Research Methods in Computer Science

(Serge Demeyer — University of Antwerp)

Universiteit Antwerpen

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Helicopter View

(Ph.D.)
Research

How to *perform* research?
(and get “empirical” results)

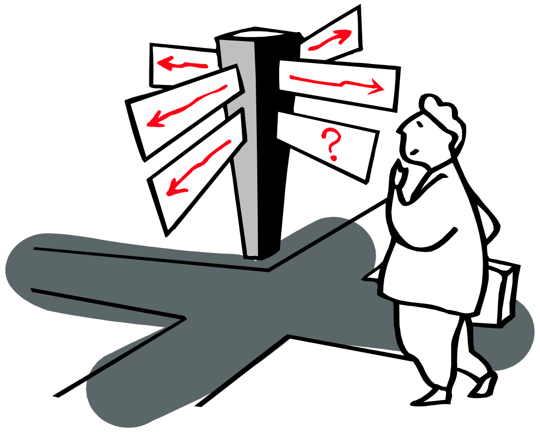
How to *write* research?
(and get papers accepted)

How many of you have
done / will do a case-study?


Zürich Kunsthaus


Antwerp Middelheim

1. Research Methods



Introduction

- Origins of Computer Science
- Research Philosophy

Research Methods

- 1. Feasibility study
- 2. Pilot Case
- 3. Comparative study
- 4. Observational Study [a.k.a. Etnography]
- 5. Literature survey
- 6. Formal Model
- 7. Simulation

Conclusion

- Studying a Case
  vs. Performing a Case Study
  + Proposition
  + Unit of Analysis
  + Threats to Validity

Computer Science

All science is either physics or stamp collecting (E. Rutherford)

We study artifacts produced by *humans*

Computer science is no more about computers than astronomy is about telescopes. (E. Dijkstra)

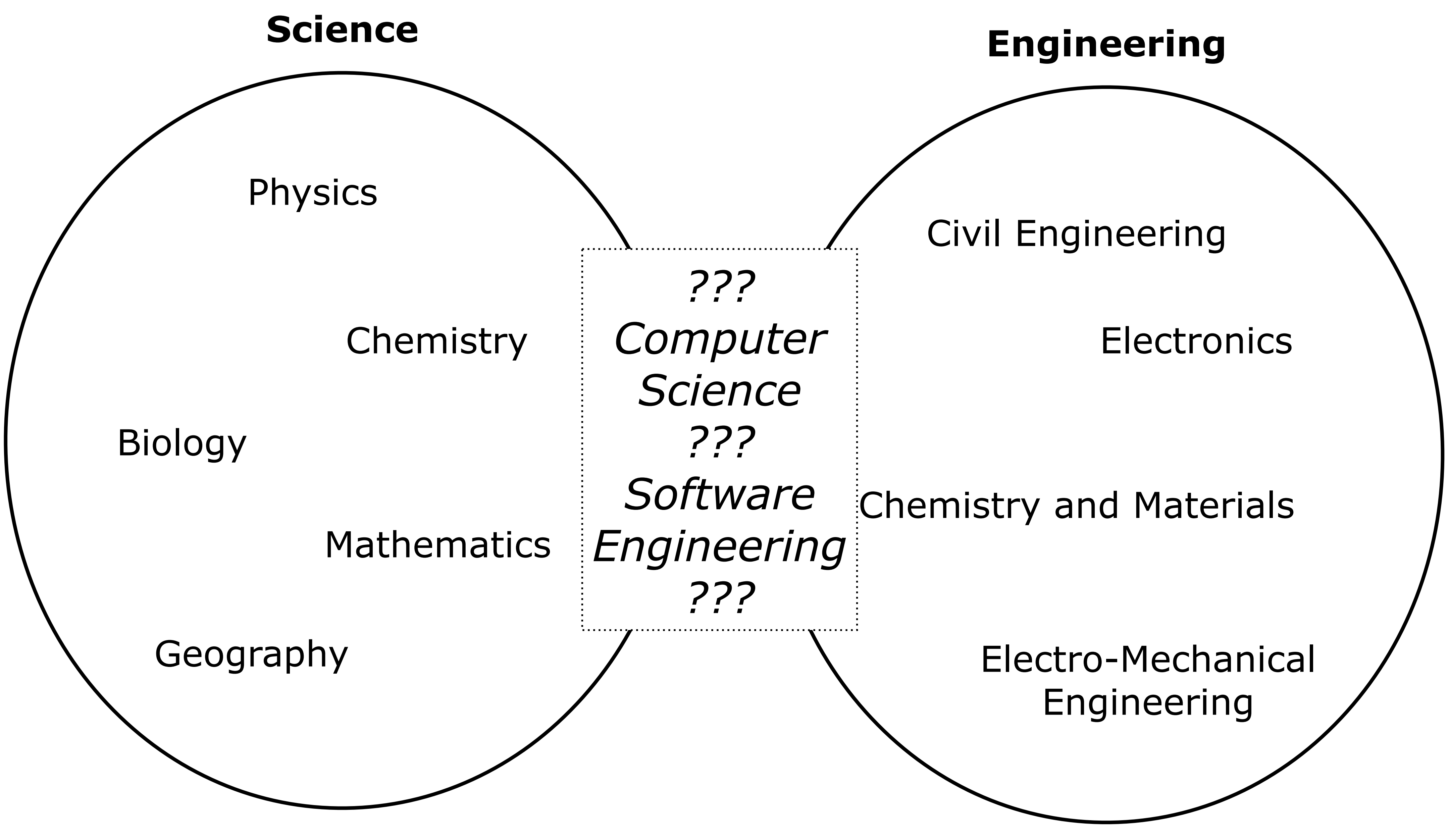
Computer science

Computer engineering

Informatics

Software Engineering

Science vs. Engineering



Mathematical Origins

Turing Machines
- Halting problem

Algorithmic Complexity
- P =? NP

Compilers
- Chomsky hierarchy

Databases
- Relational model

(inductive) Reasoning
- logical argumentation
  - + formal models, theorem proving, …
  - + axioms & lemma's
  - + foo, bar type of examples
- "deep" and generic universal knowledge

Gödel theorem: consistency of the system is not provable in the system.

⇒ A complete and consistent set of axioms for all of mathematics is impossible

Engineering Origins

Computer Engineering

- Moore's law: "the number of transistors on a chip will double about every two years"
  + Self-fulfilling prophesy
- Hardware technology
  + RISC vs. CISC
  + MPSoC
- Compiler optimization
  + peephole optimization
  + branch prediction

Empirical Approach

- Tom De Marco: "you cannot control what you cannot measure"
  + quantify
  + mathematical model
- Pareto principle
  + 80 % - 20 % rule (80% of the effects come from 20% of the causes)

As good as your next observation.
Premise: The sun has risen in the east every morning up until now.
Conclusion: The sun will also rise in the east tomorrow. … Or Not?

Influence of Society

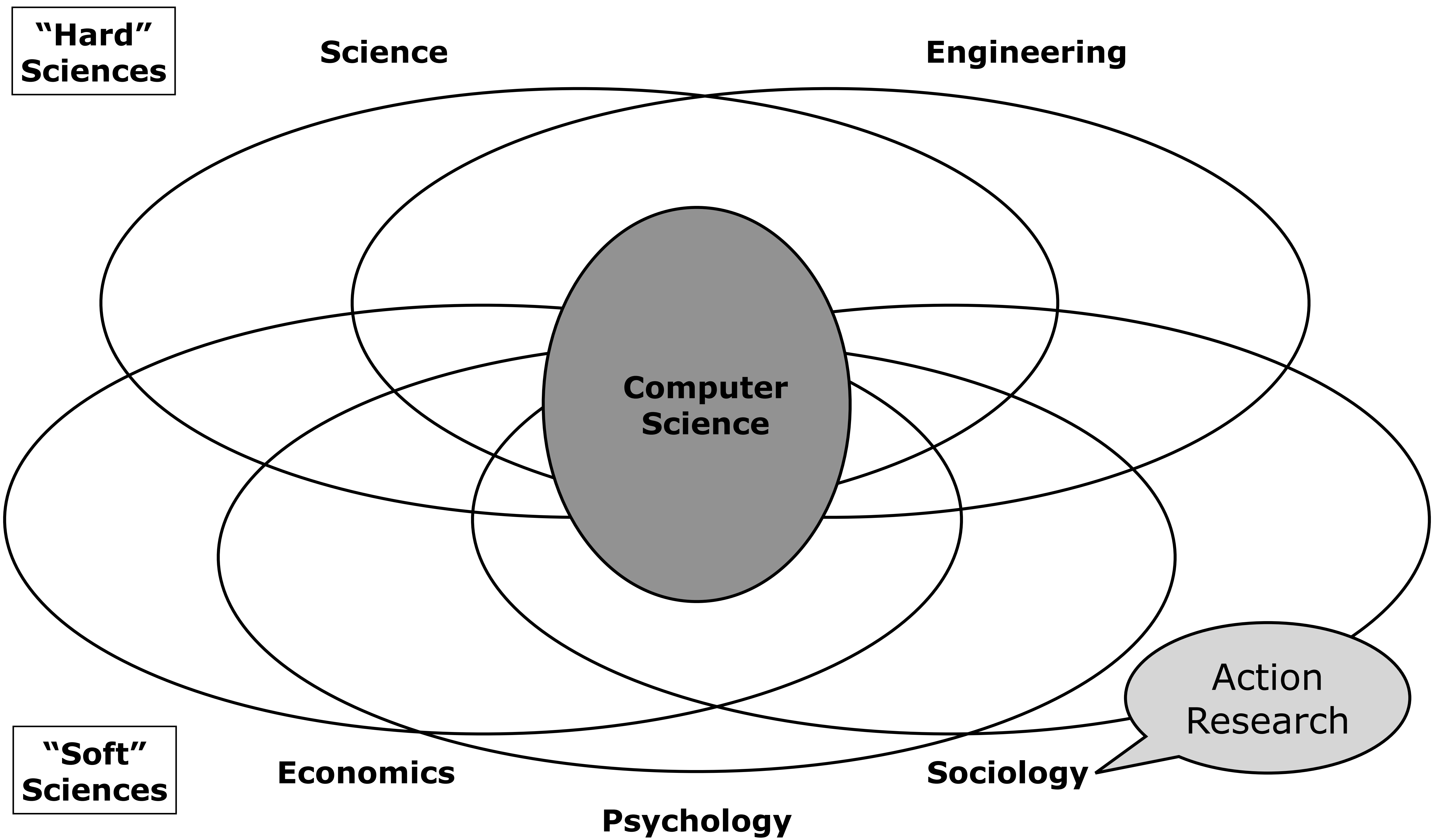
Lives are at stake
(e.g., automatic pilot,
nuclear power plants)

Huge amounts of money
are at stake
(e.g., Ariane V crash,
Denver Airport Baggage)

Software became Ubiquitous
… its not a hobby anymore

Corporate success or failure
is at stake (e.g., telephone
billing, VTM launching 2nd
channel)

Interdisciplinary Nature

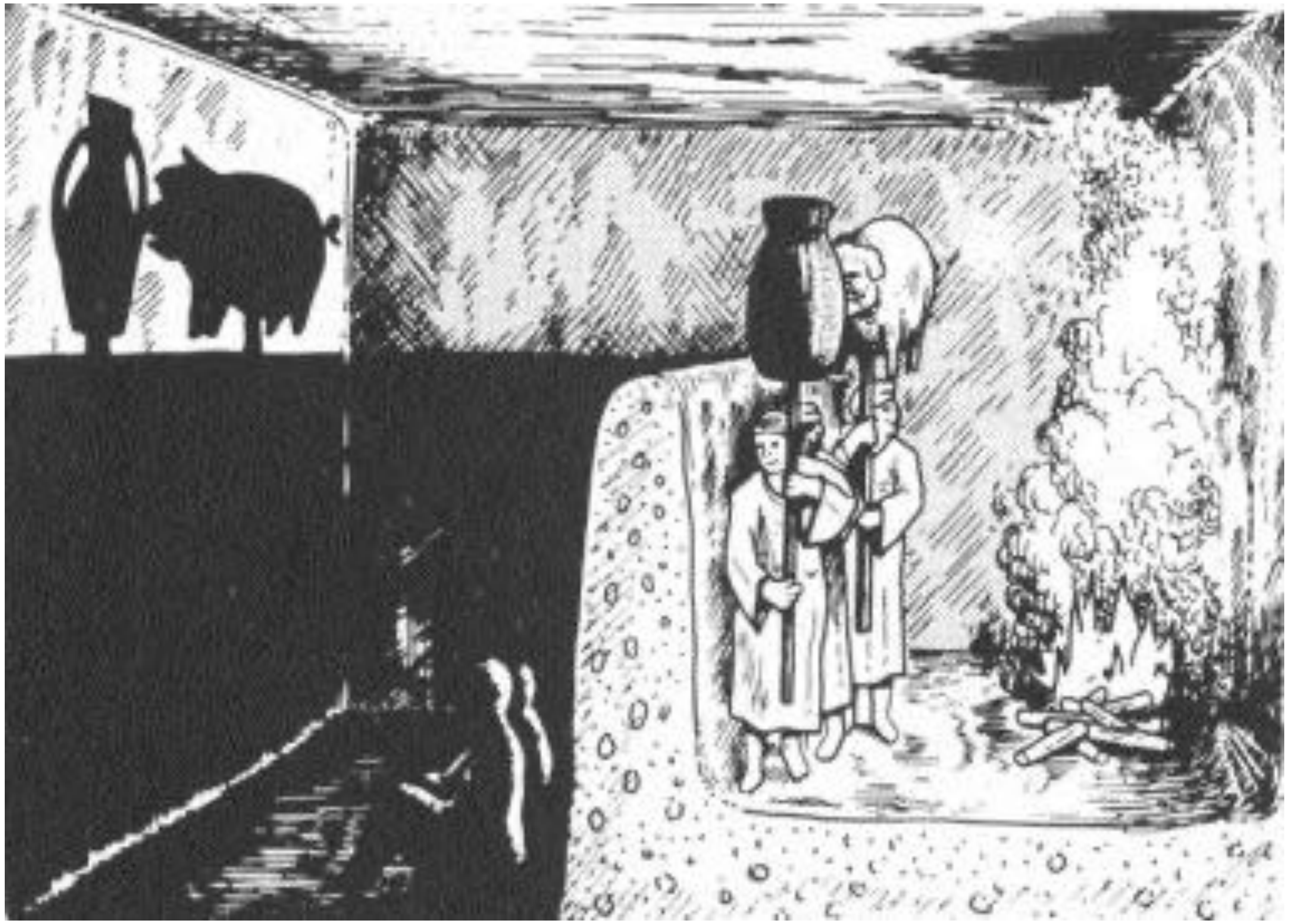


The Oak Forest
Robert Zünd - 1882

Franz and Luciano
Franz Gertsch - 1973

Objective ↔ Subjective

- Plato's cave



- Scientific Paradigm (Kuhn)
  + Dominant paradigm / Competing paradigms / Paradigm shift
    ➡ Normal science vs. Revolutionary science

Dominant view on Research Methods

Physics
("The" Scientific method)

- form hypothesis about a phenomenon
- design experiment
- collect data
- compare data to hypothesis
- accept or reject hypothesis
    + … publish (in Nature)
- get someone else to repeat experiment (replication)

Medicine
(Double-blind treatment)

- form hypothesis about a treatment
- select experimental and control groups that are comparable except for the treatment
- collect data
- commit statistics on the data
- treatment ⇒ difference (statistically significant)

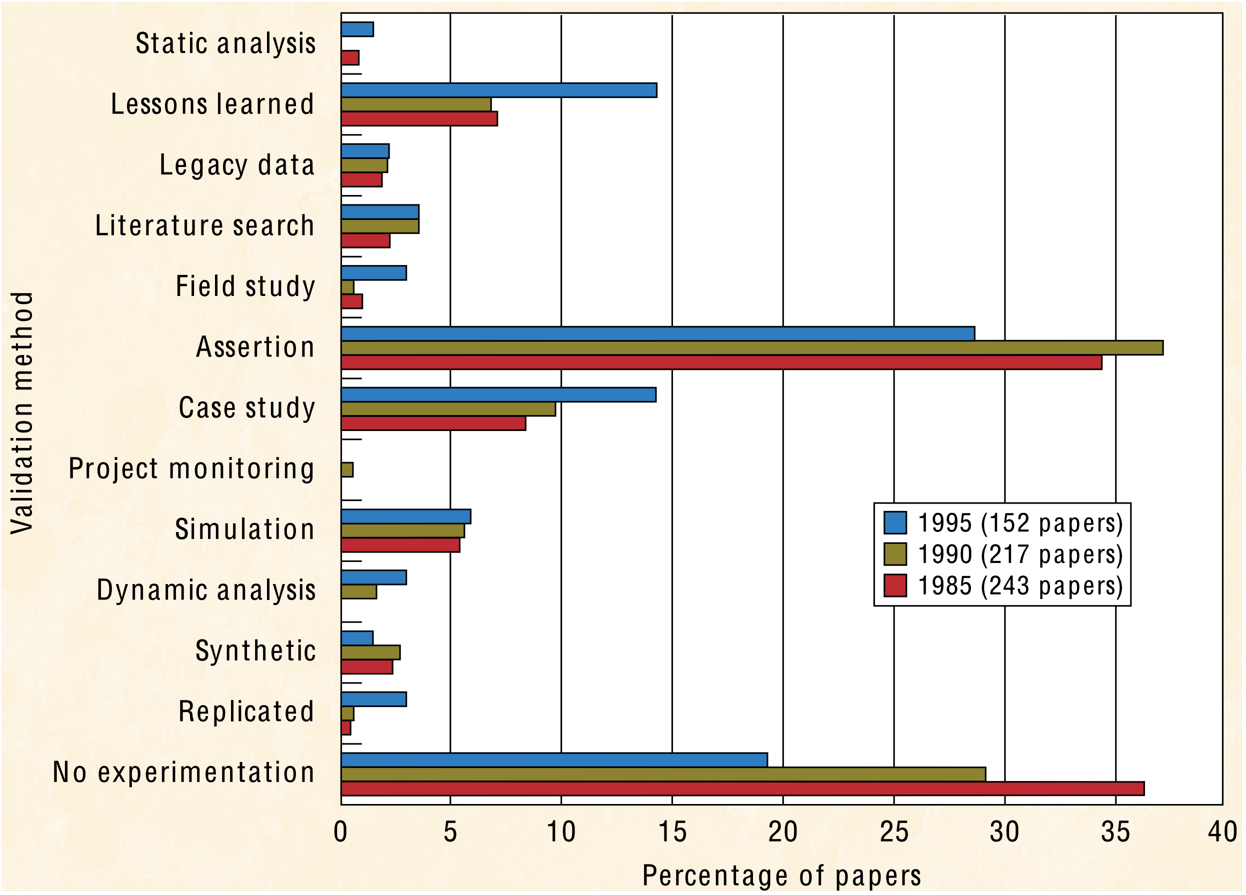
Cannot answer the "big" questions
… in timely fashion

- smoking is unhealthy
- climate change
- darwin theory vs. intelligent design
- …
- agile methods

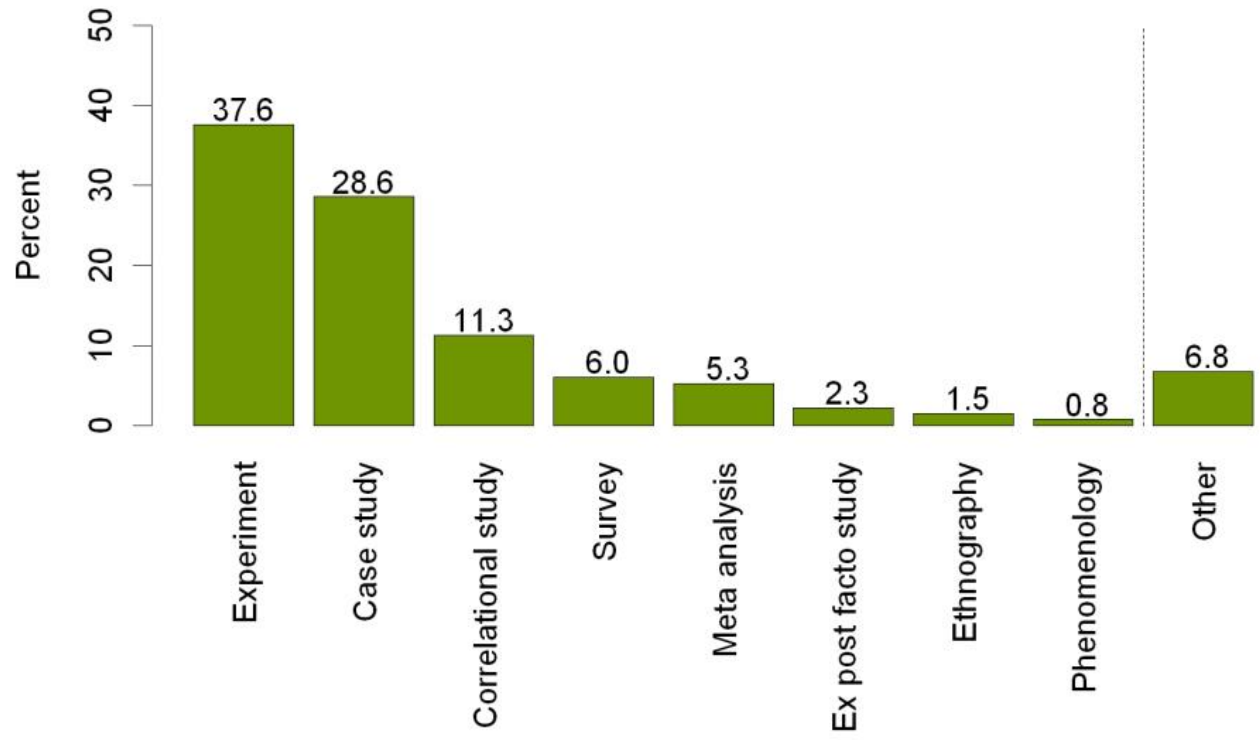
Research Methods in Computer Science

Different Sources

- Marvin V. Zelkowitz and Dolores R. Wallace, "Experimental Models for Validating Technology", IEEE Computer, May 1998.



- Easterbrook, S. M., Singer, J., Storey, M, and Damian, D. Selecting Empirical Methods for Software Engineering Research. Appears in F. Shull and J. Singer (eds) "Guide to Advanced Empirical Software Engineering", Springer, 2007.
- Gordona Dodif-Crnkovic, “Scientific Methods in Computer Science”
- Andreas Höfer, Walter F. Tichy, Status of Empirical Research in Software Engineering, Empirical Software Engineering Issues, p. 10-19, Springer, 2007.



Case studies - Spectrum

case studies are widely used in computer science
⇒ "studying a case" vs. "doing a case study"

7. Simulation
- what if?

6. Formal Model
- underlying concepts?

5. Literature survey
- what is known/unknown?

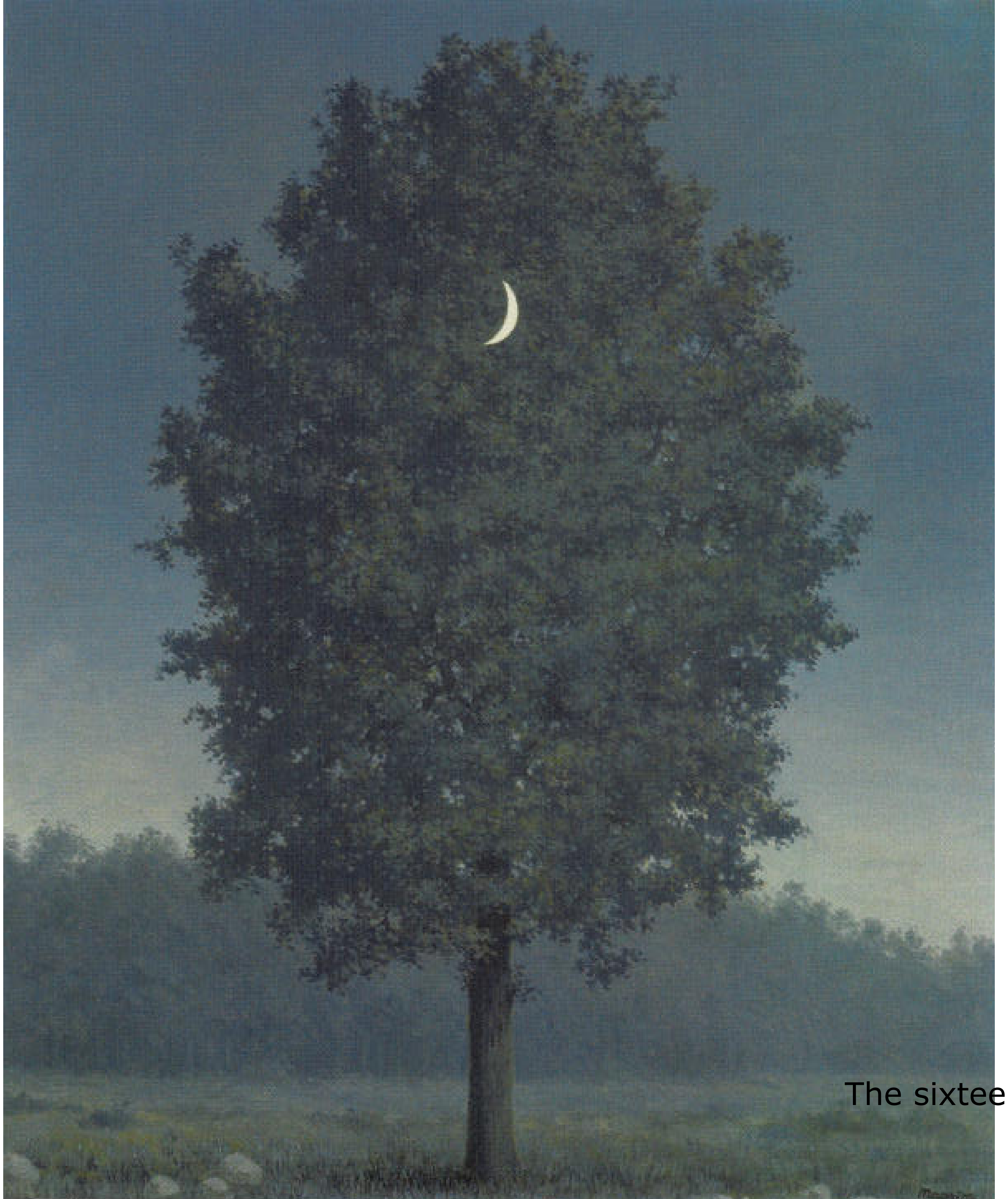
4. Observational Study
- What is "it"?

3. Comparative study
- is it better?

2. Pilot Case, Demonstrator
- is it appropriate?

1. Feasibility study
- is it possible?

Source: Personal experience
(Guidelines for Master Thesis Research – University of Antwerp)



The sixteenth of september
Rene Margritte

Feasibility Study [Proof-of-Concept]

Here is a new idea, is it possible?

➡ Metaphor: Christopher Columbus and western route to India

- Is it *possible* to solve a specific kind of problem … effectively?
  + computer science perspective (P = NP, Turing test, …)
  + engineering perspective (build efficiently; fast — small)
  + economic perspective (cost effective; profitable)
- Is the technique new / novel / innovative?
  + compare against alternatives
    ➡ See literature survey; comparative study

- Proof by construction
  + build a prototype (Proof-of-Concept)
  + often by applying on a "CASE"

- Conclusions
  + primarily qualitative; "lessons learned"
  + quantitative
    - economic perspective: cost - benefit
    - engineering perspective: speed - memory footprint

The Prophet
Pablo Gargallo

Pilot Case (a.k.a. Demonstrator)

Here is an idea that has proven valuable; does it work for us?

➡ Metaphor: Portugal (Amerigo Vespucci) explores western route

- proven valuable
  + accepted merits (e.g. “lessons learned” from feasibility study)
  + there is some (implicit) theory explaining why the idea has merit
- does it work for us
  + context is very important

- Demonstrated on a simple yet representative “CASE”
  + “Pilot case” ≠ “Pilot Study”

- Proof by construction
  + build a prototype
  + apply on a “case”

- Conclusions
  + primarily qualitative; "lessons learned"
  + quantitative; preferably with predefined criteria
    ➡ compare to context before applying the idea!!

Walking man
Standing Figure
– Alberto Giacometti

Comparative Study

Here are two techniques, which one is better?

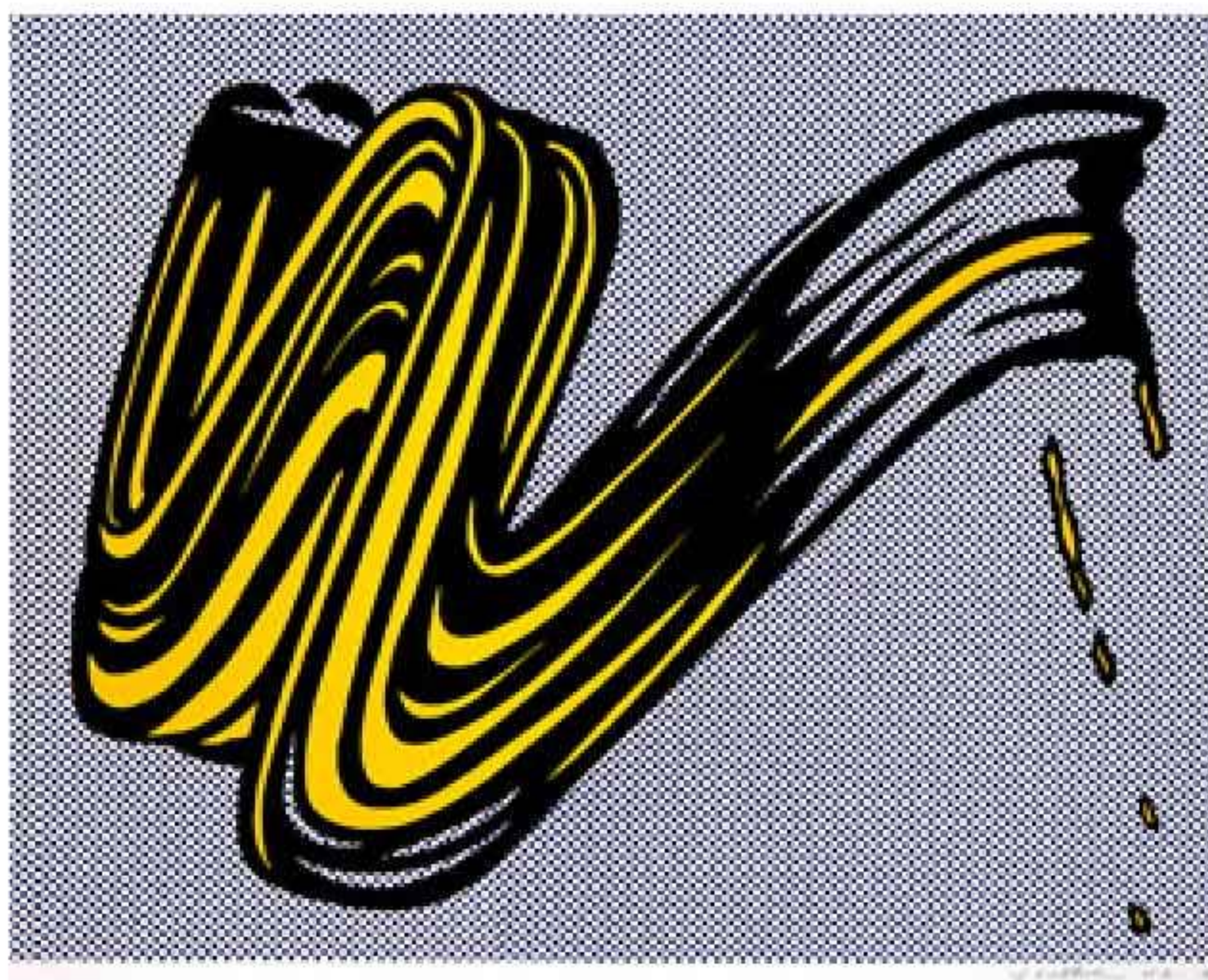
- for a given purpose!
  - \+ (Not necessarily absolute ranking)
- Where are the differences? What are the tradeoffs?

- Criteria check-list
  - \+ predefined
    - should not favor one technique
  - \+ qualitative and quantitative
    - qualitative: how to remain unbiased?
    - quantitative: represent what you want to know?
  - \+ Criteria check-list should be complete and reusable!
    - ➡ If done well, most important *contribution* (replication!)
    - ➡ See literature survey

- Score criteria check-list
  - \+ Often by applying the technique on a "CASE"

- Compare
  - \+ typically in the form of a table


Campbell's
CONDENSED
TOMATO
SOUP



Observational Study [Ethnography]

Understand phenomena through observations

➡ Metaphor: Diane Fossey "Gorillas in the Mist"

- systematic collection of data derived from direct observation of the everyday life
  - + phenomena is best understood in the fullest possible context
    - ➡ observation & participation
    - ➡ interviews & questionnaires
- Observing a series of cases "CASE"
  - + observation vs. participation?
- example: Action Research
  - + Action research is carried out by people who usually recognize a problem or limitation in their workplace situation and, together, devise a plan to counteract the problem, implement the plan, observe what happens, reflect on these outcomes, revise the plan, implement it, reflect, revise and so on.
- Conclusions
  - + primarily qualitative: classifications/observations/…

Torben Giehler
Matterhorn

Paul Klee
Niesen

Literature Survey

What is known? What questions are still open?

- source: B. A. Kitchenham, “Procedures for Performing Systematic Reviews”, Keele University Technical Report EBSE-2007-01, 2007

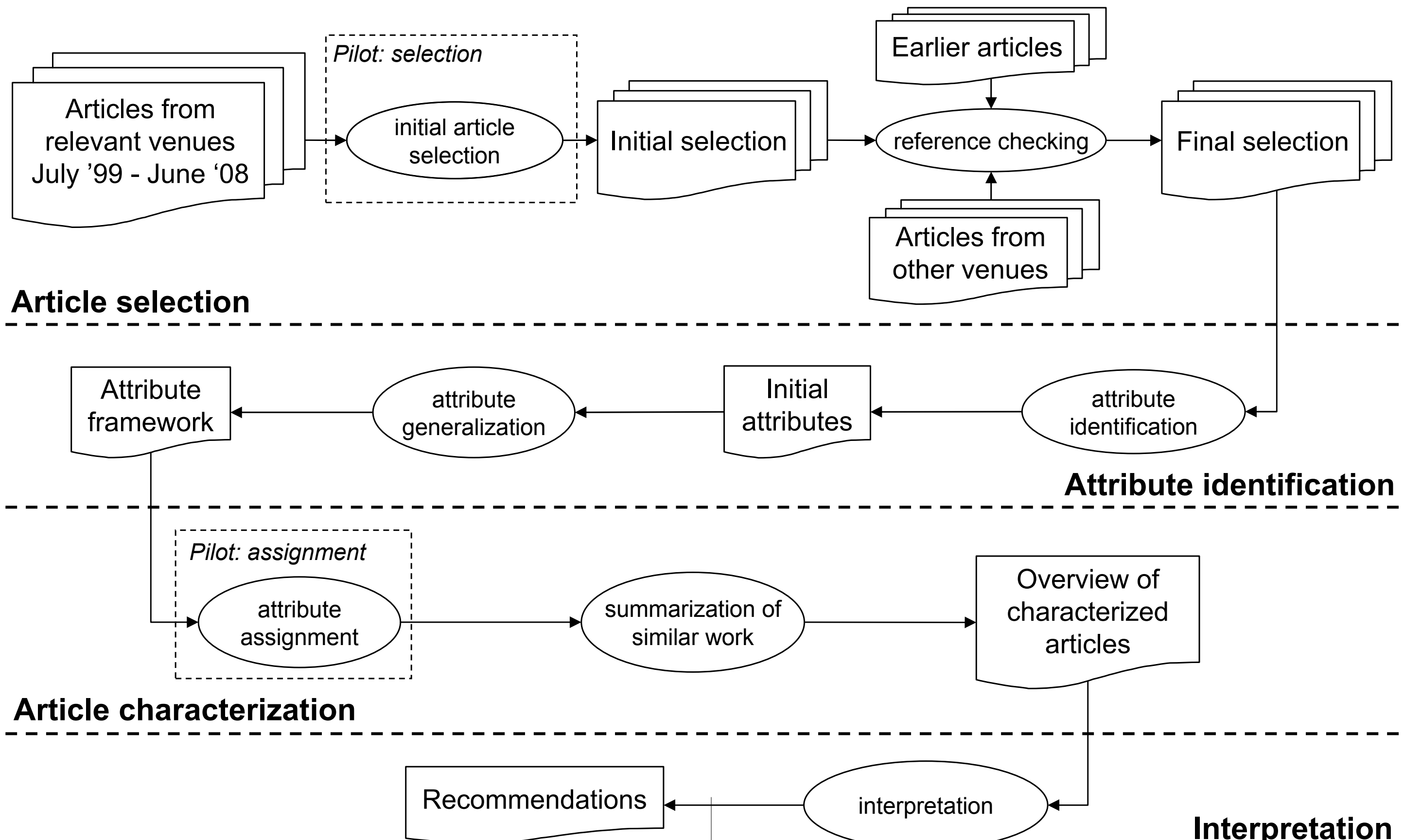
Systematic

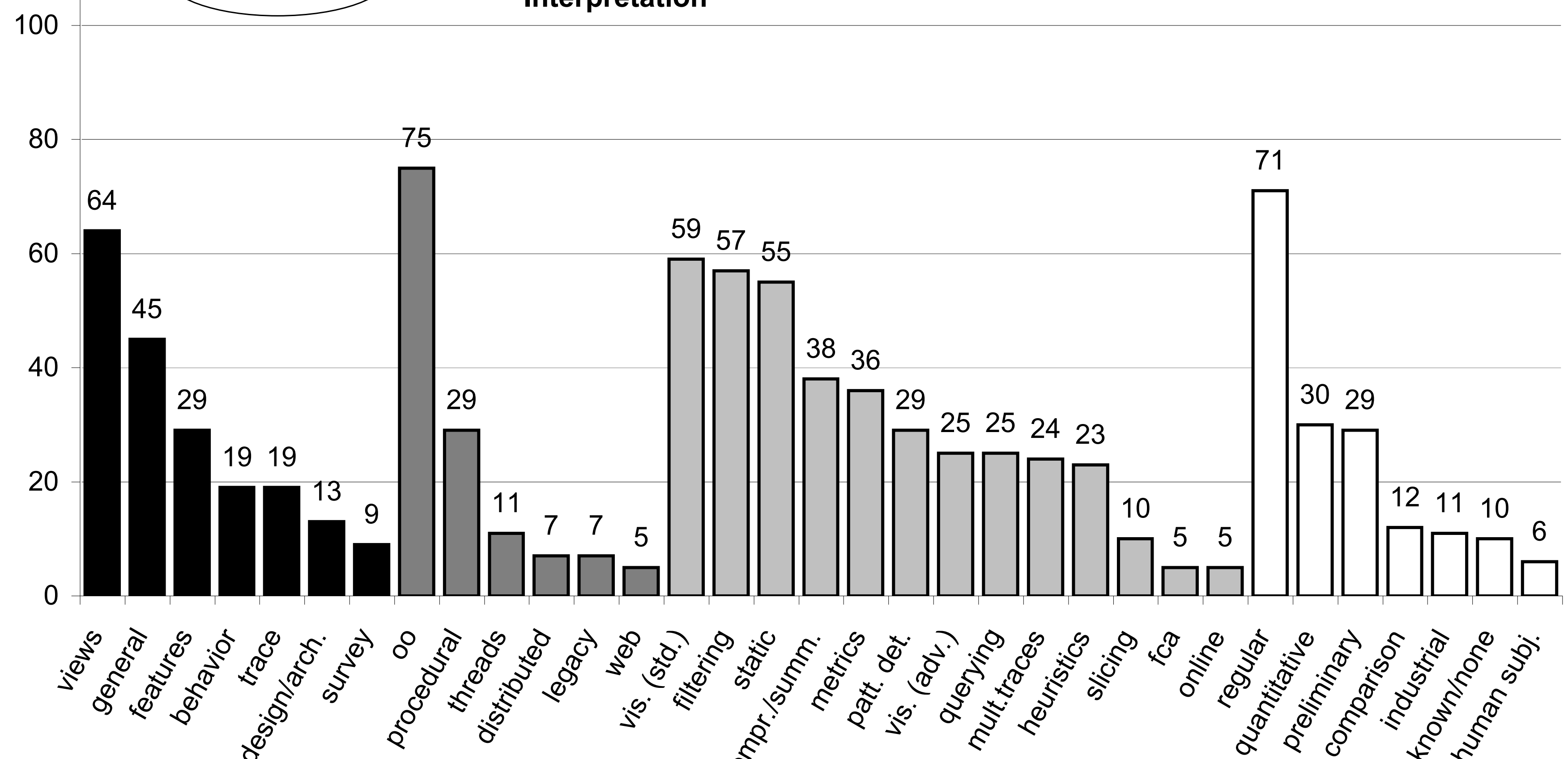
- “comprehensive”
  - ➡ precise research question is prerequisite
  - + defined search strategy (rigor, completeness, replication)
  - + clearly defined scope
    - criteria for inclusion and exclusion
  - + specify information to be obtained
    - the “CASES” are the selected papers

- outcome is organized

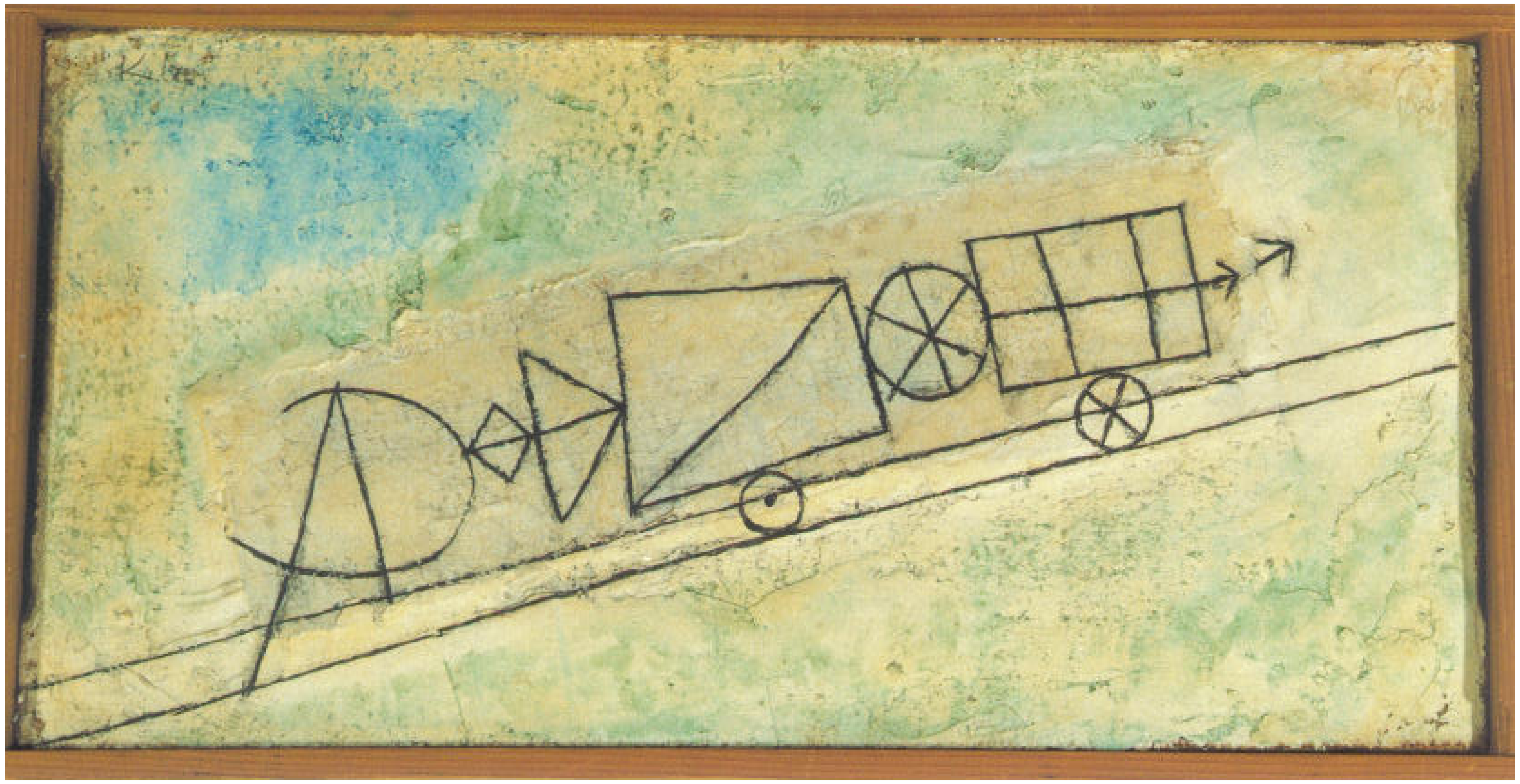
classification	taxonomy	conceptual model
table	tree	frequency

Literature survey - example





Source
Bas Cornelissen, Andy Zaidman, Arie van Deursen, Leon Moonen, Rainer Koschke. A Systematic Survey of Program Comprehension through Dynamic Analysis IEEE Transactions on Software Engineering (TSE): 35(5): 684-702, 2009.



Klee
Bergbahn

Vojin Bakic
Bull

Formal Model

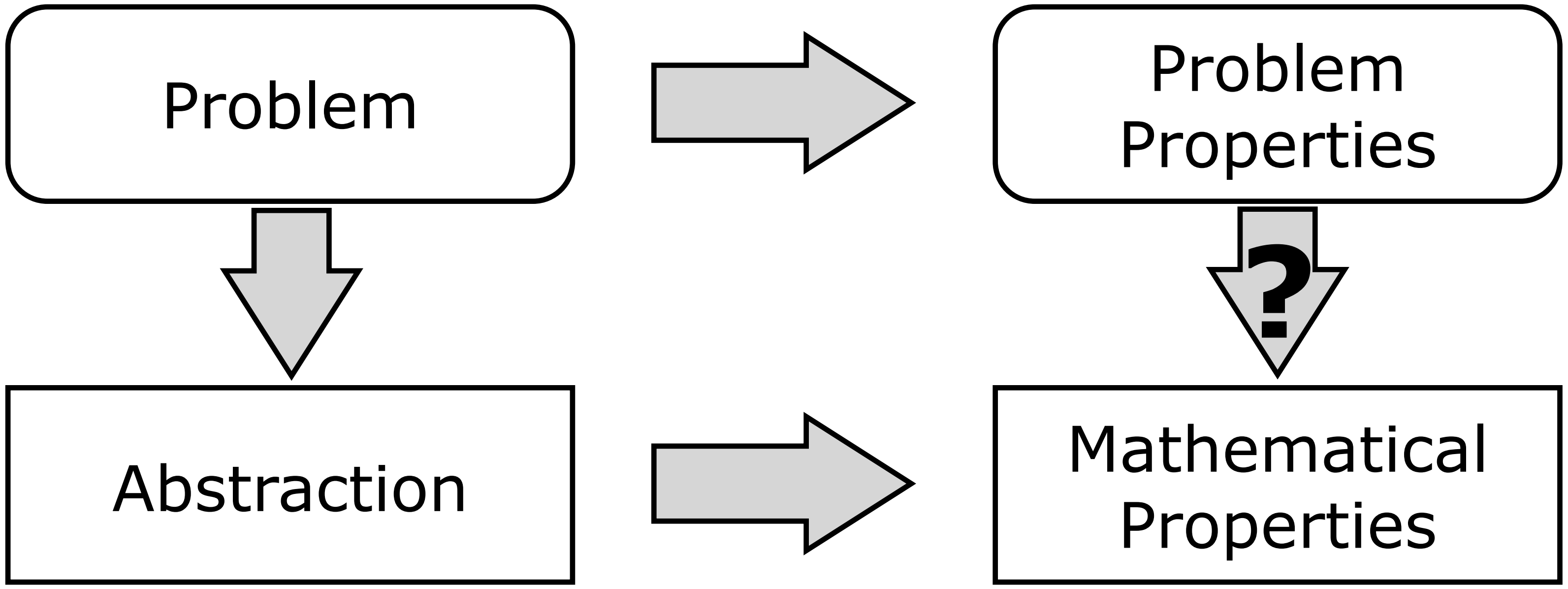
How can we understand/explain the world?

- make a mathematical abstraction of a certain problem
  - + analytical model, stochastic model, logical model, re-write system, ...
  - + often explained using a "CASE"
- prove some important characteristics
  - + based on inductive reasoning, axioms & lemma's, …

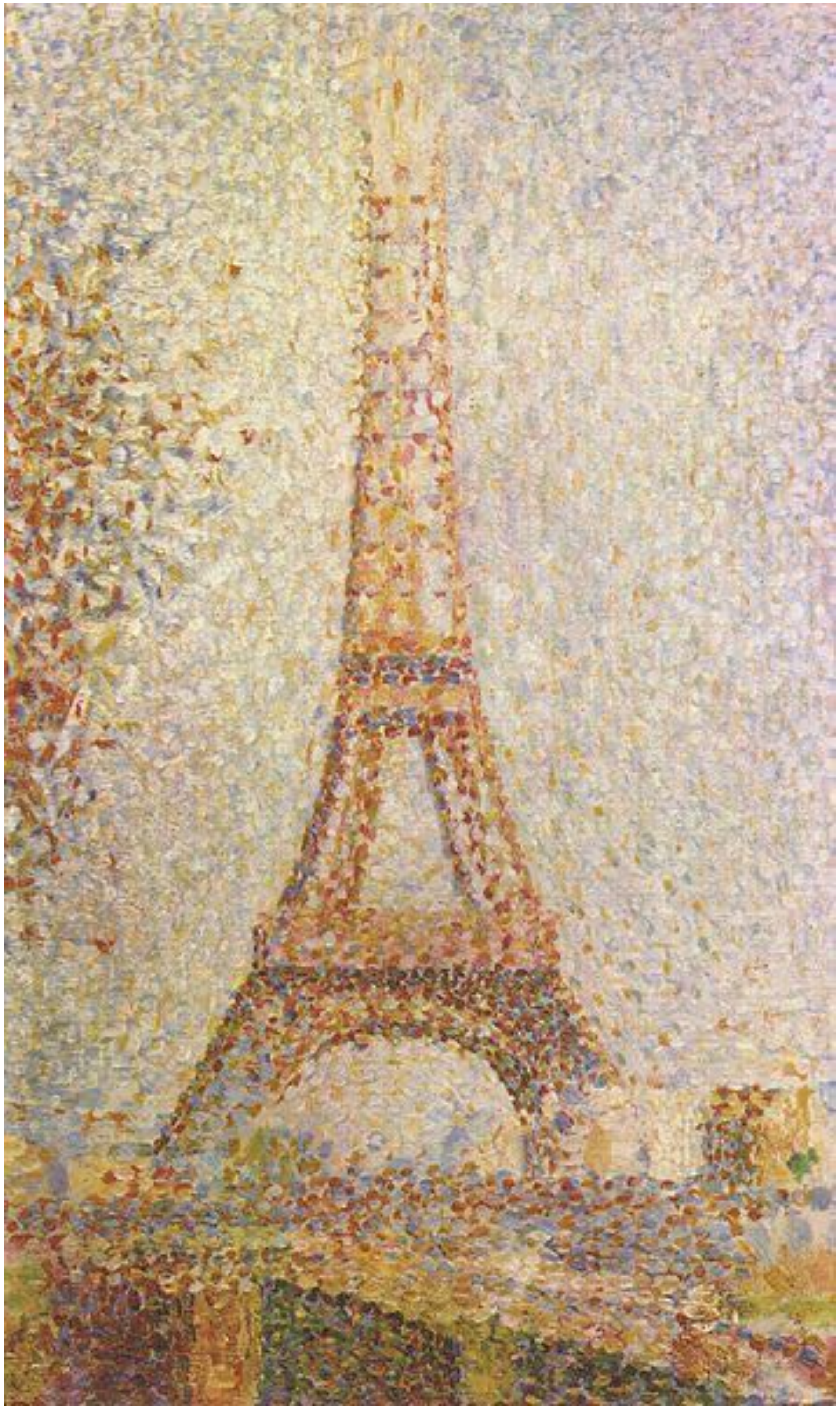
Motivate

- which factors are irrelevant (excluded) and which are not (included)?
- which properties are worthwhile (proven)?

➡ See literature survey



Hodler
Eiger, Mönch and Jungfrau in the Morning Sun



Seurat
Eiffel Tower

Simulation

What would happen if ...?

- study circumstances of phenomena in detail
  + simulated because real world too expensive; too slow or impossible
- make prognoses about what can happen in certain situations
  + test using real observations, typically obtained via a "CASE"

Motivate

- which circumstances are irrelevant (excluded) and which are not (included)?
- which properties are worthwhile (to be observed/predicted)?
  ➡ See literature survey

Examples

- distributed systems (grid); network protocols
  + too expensive or too slow to test in real life
- embedded systems — simulating hardware platforms
  + impossible to observe real clock-speed / memory footprint / ...
  ➡ Heisenberg uncertainty principle

Case studies - Revisited

case studies are widely used in computer science
⇒ "studying a case" vs. "doing a case study"

7. Simulation: test prognoses with real observations obtained via a "CASE"

6. Formal Model
often explained using a "CASE"

5. Literature survey
"CASES" = selected papers

4. Observational Study
Observing a series of "CASES"

3. Comparative study
Score criteria check-list; often by applying on a "CASE"

2. Pilot Case, Demonstrator
Demonstrated on a simple yet representative "CASE"

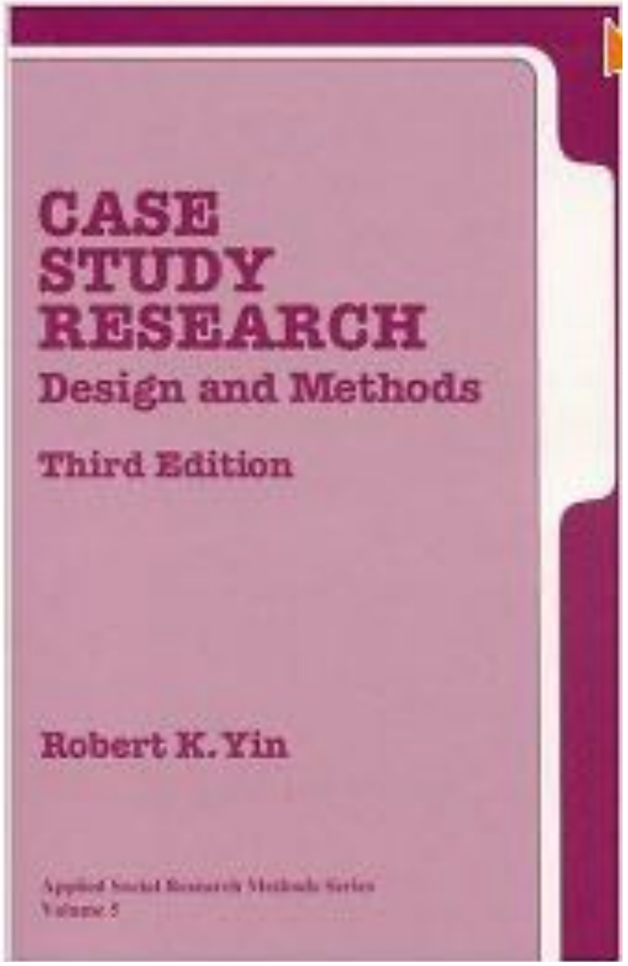
1. Feasibility study
Proof by construction; often by applying on a "CASE"

Case Study Research

Introduction
- Origins of Computer Science
- Research Philosophy

Research Methods
- 1. Feasibility study
- 2. Pilot Case
- 3. Comparative study
- 4. Observational Study [a.k.a. Etnography]
- 5. Literature survey
- 6. Formal Model
- 7. Simulation

Conclusion
- Studying a Case
  vs. Performing a Case Study
  + Proposition
  + Unit of Analysis
  + Threats to Validity



Sources
- Robert K. Yin. Case Study Research: Design and Methods. 3rd Edition. SAGE Publications. California, 2009.
- Bent Flyvbjerg, "Five Misunderstandings About Case Study Research." Qualitative Inquiry, vol. 12, no. 2, April 2006, pp. 219-245.
- Runeson, P. and Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Eng. 14, 2 (Apr. 2009), 131-164.

Spectrum of cases

created for *explanation*
- foo, bar examples
- simple model; illustrates *differences*

Toy-example

accepted teaching vehicle
- "textbook example"
- simple but illustrates *relevant* issues

Exemplar

Martin S. Feather , Stephen Fickas , Anthony Finkelstein , Axel Van Lamsweerde, Requirements and Specification Exemplars, Automated Software Engineering, v.4 n.4, p.419-438, October 1997

Case study

Runeson, P. and Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Eng. 14, 2 (Apr. 2009), 131-164.

real-life example
- industrial system, open-source system
- context is difficult to grasp

Case

Mining Software Repositories Challenge. [Yearly workshop where research tools compete against one another on a common predefined case.]

competition (tool oriented)
- approved by community
- *comparing*

Community case

Susan Elliott Sim, Steve Easterbrook, and Richard C. Holt. Using Benchmarking to Advance Research: A Challenge to Software Engineering, Proceedings of the Twenty-fifth International Conference on Software Engineering, Portland, Oregon, pp. 74-83, 3-10 May, 2003.

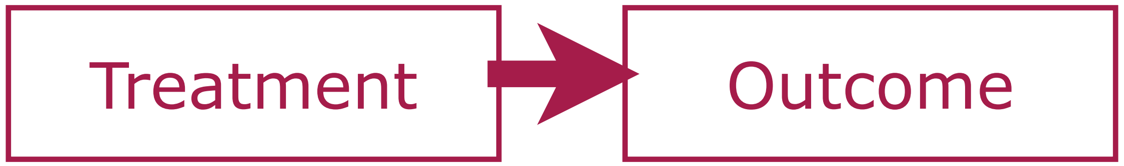
Benchmark

benchmark
- approved by community
- known context
- "*planted*" issues

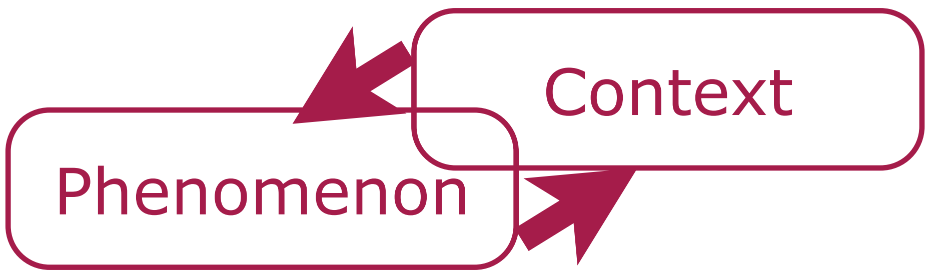
Case study — definition

A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and context are not clearly evident
[Robert K. Yin. Case Study Research: Design and Methods; p. 13]

- empirical inquiry: yes, it is empirical research
- contemporary: (close to) real-time observations
  + incl. interviews
- boundaries between the phenomenon and context not clear
  + as opposed to “experiment”

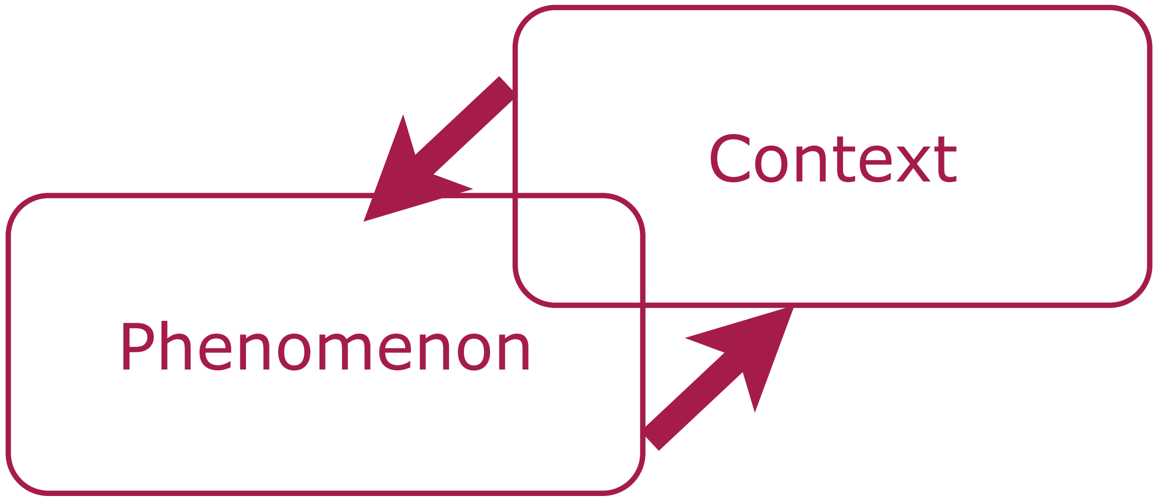


Experiment



Case Study

Case Study — Counter evidence



- many more variables than data points
- multiple sources of evidence; triangulation
- theoretical propositions guide data collection
  (try to confirm or refute propositions with well-selected cases)

Case studies also look for *counter evidence*

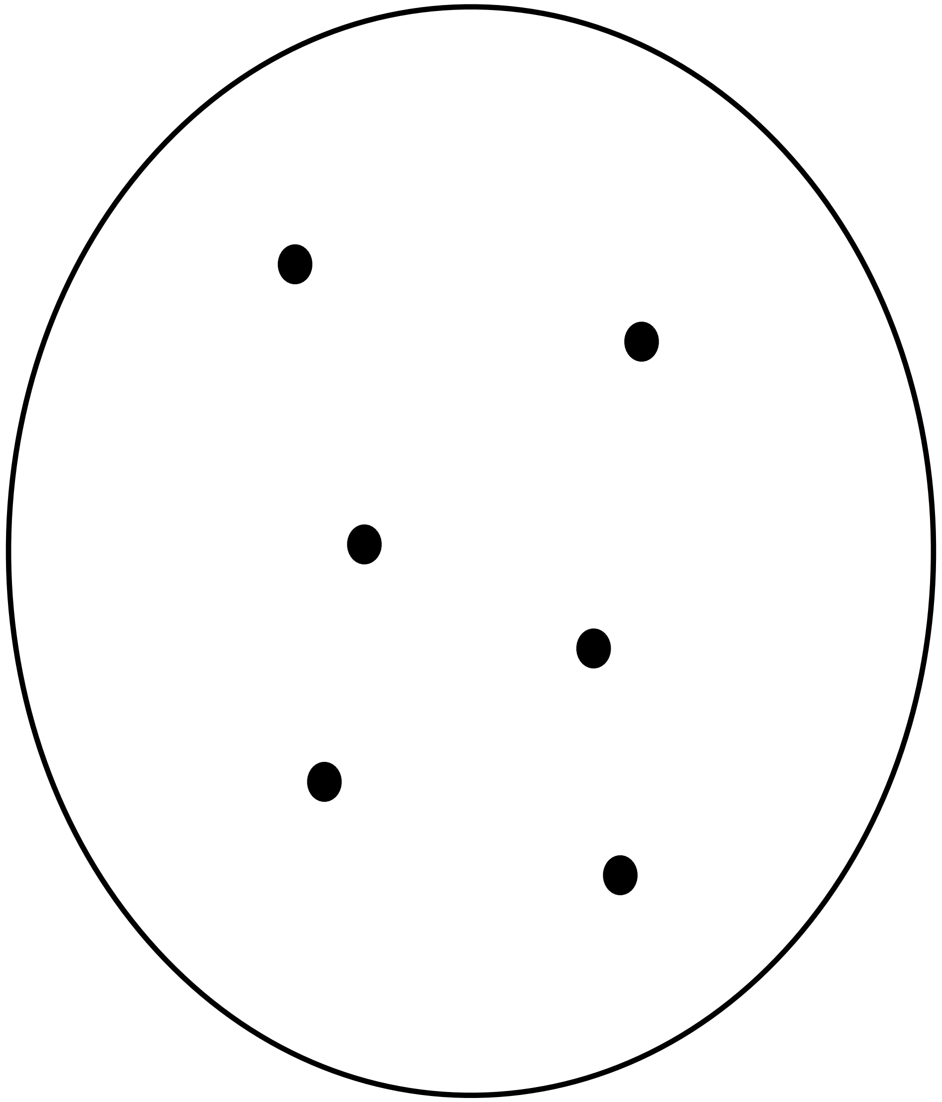
Misunderstanding 2: Generalization

One cannot generalize on the basis of an individual case; therefore the case study cannot contribute to scientific development.

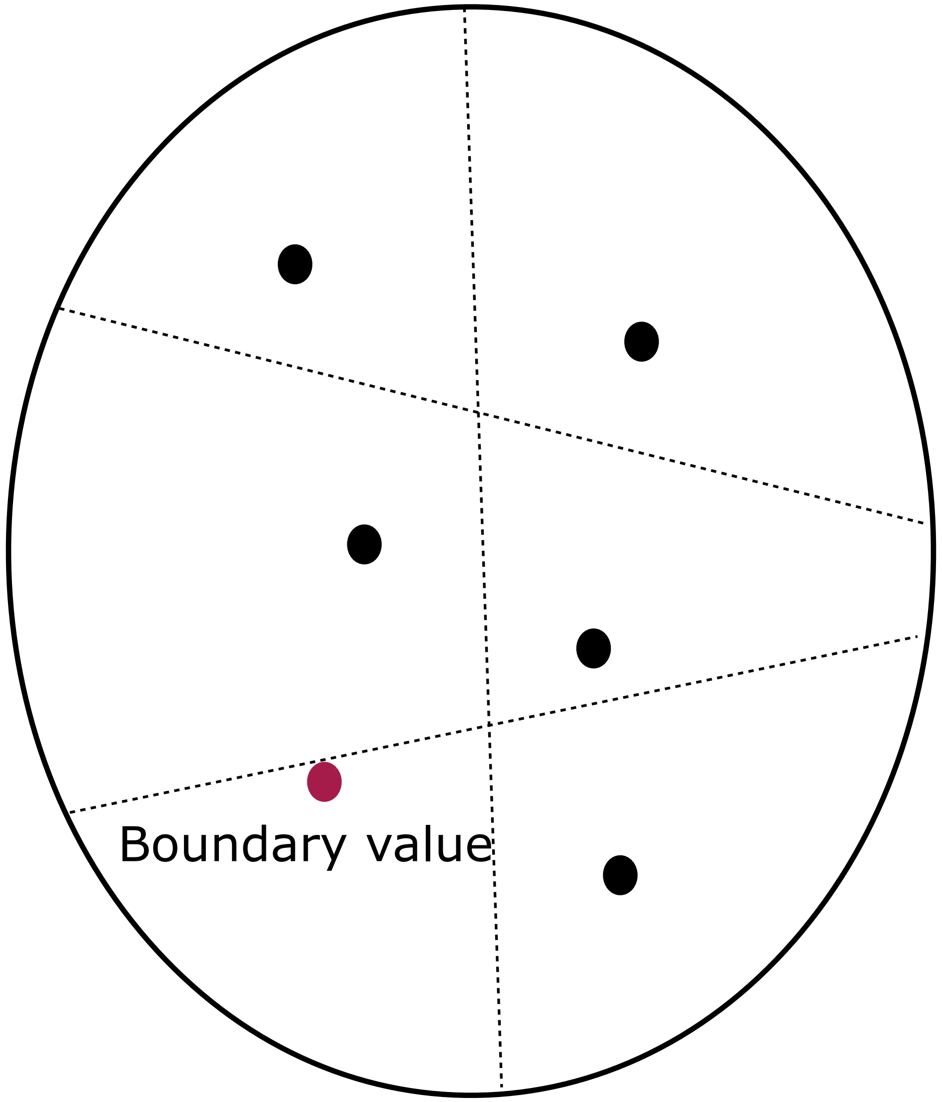
➡ *[Bent Flyvbjerg, "Five Misunderstandings About Case Study Research."]*

- Understanding
  - + The power of examples
  - + Formal generalization is overvalued
    - dominant research views of physics and medicine
- Counterexamples
  - + one black swan falsifies “all swans are white”
    - case studies generate deep understanding; what appears to be white often turns out to be black
- sampling logic vs. replication logic
  - + sampling logic: operational enumeration of entire universe
    - use statistics: generalize from “randomly selected” observations
  - + replication logic: careful selection of boundary values
    - use logic reasoning: presence of absence of property has effect

Sampling Logic vs. Replication Logic



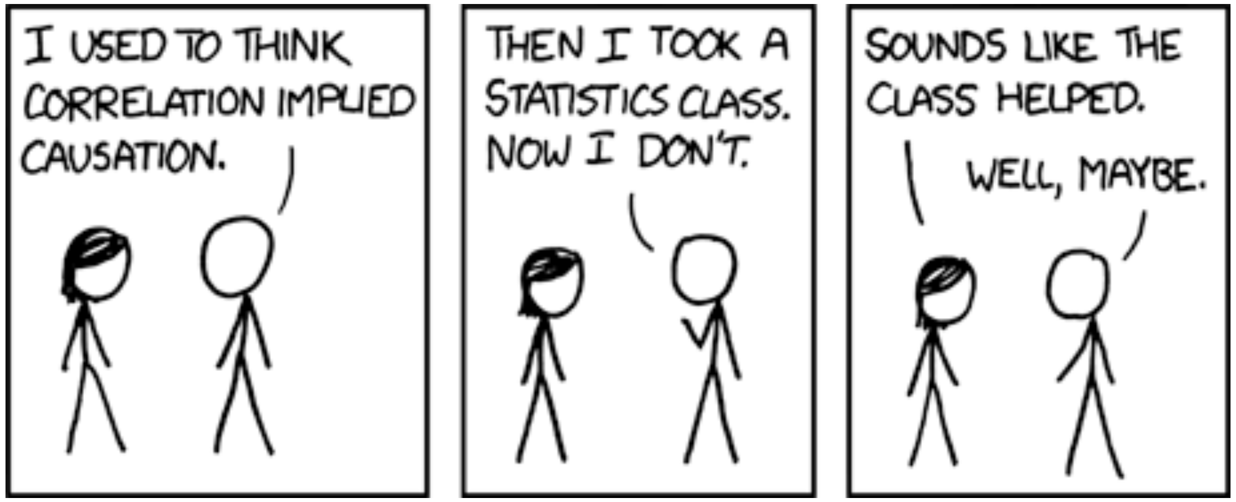
Random selection
⇒ generalize for entire population



Selection of (boundary) value
⇒ understand differences

- propositions
- units of analysis

Correlation Does Not Imply Causation



Research questions for Case Studies

Existence: ***Exploratory***
- Does X exist?

Description & Classification
- What is X like?
- What are its properties?
- How can it be categorized?
- How can we measure it?
- What are its components?

Descriptive-Comparative
- How does X differ from Y?

Frequency and Distribution
- How often does X occur?
- What is an average amount of X?

Descriptive-Process
- How does X normally work?
- By what process does X happen?
- What are the steps as X evolves?

Relationship ***Explanatory***
- Are X and Y related?
- Do occurrences of X correlate with occurrences of Y?

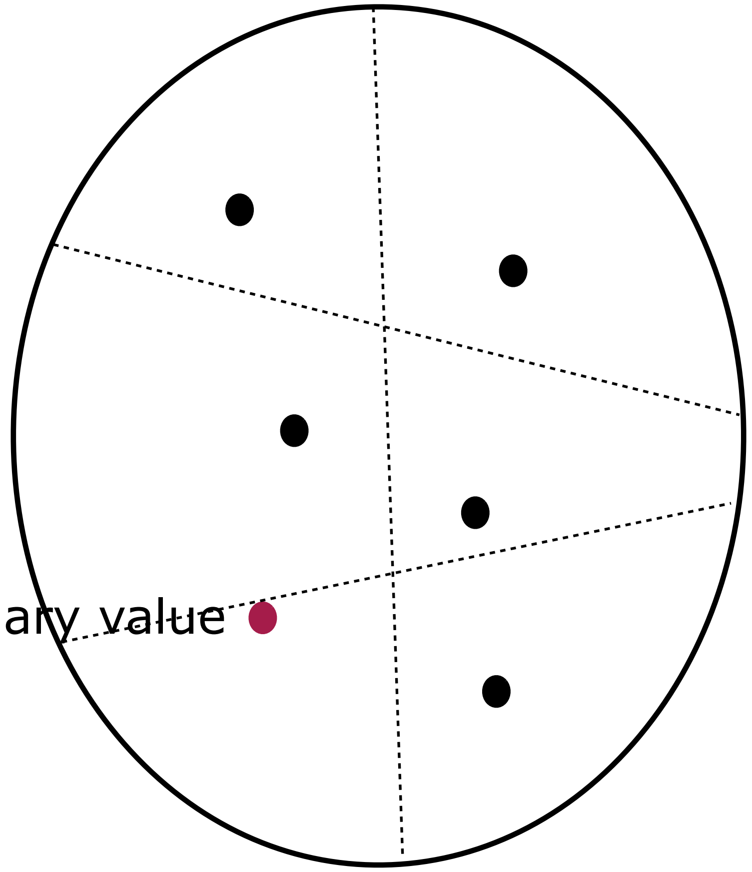
Causality
- What causes X?
- What effect does X have on Y?
- Does X cause Y?
- Does X prevent Y?

Causality-Comparative
- Does X cause more Y than does Z?
- Is X better at preventing Y than is Z?
- Does X cause more Y than does Z under one condition but not others?

Design
- What is an effective way to achieve X?
- How can we improve X?

Source: Empirical Research Methods in Requirements Engineering.
Tutorial given at RE'07, New Delhi, India, Oct 2007.

Proposition (a.k.a. Purpose)



Where to expect boundaries?
⇒ Thorough preparation is necessary!

⇒ You need an explicit *theory*.

Exploratory	Confirmatory
Exploratory case studies are used as initial investigations of some phenomena to derive new hypotheses and build theories.(*)	*Confirmatory* case studies are used to test existing theories. The latter are especially important for refuting theories: a detailed case study of a real situation in which a theory fails may be more convincing than failed experiments in the lab.(*)

(*) Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for soft- ware engineering research. In Forrest Shull, Janice Singer, and Dag I. K. Sjoberg, editors, Guide to Advanced Empirical Software Engineering, pages 285—311. Springer London, 2008.

Units of Analysis

What phenomena to analyze

- depends on research questions
- affects data collection & interpretation
- affects generalizability

Possibilities

- individual developer
- a team
- a decision
- a process
- a programming language
- a tool

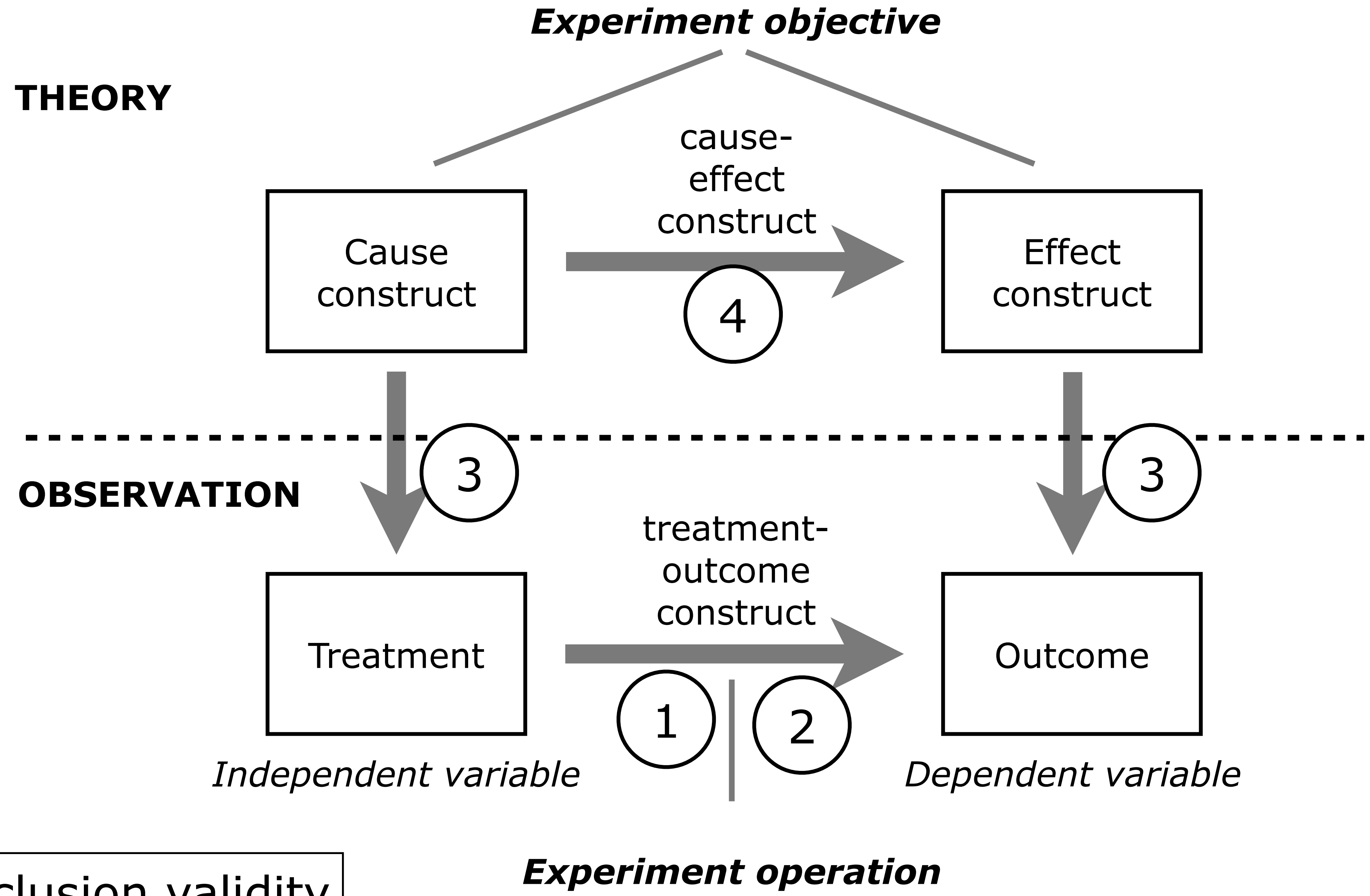
Design in advance

- avoid "easy" units of analysis
  + cases restricted to Java because parser
    - Is the language really an issue for your research question?
  + report size of the system (KLOC, # Classes, # Bug reports)
    - Is team composition not more important?

Example: Clone Detection, Bug Prediction

- the tool/algorithm
  Does it work?
- the individual developer
  How/why does he produce bugs/clones?
- about the culture/process in the team
  How does the team prevent bugs/clones?
  How successful is this prevention?
- about the programming language
  How vulnerable is the programming language towards clones / bugs?
  (COBOL vs. AspectJ)

Threats to Validity (Experiments)



1. Conclusion validity
2. Internal validity
3. Construct validity
4. External validity

Threats to validity (Case Studies)

- Source: Runeson, P. and Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering.

1. Construct validity
- Do the operational measures reflect what the researcher had in mind?

2. Internal validity
- Are there any other factors that may affect the results?
  ➡ Mainly when investigating causality!

3. External validity
- To what extent can the findings be generalized?
  ➡ Precise research question & units of analysis required

4. Reliability
- To what extent is the data and the analysis dependent on the researcher (the instruments, …)

Other categories have been proposed as well
- credibility, transferability, dependability, confirmability

Threats to validity — Examples (1/2)

1. Construct validity

- *Do the operational measures reflect what the researcher had in mind?*
- Time recorded vs. time spent
- Execution time, memory consumption, …
  + noise of operating system, sampling method
- Human-assigned classifiers (bug severity, …)
  + risk for “default” values
- Participants in interviews have pressure to answer positively

2. Internal validity

- *Are there any other factors that may affect the results?*
- Were phenomena observed under special conditions
  + in the lab, close to a deadline, company risked bankruptcy, …
  + major turnover in team, contributors changed (open-source), …
- Similar observations repeated over time (learning effects)

Threats to validity — Examples (2/2)

3. External validity

- *To what extent can the findings be generalized?*
- Does it apply to other languages? other sizes? other domains?
- Background & education of participants
- Simplicity & scale of the team
    + small teams & flexible roles vs. large organizations & fixed roles

4. Reliability

- *To what extent is the data and the analysis dependent on the researcher (the instruments, …)*
- How did you cope with bugs in the tool, the instrument?
- Classification: if others were to classify, would they obtain the same?
- How did you search for evidence in mailing archives, bug reports, …

Reliability: Instrument Validity

JOURNAL ARTICLE

A Scientist's Nightmare: Software Problem Leads to Five Retractions

Greg Miller

Science
New Series, Vol. 314, No. 5807 (Dec. 22, 2006), pp. 1856-1857 (2 pages)
Published By: American Association for the Advancement of Science

Replication

Replicating MSR:
A study of the potential replicability of papers published in the Mining Software Repositories Proceedings

Gregorio Robles
GSyC/LibreSoft
Universidad Rey Juan Carlos
Madrid, Spain
Email: grex@gsyc.urjc.es

Results show that MSR authors use in general publicly available data sources, mainly from free software repositories, but that the amount of publicly available processed datasets is very low.

Source: 7th IEEE Working Conference on Mining Software Repositories (MSR 2010), 2010, pp. 171-180, doi: 10.1109/MSR.2010.5463348.

Data Management Plan

TEMPLATE HORIZON 2020 DATA MANAGEMENT PLAN (DMP)

- **Instructions and footnotes in blue must not appear in the text.**
- **For options [in square brackets]: the option that applies must be chosen.**
- **For fields in [grey in square brackets] (even if they are part of an option as specified in the previous item): enter the appropriate data.**

Introduction

This Horizon 2020 DMP template has been designed to be applicable to any Horizon 2020 project that produces, collects or processes research data. You should develop a single DMP for your project to cover its overall approach. However, where there are specific issues for individual datasets (e.g. regarding openness), you should clearly spell this out.

Guidelines on FAIR Data Management in Horizon 2020 are available in the Online Manual.

FAIR data management

In general terms, your research data should be 'FAIR', that is findable, accessible, interoperable and re-usable. These principles precede implementation choices and do not necessarily suggest any specific technology, standard, or implementation-solution.

This template is not intended as a strict technical implementation of the FAIR principles, it is rather inspired by FAIR as a general concept.

More information about FAIR:

FAIR data principles (FORCE11 discussion forum)

FAIR principles (article in Nature)

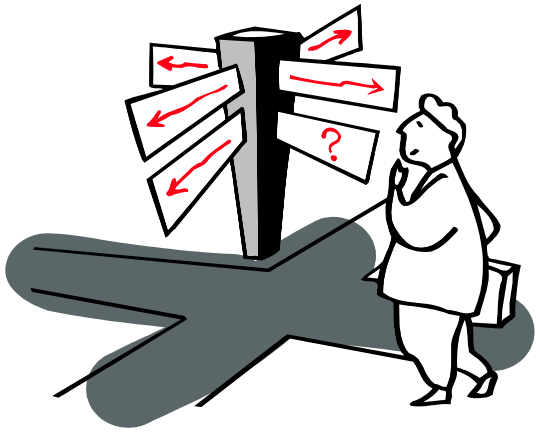
Threats to validity = Risk Management

No experimental design can be "perfect"
… but you can limit the chance of deriving false conclusions

- manage the risk of false conclusions as much as possible
  - + likelihood
  - + impact

- state clearly what and how you alleviated the risk (replication!)
  - + construct validity
    - precise metric definitions
    - GQM paradigm
  - + internal & external validity
    - report the context consciously
  - + Reliability
    - bugs in tools: testing, usage of well-known libraries, …
    - classification: develop guidelines & others repeat classification
    - search for evidence (mailing archives, bug reports, …): have an explicit search procedure

1. Research Methods



Introduction

- Origins of Computer Science
- Research Philosophy

Research Methods

- 1. Feasibility study
- 2. Pilot Case
- 3. Comparative study
- 4. Observational Study [a.k.a. Etnography]
- 5. Literature survey
- 6. Formal Model
- 7. Simulation

Conclusion

- Studying a Case
  vs. Performing a Case Study
  + Proposition
  + Unit of Analysis
  + Threats to Validity

Studying a case vs. Performing a case study

1. Questions

- most likely "How" and "Why"; also sometimes "What"

2. Propositions (a.k.a. Purpose)

- explanatory: where to look for evidence
- exploratory: rationale and direction
  - + example: Christopher Columbus asks for sponsorship
    - Why three ships (not one, not five)?
    - Why going westward (not south?)
- role of "Theories"
  - + possible explanations (how, why) for certain phenomena
    - ➡ Obtained through literature survey

3. Unit(s) of analysis

- What is the case?

4. Logic linking data to propositions
+ 5. Criteria for interpreting findings

- Chain of evidence from multiple sources
- When does data confirm proposition? When does it refute?

Threats to validity

----------Low hanging fruit----------

Information and Software Technology
Volume 133, May 2021, 106514

Case Study Research in Software Engineering—It is a Case, and it is a Study, but is it a Case Study?

Claes Wohlin

Show more

Share Cite

https://doi.org/10.1016/j.infsof.2021.106514 Get rights and content