

- Wieringa, R.J. (2009) *Design Science as Nested Problem Solving.* In*: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, Philadelphia. pp. 1-12. ACM.