Design science research methodology

in information systems and software systems engineering

Prof. dr. Roel Wieringa
University of Twente
The Netherlands
https://wwwhome.ewi.utwente.nl/~roelw/

- Motivation
- Design problems versus knowledge questions
- The design cycle
- Design theories
- The empirical cycle

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*
    - *For physicists, physics is the foundation of engineering*
  - 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**
- Problems solved in rational problem solving cycle
  - Critical investigation of alternatives
  - Confrontation with facts

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

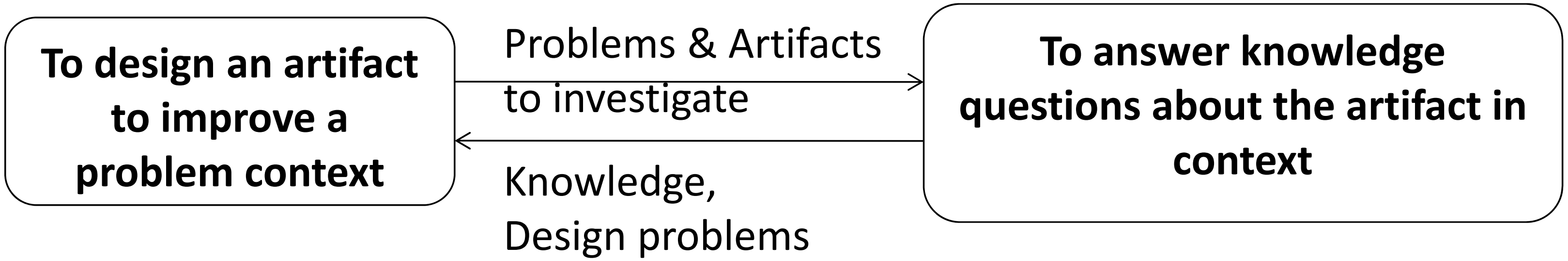
Why are we doing this?

- For senior researchers: how to compete with other disciplines for funds?
- For students: How to structure my thesis?

- How to **justify** your research goals and research results?

- Motivation
- Design problems versus knowledge questions
- The design cycle
- Design theories
- The empirical cycle

Two kinds of research problems in design science



Design software to estimate Direction of Arrival of plane waves, to be used in satelite TV receivers in cars	• *Is the DoA estimation accurate enough in this context?* • *Is it fast enough?*
Design a Multi-Agent Route Planning system to be used for aircraft taxi route planning	• *Is this routing algorithm deadlock-free on airports?* • *How much delay does it produce?*
Design a data location regulation auditing method	• *Is the method usable and useful for consultants?*

Is the artifact **useful** in this context?

Is the answer about the artifact in context **true**?

Question

- What research problem(s) are you investigating?
  - Artifact and context

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*

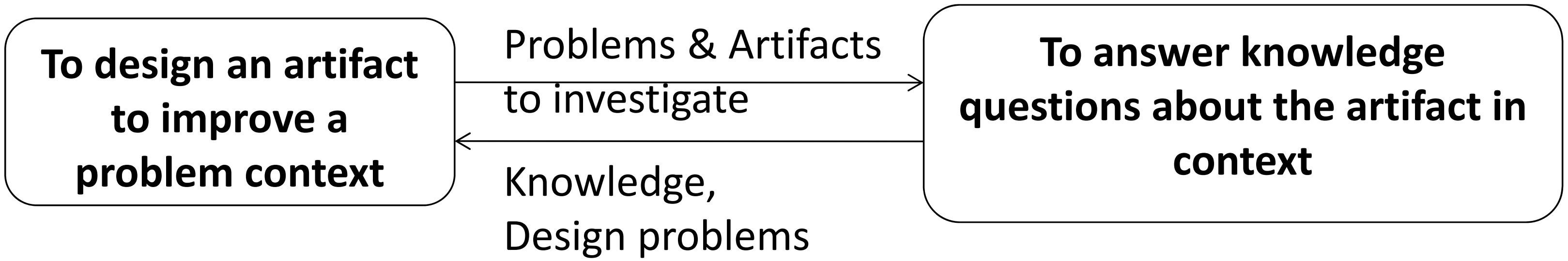
Empirical knowledge questions

- **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.



- **Curiosity/fun -driven science** starts with a knowledge question …
- … and continues with instrument design
- **Utility-driven science** starts with an improvement need of stakeholder …
- … and continues with artifact design or with a knowledge question

- Sponsors are always utility-driven
- Researchers are always curiosity and/or fun-driven

We design and study **artifacts in context**

- Reality check: What is/are the artifacts and what is/are the context(s)?
  - SIKS dissertations http://www.siks.nl/dissertations.php
  - Master theses in business informatics http://essay.utwente.nl/view/programme/60025.html
  - Master theses in computer science http://essay.utwente.nl/view/programme/60300.html
  - Master theses in human-media interaction http://essay.utwente.nl/view/programme/60030.html

- Motivation
- Design problems versus knowledge questions
- The design cycle
- Design theories
- The empirical cycle

Engineering cycle

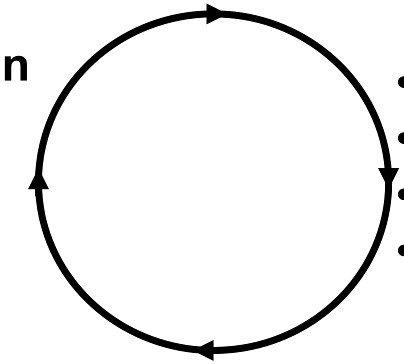
This is a checklist. See appendix A in the book & on my web site

! = Action
? = Knowledge question

Implementation evaluation = Problem investigation

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

Treatment implementation



Treatment design

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

Treatment validation

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

Implementation is introducing the treatment in the intended problem context

- If the problem is to improve 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, and using it**.
  - Part of a research project

Nesting of cycles

Research project: **design cycle**

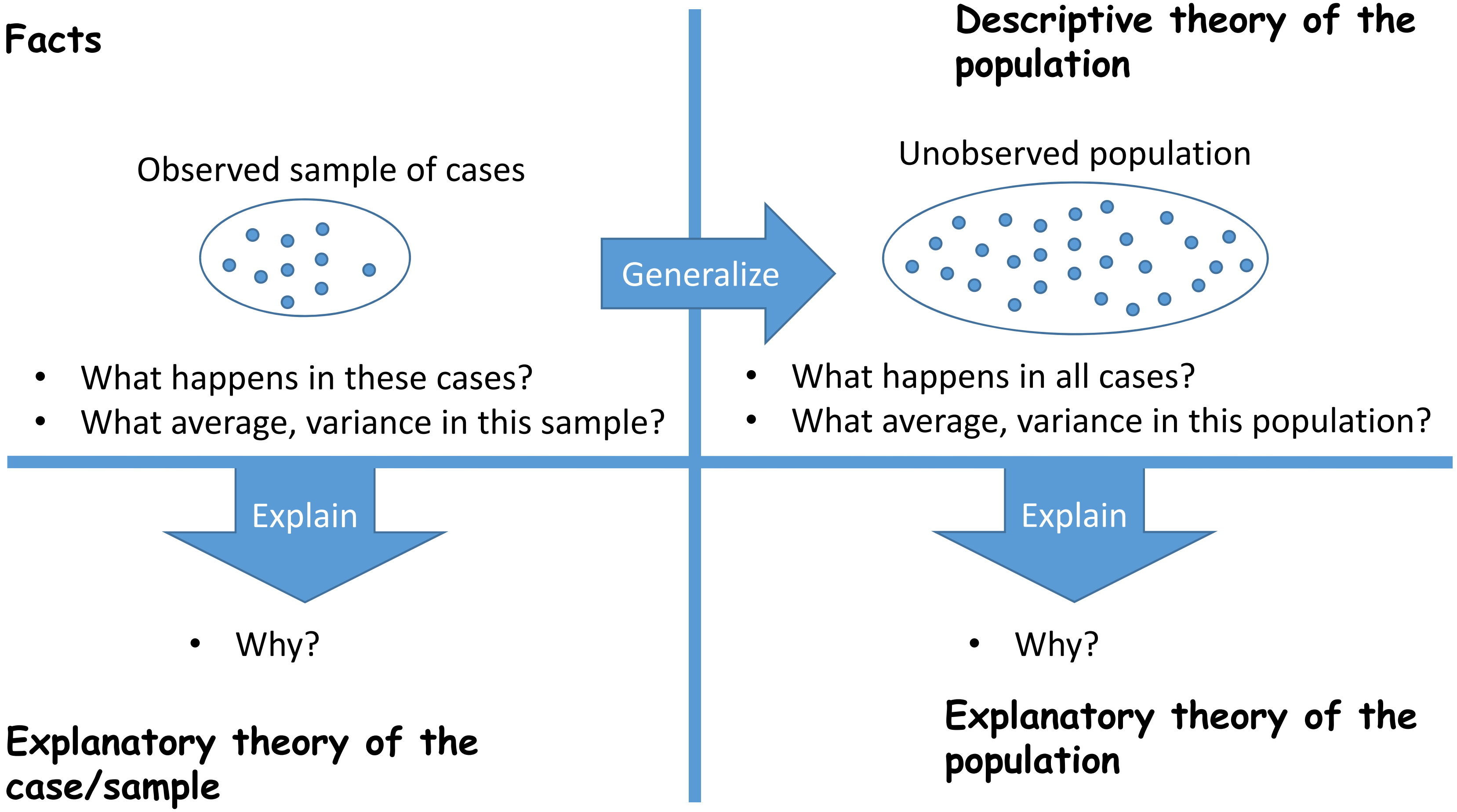
Real-world problem investigation	
Treatment design	
Treatment validation	Problem investigation (How to do the validation?)
	Design a prototype & test environment
	Validate a prototype & test environment
	Implement prototype & test environment (lab or field)
	Evaluation (analyze results)
Real-world implementation (tech transfer)	
Real-world implementation evaluation (in the field)	

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

- Do you recognize the structure of your thesis?

- Motivation
- Design problems versus knowledge questions
- The design cycle
- Design theories
- The empirical cycle

From facts to theories



What is a theory?

- A **theory** is a belief that there is a pattern in phenomena.
  - Idealizations: *“Merging two faculties reduces cost.” “This works in theory, but not in practice.”*
  - Speculations: *“The NSA is monitoring all my email.”*
  - Opinions: *“The Dutch lost the soccer competition because they are not a team.’’*
  - Wishful thinking: *``My technique works better than the others.’’*
  - **Scientific theories*:*** *Theory of electromagnetism*

Scientific theories

- A **scientific** theory is a belief that there is a pattern in phenomena, that has survived
  - Tests against experience:
    - Observation, measurement
    - Possibly: experiment, simulation, trials
  - Criticism by critical peers:
    - Anonymous peer review
    - Publication
    - Replication

- Examples
  - *Theory of electromagnetism*
  - *Technology acceptance model*
  - *Theory of the UML*

- Non-examples
  - *Religious beliefs*
  - *Political ideology*
  - *Marketing messages*
  - *Most social network discussions*

Scientific **design theories**

- A **scientific design theory** is a belief that there is a pattern in the interaction between an artifact and its context

- Examples:
  - *Theory of the UML in software engineering projects*
  - *Theory of your design in the intended problem context*

Design theory

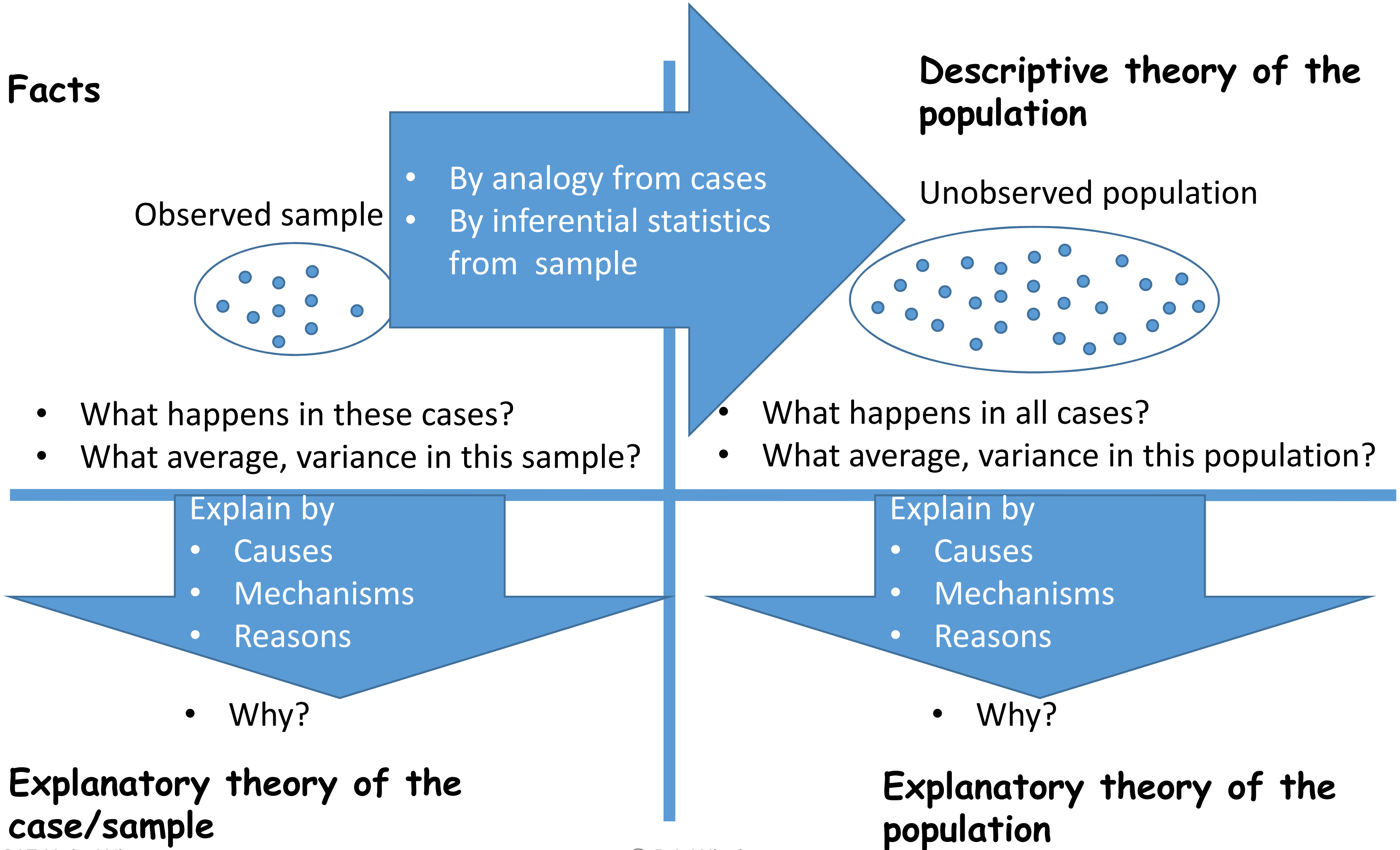
Theory of an algorithm

- *Concepts: definitions of concepts to specify a direction-of-arrival recognition algorithm, and of concepts to describe antenna array, and of accuracy and excution time*
- *Descriptive generalization: (Algorithm MUSIC) x (antenna array, plane waves, white noise) → (execution time less than 7.2 ms.)*
- *Explanatory generalization: qualitative explanation by analysis of the algorithm.*

Another design theory

- *Descriptive UML theory*
  - *Concepts: UML concepts, definitions of software project, of software error, project effort.*
  - *Descriptive generalization: (UML) X (SE project) → (Less errors, less effort than similar non-UML projects)*

- *Explanatory UML theory:*
  - *Concepts: definition of concept of domain, understandability*
  - *Explanatory generalizations:*
    - *UML models resemble the domain more than other kinds of models;*
    - *they are easier to understand for software engineers;*
    - *So they they make less errors and there is less rework (implying less effort).*


Facts
Observed sample
By analogy from cases
By inferential statistics from sample
Descriptive theory of the population
Unobserved population
What happens in these cases?
What average, variance in this sample?
What happens in all cases?
What average, variance in this population?
Explain by
Causes
Mechanisms
Reasons
Explain by
Causes
Mechanisms
Reasons
Why?
Why?
Explanatory theory of the case/sample
Explanatory theory of the population

- Motivation
- Design problems versus knowledge questions
- The design cycle
- Design theories
- The empirical cycle

Data analysis

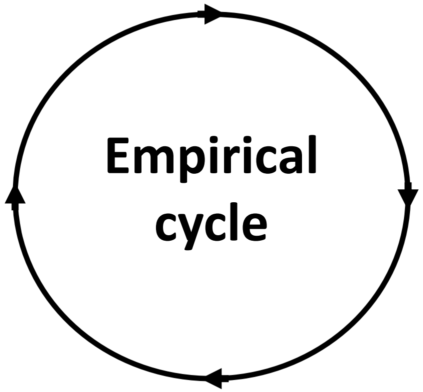
12. Descriptions?
13. Statistical conclusions?
14. Explanations?
15. Generalizations?
16. Answers?

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?

Research & inference design

Research setup

7. Objects of study?
8. Treatment specification?
9. Measurement specification?

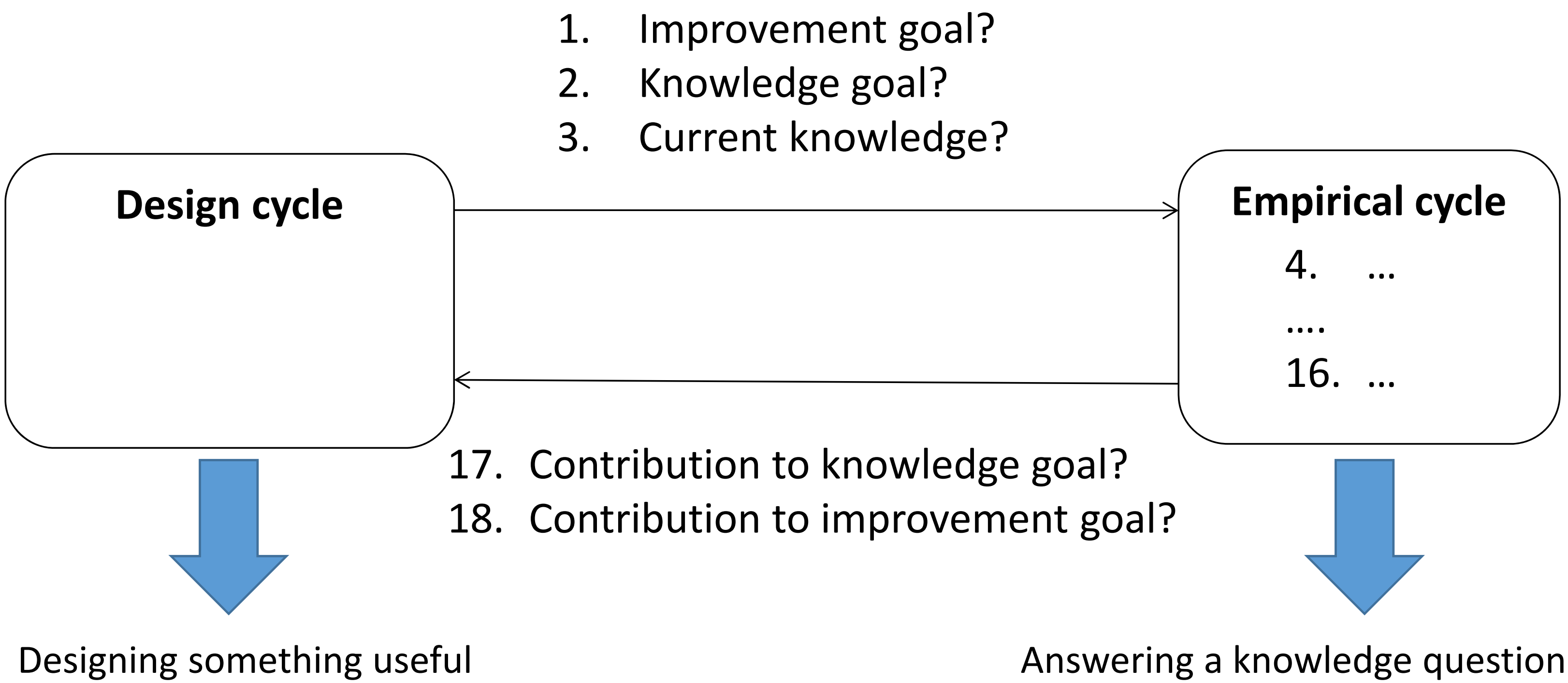
Inference

10. Inference?

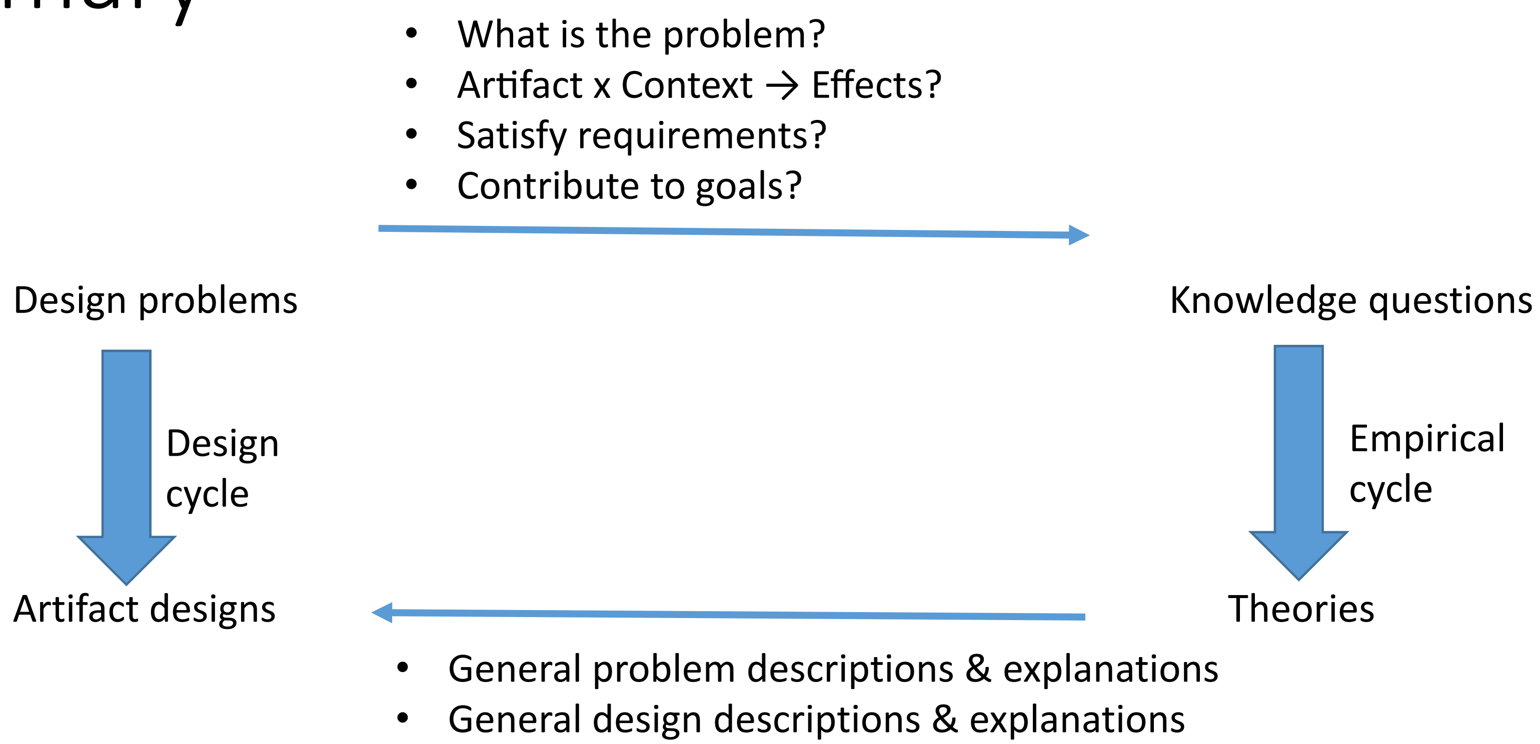
Design validation

7. Objects of study validity?
8. Treatment specification validity?
9. Measurement specification validity?
10. Inference validity?

Checklist for research design: context



Summary


• What is the problem?
• Artifact x Context → Effects?
• Satisfy requirements?
• Contribute to goals?
Design problems
Knowledge questions
Design cycle
Empirical cycle
Artifact designs
Theories
• General problem descriptions & explanations
• General design descriptions & explanations