Empirical Software Engineering Research

Controlled Experiments

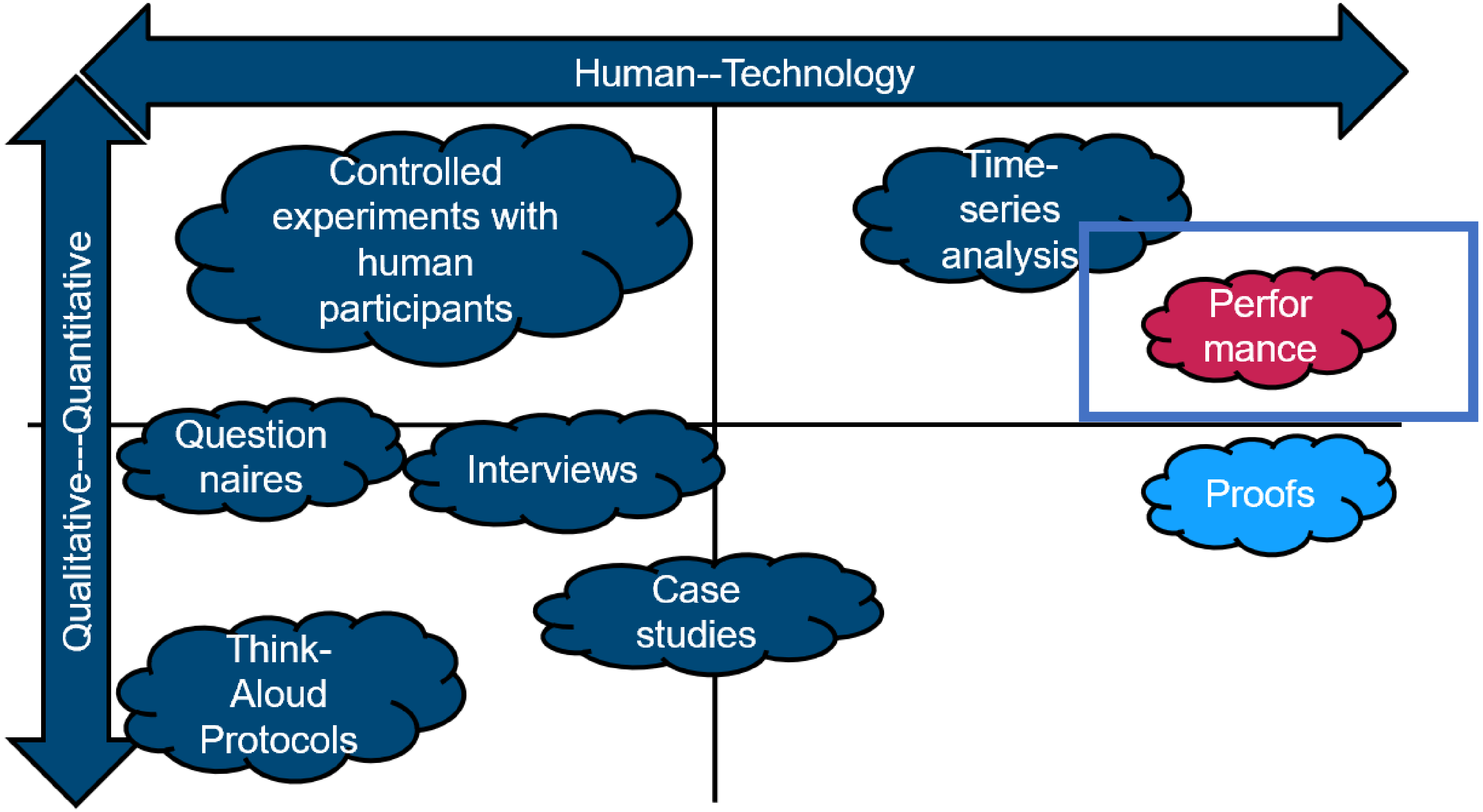
Norman Peitek, Annabelle Bergum, Lina Lampel, Sven Apel

Controlled Experiments: Performance

Learning Goals

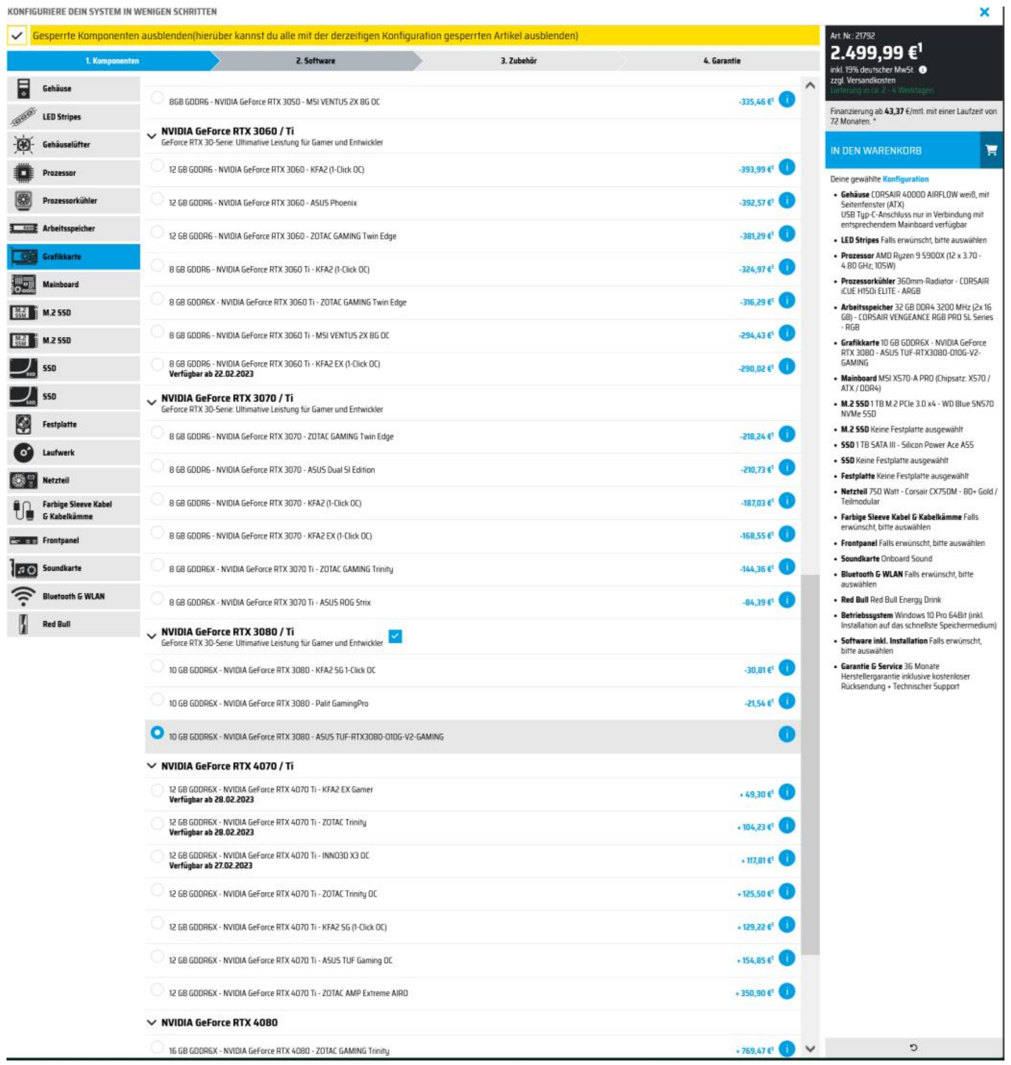
- Understand difficulty of performance analysis
- Evaluate performance analyses

Context


Human--Technology
Qualitative---Quantitative
Controlled experiments with human participants
Time-series analysis
Perfor mance
Question naires
Interviews
Proofs
Case studies
Think-Aloud Protocols

Motivation: Examples

SE UNIVERSITÄT DES SAARLANDES



KONFIGURIERE DEIN SYSTEM IN WENIGEN SCHRITTEN

✓ Gesperrte Komponenten ausblenden(hierüber kannst du alle mit der derzeitigen Konfiguration gesperrten Artikel ausblenden)

1. Komponenten | 2. Software | 3. Zubehör | 4. Garantie

- Gehäuse
- LED Stripes
- Gehäuselüfter
- Prozessor
- Prozessorkühler
- Arbeitsspeicher
- Grafikkarte
- Mainboard
- M.2 SSD
- M.2 SSD
- SSD
- SSD
- Festplatte
- Laufwerk
- Netzteil
- Farbige Sleeve Kabel & Kabelkämme
- Frontpanel
- Soundkarte
- Bluetooth & WLAN
- Red Bull

8GB GDDR6 - NVIDIA GeForce RTX 3050 - MSI VENTUS 2X 8G OC	-335,46 €¹
NVIDIA GeForce RTX 3060 / Ti GeForce RTX 30-Serie: Ultimative Leistung für Gamer und Entwickler	
12 GB GDDR6 - NVIDIA GeForce RTX 3060 - KFA2 (1-Click OC)	-393,99 €¹
12 GB GDDR6 - NVIDIA GeForce RTX 3060 - ASUS Phoenix	-392,57 €¹
12 GB GDDR6 - NVIDIA GeForce RTX 3060 - ZOTAC GAMING Twin Edge	-381,29 €¹
8 GB GDDR6 - NVIDIA GeForce RTX 3060 Ti - KFA2 (1-Click OC)	-324,97 €¹
8 GB GDDR6X - NVIDIA GeForce RTX 3060 Ti - ZOTAC GAMING Twin Edge	-316,29 €¹
8 GB GDDR6 - NVIDIA GeForce RTX 3060 Ti - MSI VENTUS 2X 8G OC	-294,43 €¹
8 GB GDDR6 - NVIDIA GeForce RTX 3060 Ti - KFA2 EX (1-Click OC) **Verfügbar ab 22.02.2023**	-290,02 €¹
NVIDIA GeForce RTX 3070 / Ti GeForce RTX 30-Serie: Ultimative Leistung für Gamer und Entwickler	
8 GB GDDR6 - NVIDIA GeForce RTX 3070 - ZOTAC GAMING Twin Edge	-218,24 €¹
8 GB GDDR6 - NVIDIA GeForce RTX 3070 - ASUS Dual SI Edition	-210,73 €¹
8 GB GDDR6 - NVIDIA GeForce RTX 3070 - KFA2 (1-Click OC)	-187,03 €¹
8 GB GDDR6 - NVIDIA GeForce RTX 3070 - KFA2 EX (1-Click OC)	-168,55 €¹
8 GB GDDR6X - NVIDIA GeForce RTX 3070 Ti - ZOTAC GAMING Trinity	-144,36 €¹
8 GB GDDR6X - NVIDIA GeForce RTX 3070 Ti - ASUS ROG Strix	-84,39 €¹
NVIDIA GeForce RTX 3080 / Ti ✓ GeForce RTX 30-Serie: Ultimative Leistung für Gamer und Entwickler	
10 GB GDDR6X - NVIDIA GeForce RTX 3080 - KFA2 SG 1-Click OC	-30,81 €¹
10 GB GDDR6X - NVIDIA GeForce RTX 3080 - Palit GamingPro	-21,54 €¹
[✓ 10 GB GDDR6X - NVIDIA GeForce RTX 3080 - ASUS TUF-RTX3080-O10G-V2-GAMING]	
NVIDIA GeForce RTX 4070 / Ti	
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - KFA2 EX Gamer **Verfügbar ab 28.02.2023**	+ 49,30 €¹
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - ZOTAC Trinity **Verfügbar ab 28.02.2023**	+ 104,23 €¹
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - INNO3D X3 OC **Verfügbar ab 27.02.2023**	+ 117,81 €¹
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - ZOTAC Trinity OC	+ 125,50 €¹
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - KFA2 SG (1-Click OC)	+ 129,22 €¹
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - ASUS TUF Gaming OC	+ 154,85 €¹
12 GB GDDR6X - NVIDIA GeForce RTX 4070 Ti - ZOTAC AMP Extreme AIRO	+ 350,90 €¹
NVIDIA GeForce RTX 4080	
16 GB GDDR6X - NVIDIA GeForce RTX 4080 - ZOTAC GAMING Trinity	+ 769,47 €¹

Art. Nr.: 21792

2.499,99 €¹

inkl. 19% deutscher MwSt.
zzgl. Versandkosten
Lieferung in ca. 2 - 4 Werktagen

Finanzierung ab **43,37** €/mtl. mit einer Laufzeit von 72 Monaten. *

IN DEN WARENKORB

Deine gewählte **Konfiguration**

- **Gehäuse** CORSAIR 4000D AIRFLOW weiß, mit Seitenfenster (ATX) USB Typ-C-Anschluss nur in Verbindung mit entsprechendem Mainboard verfügbar
- **LED Stripes** Falls erwünscht, bitte auswählen
- **Prozessor** AMD Ryzen 9 5900X (12 x 3.70 - 4.80 GHz; 105W)
- **Prozessorkühler** 360mm-Radiator - CORSAIR iCUE H150i ELITE - ARGB
- **Arbeitsspeicher** 32 GB DDR4 3200 MHz (2x 16 GB) - CORSAIR VENGEANCE RGB PRO SL Series - RGB
- **Grafikkarte** 10 GB GDDR6X - NVIDIA GeForce RTX 3080 - ASUS TUF-RTX3080-O10G-V2-GAMING
- **Mainboard** MSI X570-A PRO (Chipsatz: X570 / ATX / DDR4)
- **M.2 SSD** 1 TB M.2 PCIe 3.0 x4 - WD Blue SN570 NVMe SSD
- **M.2 SSD** Keine Festplatte ausgewählt
- **SSD** 1 TB SATA III - Silicon Power Ace A55
- **SSD** Keine Festplatte ausgewählt
- **Festplatte** Keine Festplatte ausgewählt
- **Netzteil** 750 Watt - Corsair CX750M - 80+ Gold / Teilmodular
- **Farbige Sleeve Kabel & Kabelkämme** Falls erwünscht, bitte auswählen
- **Frontpanel** Falls erwünscht, bitte auswählen
- **Soundkarte** Onboard Sound
- **Bluetooth & WLAN** Falls erwünscht, bitte auswählen
- **Red Bull** Red Bull Energy Drink
- **Betriebssystem** Windows 10 Pro 64Bit (inkl. Installation auf das schnellste Speichermedium)
- **Software inkl. Installation** Falls erwünscht, bitte auswählen
- **Garantie & Service** 36 Monate Herstellergarantie inklusive kostenloser Rücksendung + Technischer Support

MySQL™

PostgreSQL

MariaDB®

Motivation: Cache Example

Imagine you work as a software engineer at an online retail company.

Your boss comes to you with the following task: „Our website currently does not have a caching system and our response time is too slow. We're losing revenue! Evaluate whether a cache system would make our website faster"

What would you do?

Motivation

Why do we need performance analysis?

- Compare alternatives to make informed decisions
- Understand influence of configuration option(s)
- System tuning
- Monitor relative performance over time
- Understand absolute performance for single case
- Set expectations (e.g., min/recommended system requirements for PC games)
- Analyze system behavior

Analysis Techniques

- Measurement: Collect data from real world
  - No simplifying assumptions
  - Most trustworthy results
  - Inflexible, one selected system
- Simulation
  - Abstraction
  - Less certainty whether it applies to real world
  - Flexible
- Analytical modelling
  - Mathematical description of system
  - Strong abstraction, results can be unrealistic
  - Useful for early validation

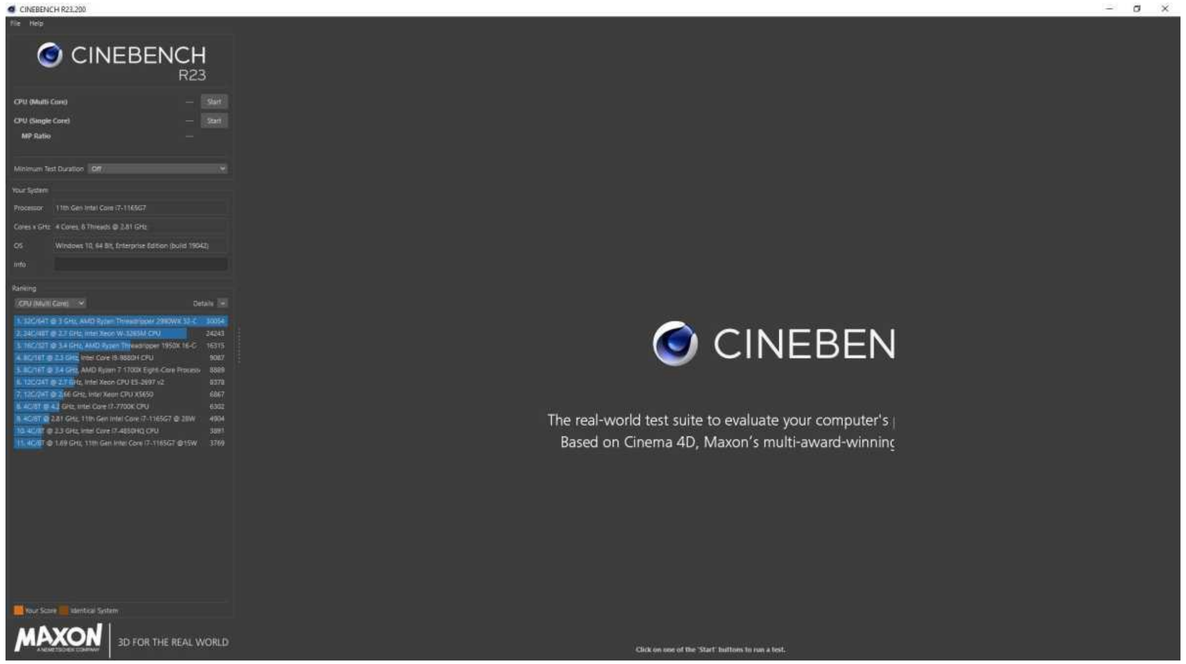
Analysis Techniques

- Measurement: Collect data from real world
  - No simplifying assumptions
  - Most trustworthy results
  - Inflexible, one selected system
- Simulation
  - Abstraction
  - Less certainty whether it applies to real world
  - Flexible
- Analytical modelling
  - Mathematical description of system
  - Strong abstraction, results can be unrealistic
  - Useful for early validation

Benchmark: Examples

Execute existing programs on existing hardware in a realistic environment to measure performance, memory consumption, …

- Cinebench (CPU)
- 3DMark (GPU)
- TPC-H (Datawarehouse)
- TCP-C (On-line transaction processing)
- Sintel (Video encoder)



Minimum Test Duration: Off

Your System

Processor	11th Gen Intel Core i7-1165G7
Cores x GHz	4 Cores, 8 Threads @ 2.81 GHz
OS	Windows 10, 64 Bit, Enterprise Edition (build 19042)
Info	

Ranking

1. 32C/64T @ 3 GHz, AMD Ryzen Threadripper 2990WX 32-C	30054
2. 24C/48T @ 2.7 GHz, Intel Xeon W-3265M CPU	24243
3. 16C/32T @ 3.4 GHz, AMD Ryzen Threadripper 1950X 16-C	16315
4. 8C/16T @ 2.3 GHz, Intel Core i9-9880H CPU	9087
5. 8C/16T @ 3.4 GHz, AMD Ryzen 7 1700X Eight-Core Process	8889
6. 12C/24T @ 2.7 GHz, Intel Xeon CPU E5-2697 v2	8378
7. 12C/24T @ 2.66 GHz, Intel Xeon CPU X5650	6867
8. 4C/8T @ 4.2 GHz, Intel Core i7-7700K CPU	6302
9. 4C/8T @ 2.81 GHz, 11th Gen Intel Core i7-1165G7 @ 28W	4904
10. 4C/8T @ 2.3 GHz, Intel Core i7-4850HQ CPU	3891
11. 4C/8T @ 1.69 GHz, 11th Gen Intel Core i7-1165G7 @15W	3769

The real-world test suite to evaluate your computer's
Based on Cinema 4D, Maxon's multi-award-winning

Click on one of the 'Start' buttons to run a test.

Benchmarks: Measures of System Performance

What can we measure?

- Execution time
- CPU cycles
- MIPS (Million instructions per second)
- MFLOPS (Million floating-point operations per second)
- SPEC (System Performance Evaluation Cooperative)
- QUIPS (Quality improvements per second)
- Transactions per second

Benchmarks: Measures of System Performance

What can we measure?

- Execution time
- CPU cycles
- MIPS (Million instructions per second)
- MFLOPS (Million floating-point operations per second)
- SPEC (System Performance Evaluation Cooperative)
- QUIPS (Quality improvements per second)
- Transactions per second

Are these good measures?

What criteria should a good measure fulfill?

Criteria of Measures

Meaningless indicator of performance?

Criterion	Description	Execution Time	MIPS	Transactions/sec
Linearity	Value in change of metric should be reflecting in change of value in performance	+	–	+
Reliability	Values correctly compare two systems	+	–	+
Repeatability	Value should lead to the same value for each experiment	+	+	+
Easy to measure		+	+	+
Consistency	Value should not depend on underlying architecture	+	–	+
Independence	Value cannot be (misleadingly) optimized	+	+	+

Confounding Parameters

- Confounding parameters influence measurement results in a systematical or unsystematical way
- For example: start-up time
  - Influences the results without intention
  - We may just care about actual performance
- Anything that can affect the measurement can be a confounding parameter

Confounding Parameters

- What are other potential confounding parameters of system performance?
    - Start-up time
    - Background processes/noise
        - System interrupts
        - On multi-core system: parallel processes
        - Random garbage collector events (e.g., Java)
    - Differences in hardware
    - Differences in environmental factors (e.g., temperature, …)
    - Input data (if unbalanced can heavily bias results)

Confounding Parameters

How can we control the influence of confounding parameters?

- Optimize benchmark (e.g., exclude start-up time)
- Control system (e.g., disable garbage collector)
- Control environment (e.g., system in a well-ventilated server room)

But can we do more?

Reducing Confounding Parameters

- Repeat measurement multiple times
- Best, 2nd best, or worst measurement
    - But, they can be misleading depending on scenario
    - Sometimes 99%, 10%, 1% percentile offers better insight on large samples
- Repeated measurements lead to larger samples
    - depending on type of data, they require a suitable statistical analysis

Later lecture

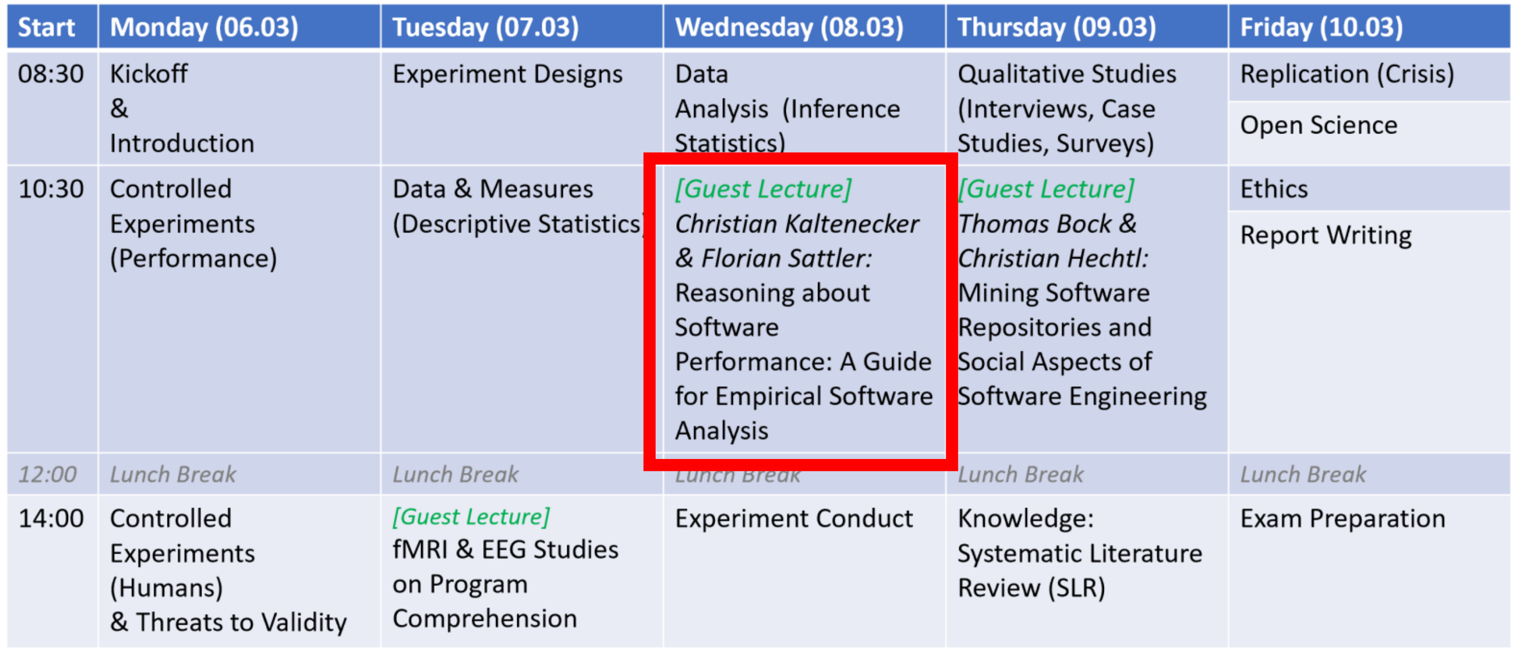
Later lecture

Example

- Imagine you want to buy a new computer and are deciding between two configurations your friend A and friend B have
    - You could run each of your use cases on their systems and compare the performance
- You must find good measures to evaluate performance
- You must consider confounding parameters
- Results can be indeterminate

Software Performance Analysis

- In Wednesday's guest lecture, Florian and Christian will cover analysis of software performance in much more detail



Start	Monday (06.03)	Tuesday (07.03)	Wednesday (08.03)	Thursday (09.03)	Friday (10.03)
08:30	Kickoff & Introduction	Experiment Designs	Data Analysis (Inference Statistics)	Qualitative Studies (Interviews, Case Studies, Surveys)	Replication (Crisis) Open Science
10:30	Controlled Experiments (Performance)	Data & Measures (Descriptive Statistics)	*[Guest Lecture] Christian Kaltenecker & Florian Sattler:* Reasoning about Software Performance: A Guide for Empirical Software Analysis	*[Guest Lecture] Thomas Bock & Christian Hechtl:* Mining Software Repositories and Social Aspects of Software Engineering	Ethics Report Writing
12:00	*Lunch Break*	*Lunch Break*	*Lunch Break*	*Lunch Break*	*Lunch Break*
14:00	Controlled Experiments (Humans) & Threats to Validity	*[Guest Lecture]* fMRI & EEG Studies on Program Comprehension	Experiment Conduct	Knowledge: Systematic Literature Review (SLR)	Exam Preparation

Further Methods: Logging

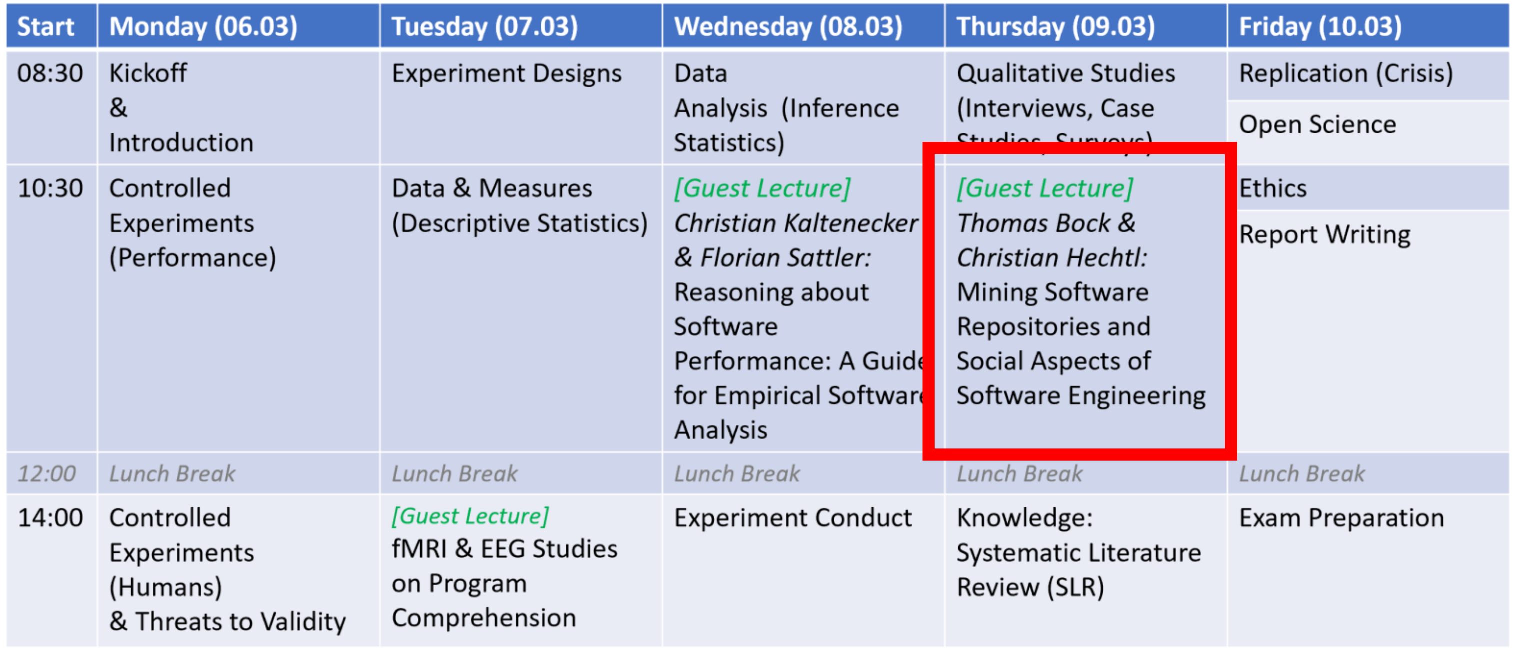
- Researchers can use already existing data (e.g., log files of a build server) rather than actively trigger data collection
    - Some options require an active implementation (e.g., logging of IDE interactions)
- Little time commitment for the researcher and somewhat accurate data
- Ethical concerns

Further Methods: Mining Software Repositories (MSR)

- Researchers can “mine” (open source) software repositories for source code, commit history, issues, …
  - Textual analysis (of code or documentation)
  - Static or dynamic analysis of software code
- Advantage: large amount of data that can be easily collected by the researcher
- Disadvantage: large amount of data
- Many more details during Thursday’s guest lecture
- Ethical concerns

Further Methods: Mining Software Repositories (MSR)

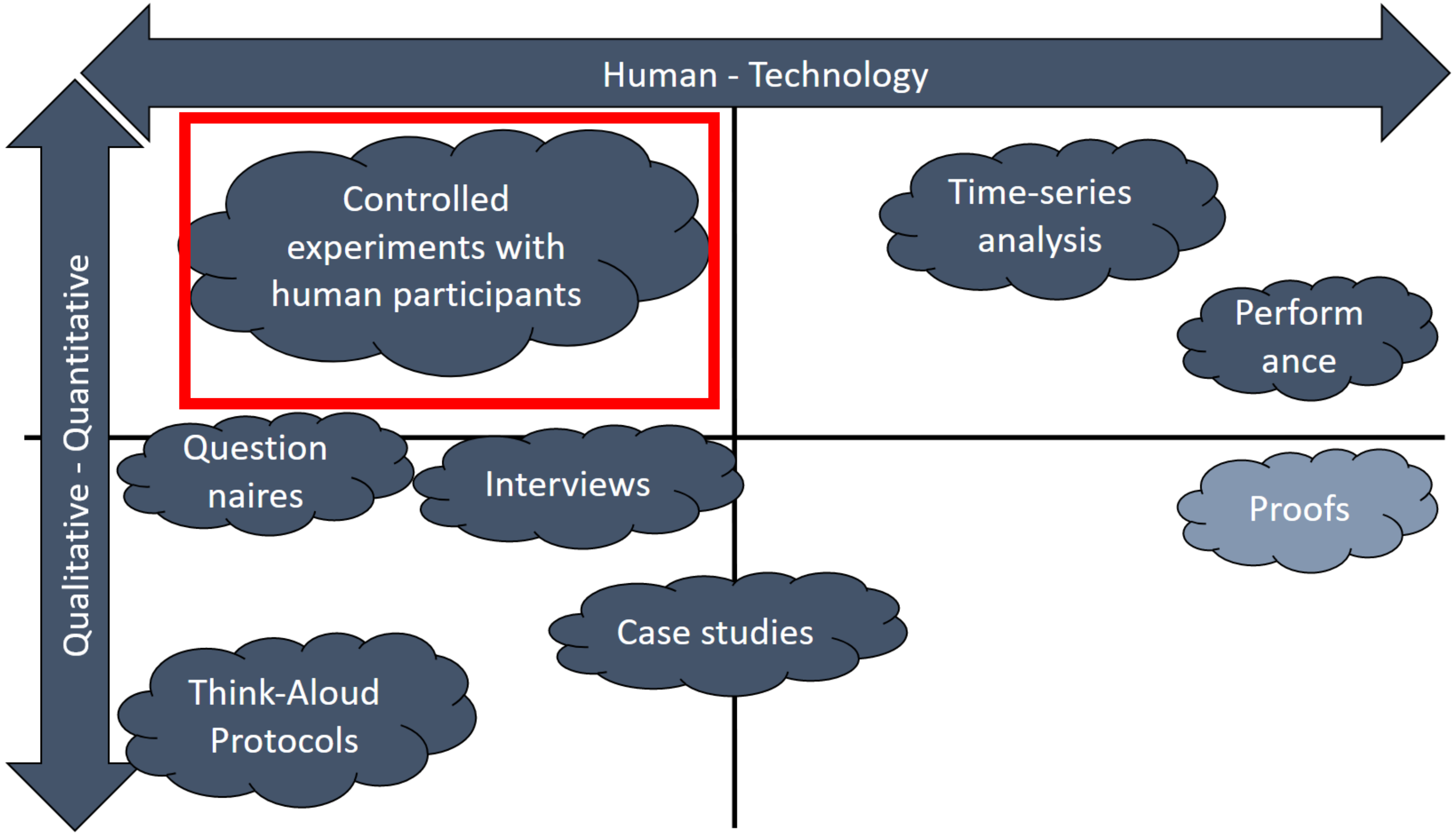
- In Thursday's guest lecture, Thomas and Christian will cover mining software repositories and the reality of research in much more detail



Start	Monday (06.03)	Tuesday (07.03)	Wednesday (08.03)	Thursday (09.03)	Friday (10.03)
08:30	Kickoff & Introduction	Experiment Designs	Data Analysis (Inference Statistics)	Qualitative Studies (Interviews, Case Studies, Surveys)	Replication (Crisis) Open Science
10:30	Controlled Experiments (Performance)	Data & Measures (Descriptive Statistics)	*[Guest Lecture]* *Christian Kaltenecker & Florian Sattler:* Reasoning about Software Performance: A Guide for Empirical Softwar Analysis	*[Guest Lecture]* *Thomas Bock & Christian Hechtl:* Mining Software Repositories and Social Aspects of Software Engineering	Ethics Report Writing
12:00	*Lunch Break*	*Lunch Break*	*Lunch Break*	*Lunch Break*	*Lunch Break*
14:00	Controlled Experiments (Humans) & Threats to Validity	*[Guest Lecture]* fMRI & EEG Studies on Program Comprehension	Experiment Conduct	Knowledge: Systematic Literature Review (SLR)	Exam Preparation

Controlled Experiments: Experiment Process & Human Participants

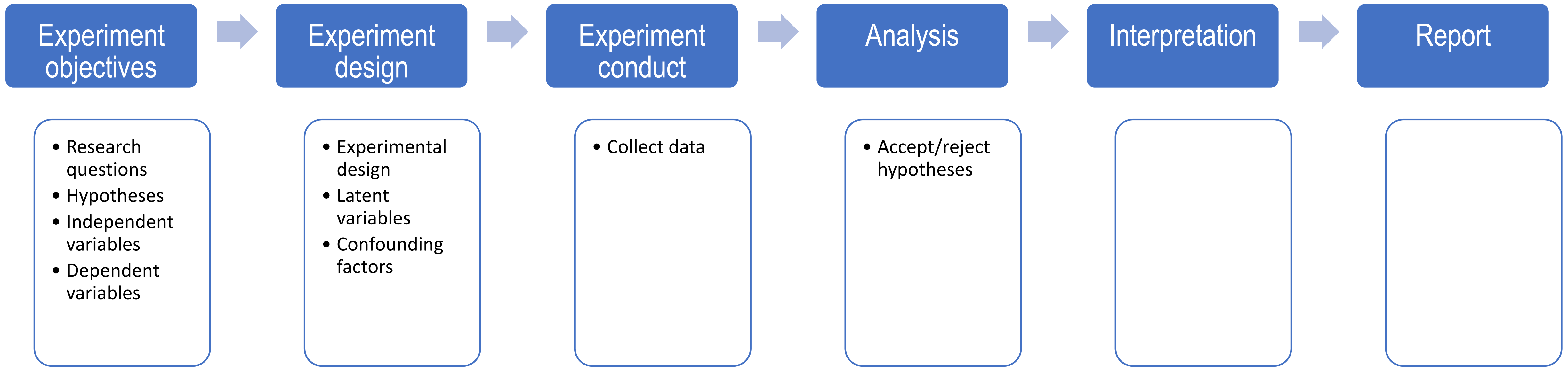
Context


Human - Technology
Qualitative - Quantitative
Controlled experiments with human participants
Time-series analysis
Perform ance
Question naires
Interviews
Proofs
Case studies
Think-Aloud Protocols

Learning Goals

- Familiarity with key terms of experiments (independent, dependent, latent variables)
- Being able to define high-quality research questions and hypotheses
- Understand different kinds of threats to validity and the trade-off between high internal and high external validity

Overview



- Research questions
- Hypotheses
- Independent variables
- Dependent variables

- Experimental design
- Latent variables
- Confounding factors

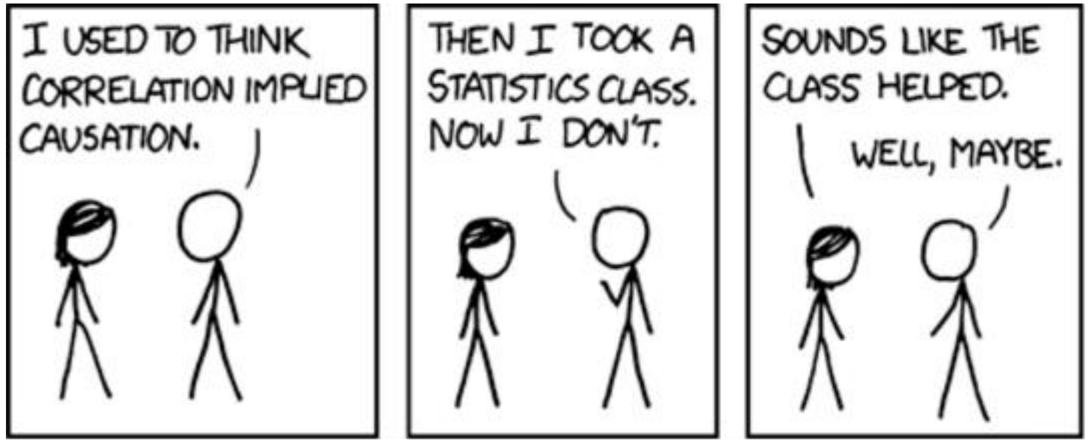
- Collect data

- Accept/reject hypotheses

Quality of the experiment can only degrade, not improve (a great analysis does not save bad design)

Controlled Experiments

- A controlled experiment is …
  - a systematic study
  - where one or more factors are varied
  - everything else is held constant
  - result of the systematic variation is observed


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.
Source: http://xkcd.com/552

- Usual goal: find descriptive or correlative relationship between variables

- Ultimate goal: observe causality: understand the effect of a factor

- A researcher aims to achieve their goals even in imperfect circumstances

Controlled Experiments

Example: Compare System A and System B via a benchmark

Historic Example: Galieo Galilei proposed around 1589 that heavier objects do not fall faster than lighter objects and conducted a series of experiments to provide empirical evidence

Experiment Objectives

Goal question metric (GQM):

Analyze <*Object of study*>

For the purpose of <*purpose*>

With respect to their <*quality focus*>

From the point of view of the <*perspective*>

In the context of <*context*>

Experiment Objectives

Goal question metric (GQM):

Analyze <Static or dynamic typing>

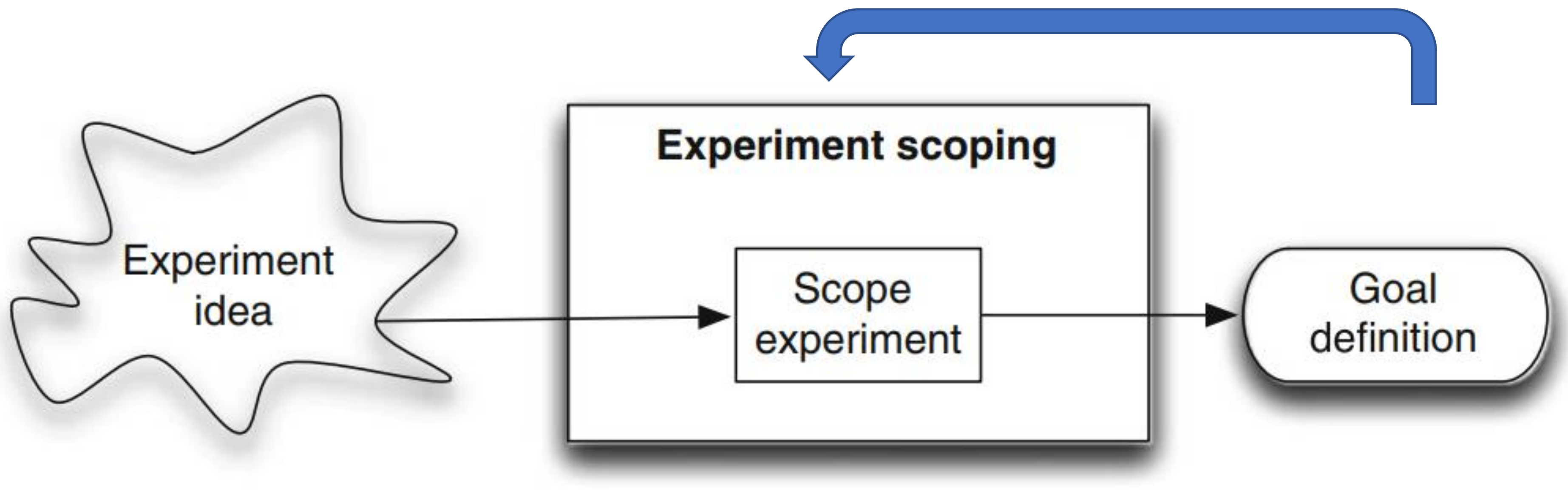
For the purpose of <*evaluation*>

With respect to <*program comprehension*>

From the point of view of the <programmer>

In the context of <*software code*>

Experiment Scoping



Independent Variable (IV)

$$\begin{tikzcd} \text{IV} \arrow[r] & \text{DV} \end{tikzcd}$$

- The independent variable is systematically and purposefully varied by the researcher
  - In some fields, they are named factor or predictor (variable)

- The independent variable typically has "levels"
  - In some fields, the levels are named alternatives or treatments

- Examples
  - Programming language (Python vs. Java)
  - IDE (Visual Code, IntelliJ IDEA, notepad++)
  - Teaching method (weekly, block)

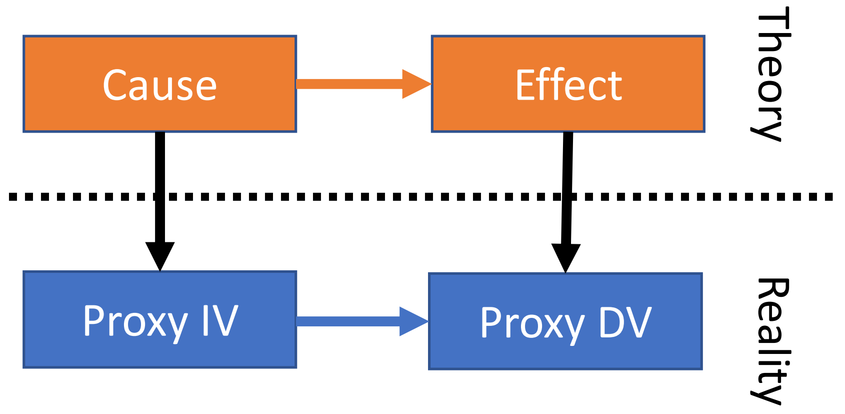
Dependent Variable (DV)

$$\begin{tikzcd} \text{IV} \arrow[r] & \text{DV} \end{tikzcd}$$

- The dependent variable is the result of an experiment that is observed by the researcher
- Ideally, it depends on the variation of the independent variable
- Examples
  - Productivity of a programmer
  - Bugs in a program
  - Exam grades (depending on teaching method)
  - Energy consumption of a system

Latent Variables

- „Constructs“ that cannot be observed directly
  - Researcher must find a measurable proxy for a latent variable
  - Often independent and/or dependent variables are latent variables



- Examples
  - Program comprehension is impossible to measure
    - But we can measure speed and correctness of tasks
  - Intelligence of a human is a construct
    - We can use an IQ test as a proxy
    - Typically, proxies have some limitations or only measure one part of the latent variable

Latent Variables: Operationalization

- Operationalization allows you to measure a latent variable
  - Latent variable → proxy measure
- Proxy measure must not contradict common sense

- Examples
  - Program comprehension
    - Task correctness
    - Task response time
    - Number of bugs
    - Subjective rating
    - …

They do not have to agree!

→ triangulation (multiple measures)

Exercise: Operationalization

Find an operationalization of the following latent constructs:

- Learnability of a programming language

- Maintainability of a software artifact

Exercise: Operationalization

Find an operationalization of the following latent constructs:

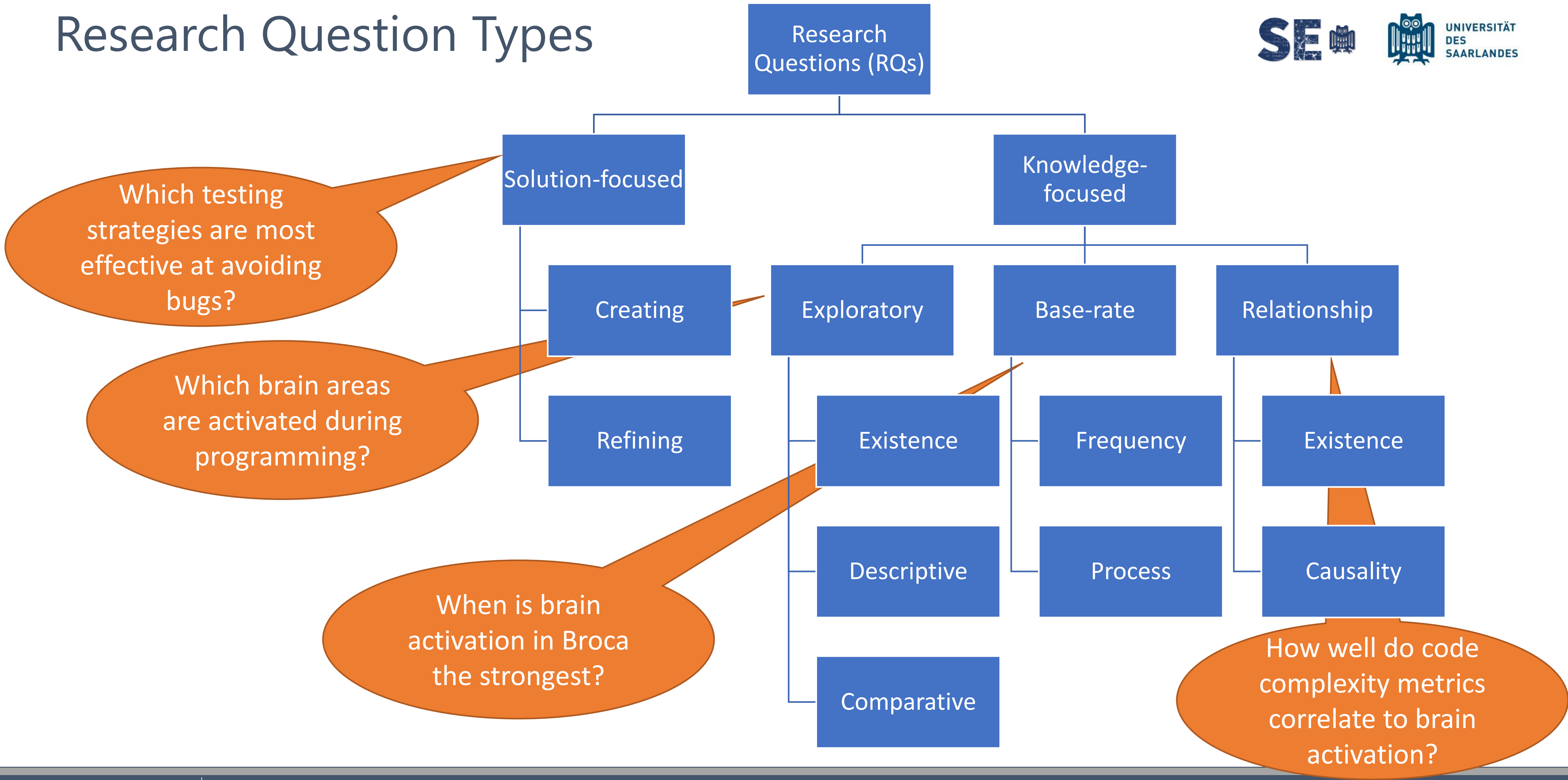
- Learnability of a programming language
  - Time to implement a task (e.g., sorting algorithm)
  - Subjective rating through a questionnaire/interview
  - Number of bugs found
  - …
- Maintainability of a software artifact
  - Similar operationalization as above
  - Number of present bugs
  - …

Research Questions & Hypotheses

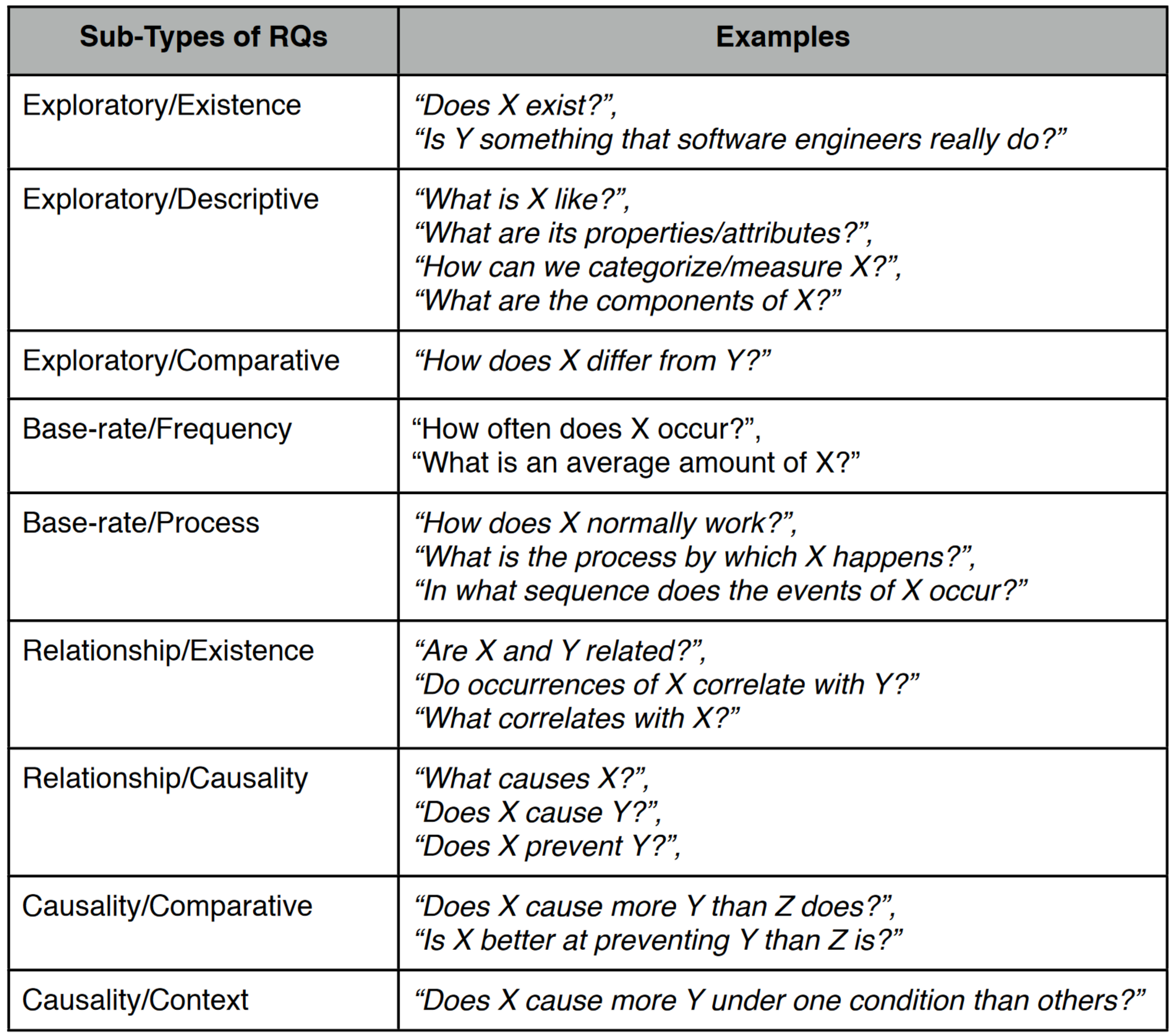
- Define research questions
  - Inquiries that guide our research
  - Clarify the objectives of the research to the reader (and to the researcher themselves)
  - Typically “How?”, “Why”, or “Which?” questions

- Hypotheses
  - Expectations of results
  - Expectations need to be justified (through theory, practice, or literature)
  - Must be simple and crystal clear
  - Must be falsifiable („Falsifiability“)

Research Question Types


Research Questions (RQs)
Solution-focused
Knowledge-focused
Creating
Refining
Exploratory
Base-rate
Relationship
Existence
Descriptive
Comparative
Frequency
Process
Existence
Causality
Which testing strategies are most effective at avoiding bugs?
Which brain areas are activated during programming?
When is brain activation in Broca the strongest?
How well do code complexity metrics correlate to brain activation?

Research Questions



Sub-Types of RQs	Examples
Exploratory/Existence	*“Does X exist?”,* *“Is Y something that software engineers really do?”*
Exploratory/Descriptive	*“What is X like?”,* *“What are its properties/attributes?”,* *“How can we categorize/measure X?”,* *“What are the components of X?”*
Exploratory/Comparative	*“How does X differ from Y?”*
Base-rate/Frequency	“How often does X occur?”, “What is an average amount of X?”
Base-rate/Process	*“How does X normally work?”,* *“What is the process by which X happens?”,* *“In what sequence does the events of X occur?”*
Relationship/Existence	*“Are X and Y related?”,* *“Do occurrences of X correlate with Y?”* *“What correlates with X?”*
Relationship/Causality	*“What causes X?”,* *“Does X cause Y?”,* *“Does X prevent Y?”,*
Causality/Comparative	*“Does X cause more Y than Z does?”,* *“Is X better at preventing Y than Z is?”*
Causality/Context	*“Does X cause more Y under one condition than others?”*

Finding Research Questions

- First, define what is your specific topic
- Second, do you want to increase your *understanding* or evaluate a *solution* to a problem?
  - Knowledge
    - If there is little existing knowledge → exploratory RQ
    - Some knowledge, but not when/how it occurs → base-rate RQ
    - Comprehensive knowledge, but miss underlying causes → relationship RQ
- If there are multiple (sub)questions, use multiple research questions to envelop your research (and possible narrow down later)

Hypotheses

- Research questions looks for an open-ended answer
- Hypotheses are derived from the research questions and more specifically look for a binary decision (support/reject)

- H1: Bad source-code comments are bad for program comprehension
- H2: Good source-code comments are good for program comprehension

Hypotheses

→ **Comments describing every statement of source code** do not affect the **time programmers need to understand** a Java source-code snippet

→ **Comments containing incorrect information** decrease the **task correctness of programmers understanding** a Java source-code snippet

→ **Comments describing the purpose of source-code statements** decrease the **time programmers need to understand** a Java source-code snippet

Hypotheses

Why do we need hypotheses?

- Hypotheses guide us when designing an experiment
- Prohibit fishing for results
- Connects theory and empirical research
  - Derived from theory
  - Evaluated with empirical research

Hypotheses

Develop a hypothesis on the following research questions

- Does Python increase productivity of programmers?
- Is Python better than Java?
- Is the new UI more productive?

Reminder: hypotheses need to be evaluated, justified, falsifiable and variables operationalized

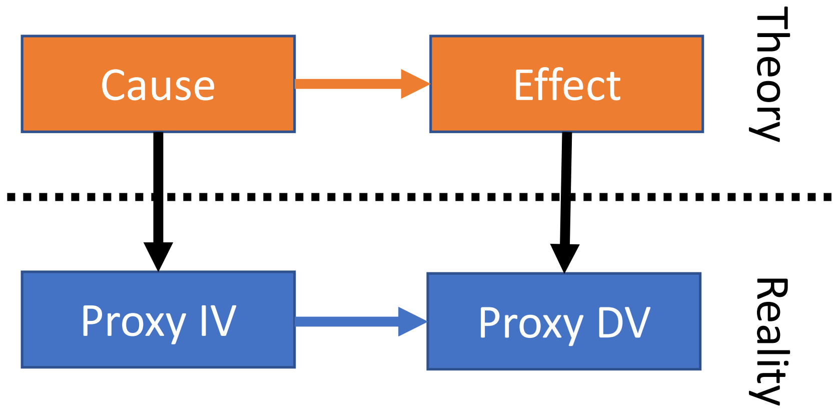
Further Reading

- Guide to create research qestions:

  http://robertfeldt.net/advice/guide_to_creating_research_questions.pdf

Threats to Validity

Threats to Validity



Do we measure what we want to measure?

- Construct validity
- **Internal validity**
- **External validity**
- Conclusion validity

Important to understand to integrate an experiment into the overall picture!

Often a required section by publication vendors

- Many more… (https://en.wikipedia.org/wiki/Validity_(statistics))

Construct Validity

Construct validity describes how well the construct is being measured

- Inadequate theory or measurement bias
- For example
  - Number of syntax errors found as measure of programming skill (experts tend to miss syntax errors)
  - IQ test may not measure (a certain type of) intelligence well

Internal Validity

Internal validity is the amount to which the value of the dependent variable can be explained by the systematic variation of the independent variable

- For example, threats to validity can be…
  - Learning effects
  - Maturation
  - Mortality

External Validity

External validity is the amount to which the results are transferable to other circumstances (including participants, material, ...)

For example, can we generalize our study on novice programmers to all programmers?

Conclusion Validity

Conclusion validity describes how suitable and appropriate the used statistical method was for analyzing the data

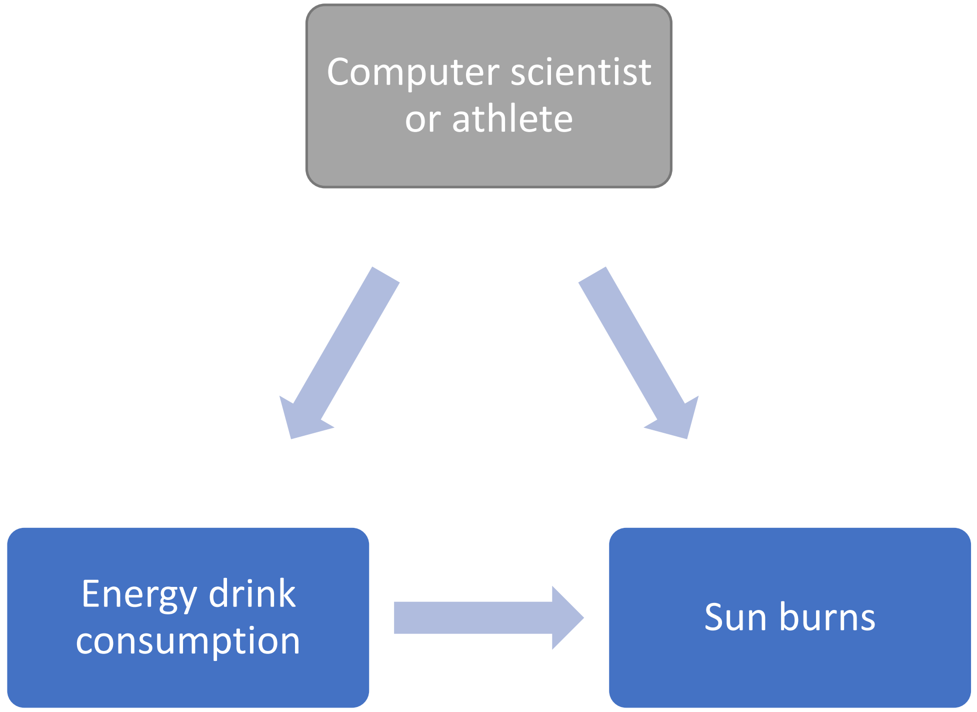
- Sometimes referred to as statistical conclusion validity
- For example, threats to conclusion validity could be:
  - The use of a parametric test on non-parametric data
  - Including a confounding factor requires a specific statistical test

Confounding Factors

Confounding factors are an extra variable that is related to the independent and dependent variables of the experiment.

They pose a serious threat to (internal) validity.

Example: omitted variable bias



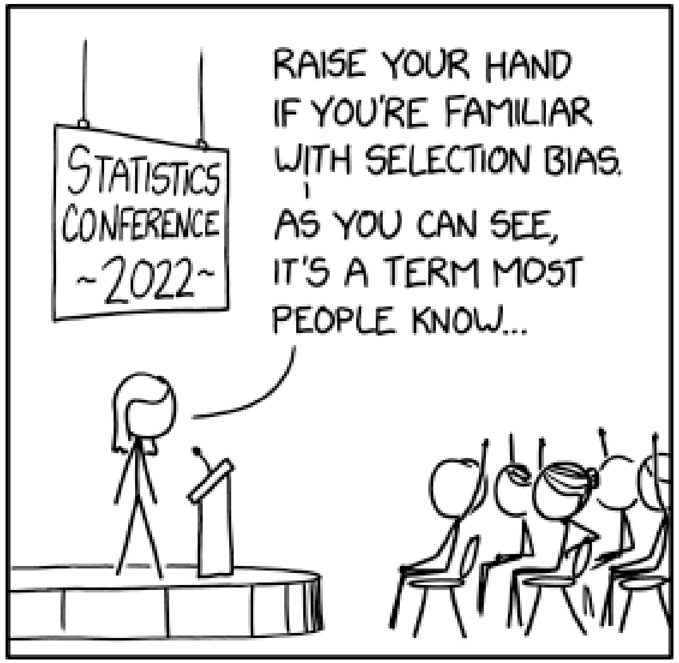
Confounding Factors

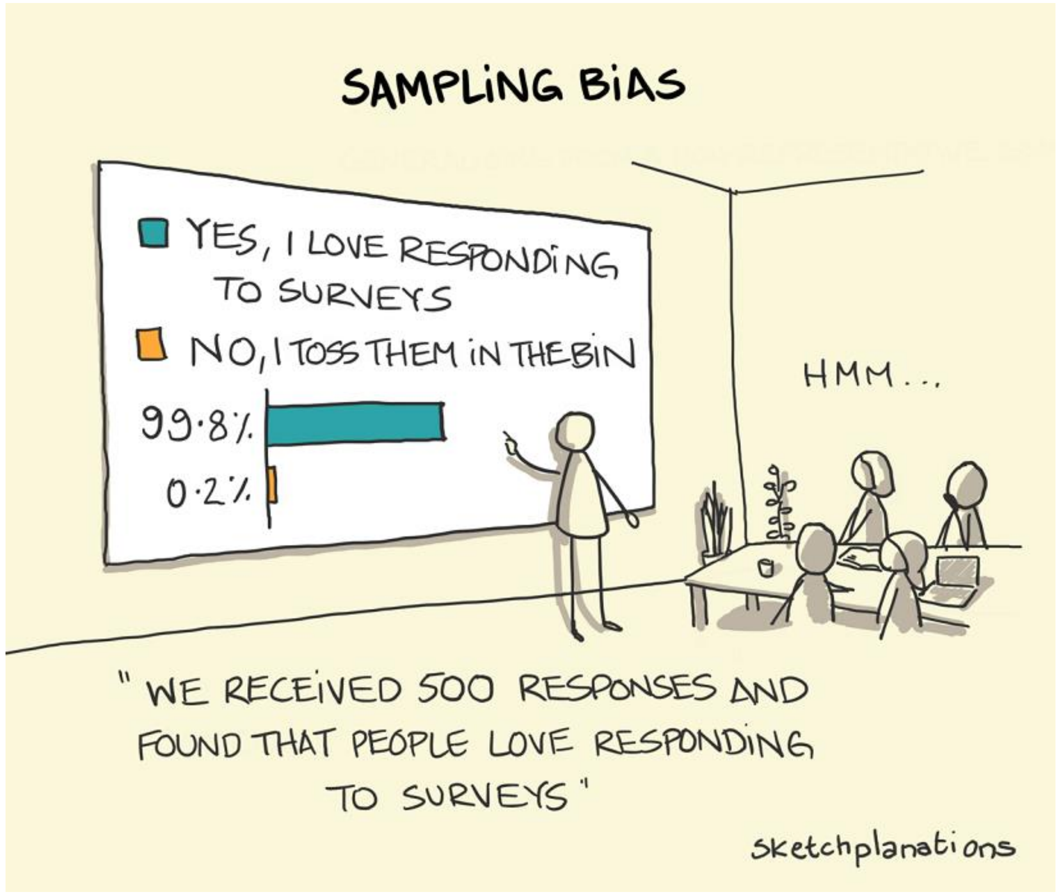
The Hawthorne effect describes participants changing their behavior when they know they are being observed

Hawthorne Labs
Just verbalize whatever comes to mind.
I hope I say the right thing to get my $100 and pizza slice!
YouShouldTestThat.com
c goward

Confounding Factors

- Selection bias: selection is not representative
- Sampling bias: chance of being part of the sample


STATISTICS CONFERENCE ~2022~
RAISE YOUR HAND IF YOU'RE FAMILIAR WITH SELECTION BIAS.
AS YOU CAN SEE, IT'S A TERM MOST PEOPLE KNOW...


SAMPLING BIAS
YES, I LOVE RESPONDING TO SURVEYS
NO, I TOSS THEM IN THE BIN
99·8%
0·2%
HMM...
"WE RECEIVED 500 RESPONSES AND FOUND THAT PEOPLE LOVE RESPONDING TO SURVEYS"
sketchplanations

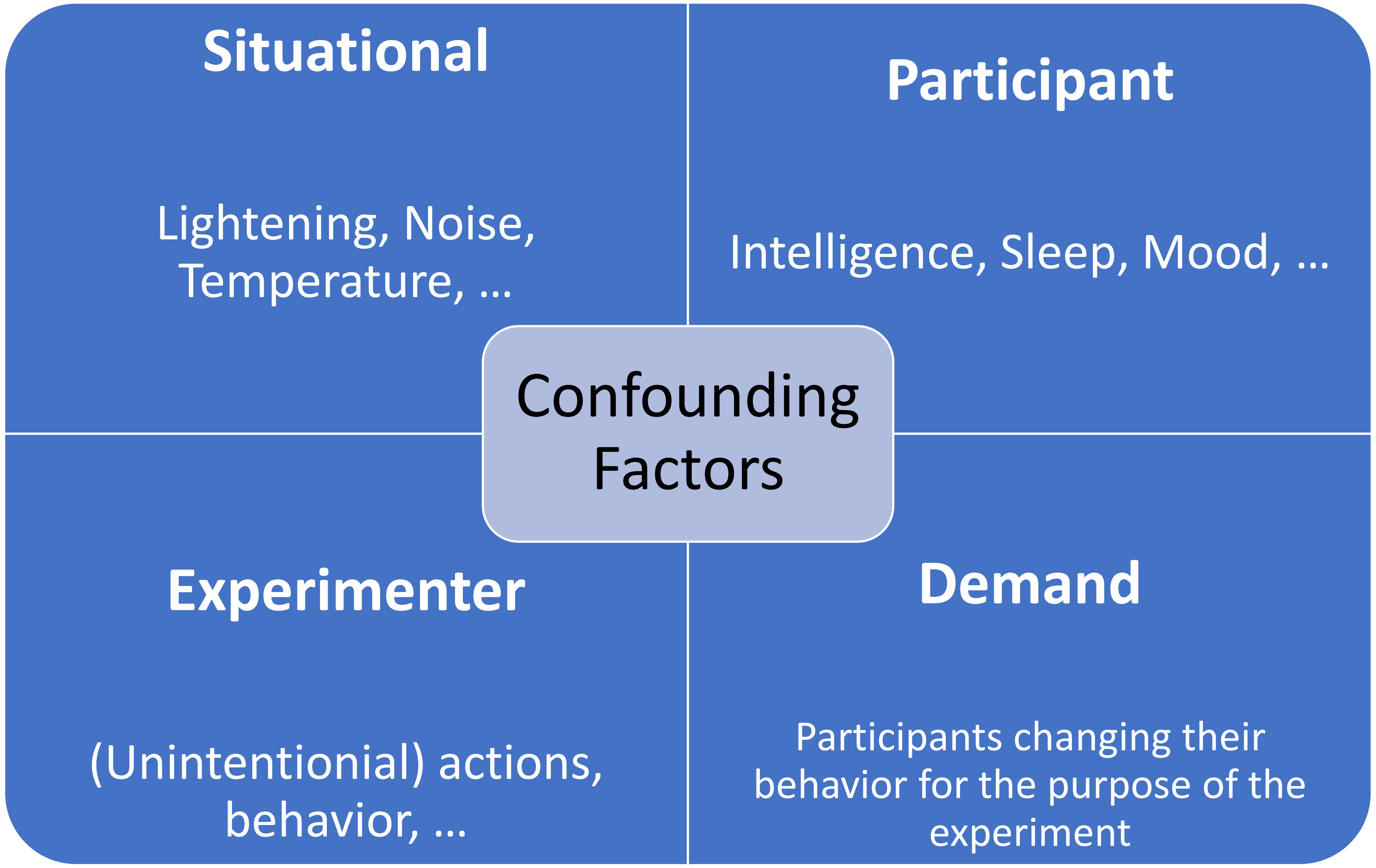
Confounding Factors

Confounding factors can be very subtle but still highly influential, despite a great study design. If a researcher does not consider an important confounding factor, the observed relationship may not exist at all (or be estimated incorrectly).

Examples:

- Margaret Shih’s study on prejudices of Asian women on their math ability
- Experimenter‘s aftershave on experienced pain

Confounding Factors


Situational
Lightening, Noise, Temperature, …
Participant
Intelligence, Sleep, Mood, …
Confounding Factors
Experimenter
(Unintentionial) actions, behavior, …
Demand
Participants changing their behavior for the purpose of the experiment

Confounding Factors

In general, researchers should try to consider confounding factors

- But, there is an infinite number of confounding factors → perfection is impossible
- Some confounding factors can only be identified through replications
  - will discuss those in more detail on Friday

Confounding Factors

If your experiment measures program comprehension…

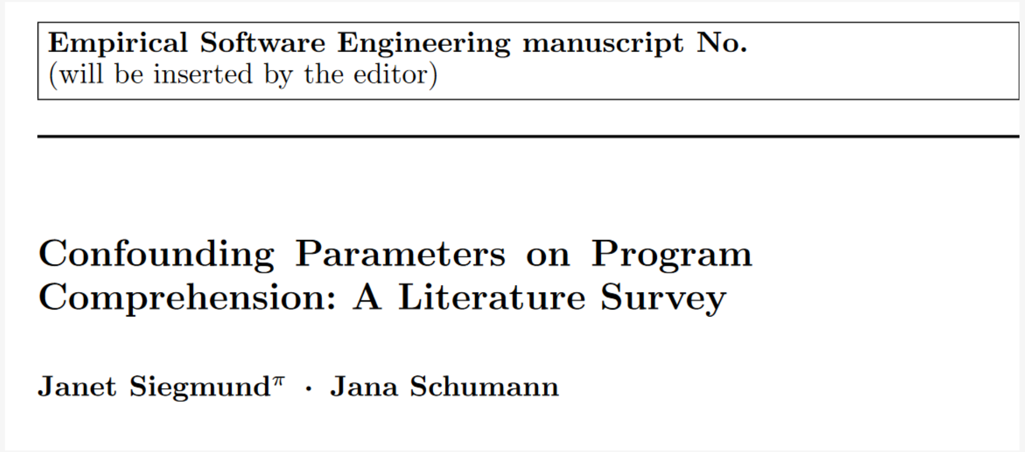
- What are potential confounding factors?

- How can you control the confounding factors?

Confounding Factors

If your experiment measures program comprehension…

- What are potential confounding factors?
  - Programming experience
  - Experience with programming language
  - Familiarity with the IDE
  - Intelligence
  - plus at least 35 more …
  - Paper: https://www.infosun.fim.uni-passau.de/cl/publications/docs/SiSchu14.pdf


Empirical Software Engineering manuscript No.
(will be inserted by the editor)

Confounding Parameters on Program Comprehension: A Literature Survey

Janet Siegmund$^{\pi}$ · Jana Schumann

Confounding Factors

If your experiment measures program comprehension…

- How can you control the confounding factors?
- → Identify potential confounding factors and understand your sample!
    - Randomization
    - Matching/parallelization/balancing
    - Adapt confounding factor as independent variable
    - Keep confounding factor constant
    - Analyze data afterwards

Randomization

- Random number generator, throw a dice, …
  - randomization of stimuli reduces chance of a learning effect
  - randomization of participants reduces chance of a selection bias
  - …

- Experiment often requires more participants
  - 5 per group often too low
  - >= 10 generally seems a reasonable minimum

Matching/Parallelization/Balancing

- If you do not have enough participants for randomization
- Odd-even-even-odd principle

- Must have a measure for the confounding factor (program compr., IQ test, …)
  - Measure takes a long time before the experiment start (questionnaire, interview)
  - Hard to plan in advance
- Advantage over randomization: controlled assignment, don't have to rely on large numbers

Confounding Factor → Independent Variable

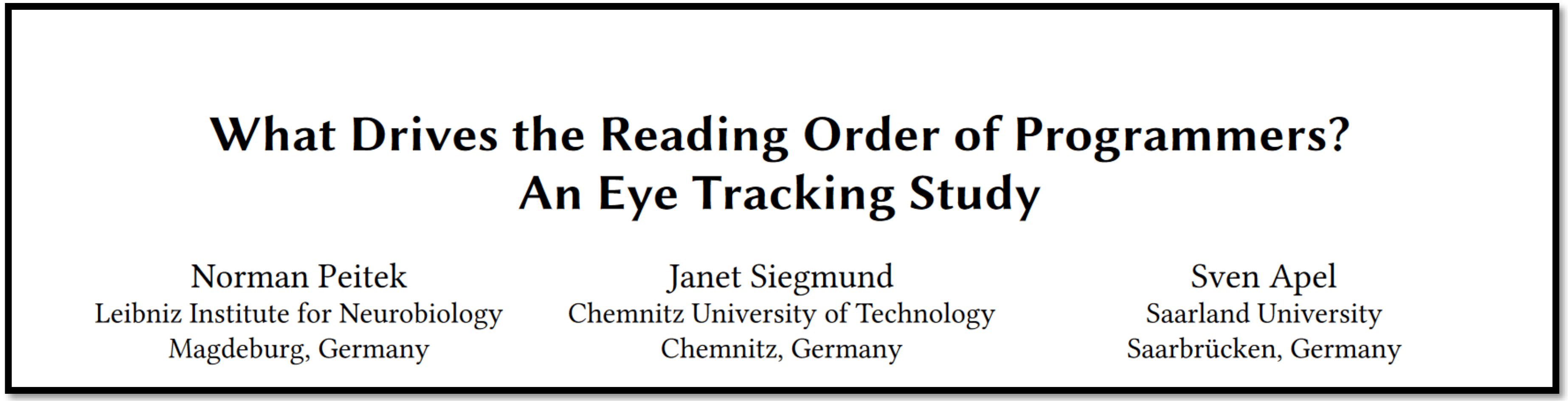
- Varied systematically by the researcher
- Means confounding factor is operationalized (which can be tricky)
- Increases complexity of the experiment
  - 2 levels x 2 levels of experience = 4
  - Requires more participants
  - Complex data analysis
- But there are limits!
  - 23 confounding factors, each with two levels, 10 participants per group
  - … >83 million participants

Confounding Factor → Constant

- Only one level of confounding factor ("controlled variable")
- Programming experience:
  - Only undergraduate students
  - Only experts
  - No experience
- Intelligence
  - Minimum grades in school
- But, selection comes with assumptions and only provides some control
  - An improvement over doing nothing

Confounding Factor → Analyze Afterwards

- Measure confounding factor
- Analyze whether confounding factor influenced the dependent variable afterwards
  - Especially useful if balancing or similar are not possible
  - Could make an experiment "useless" as it may drive the results



What Drives the Reading Order of Programmers? An Eye Tracking Study

Norman Peitek
Leibniz Institute for Neurobiology
Magdeburg, Germany

Janet Siegmund
Chemnitz University of Technology
Chemnitz, Germany

Sven Apel
Saarland University
Saarbrücken, Germany

Relationship between Internal and External Validity

- Internal and external validity request different things
  - Internal validity: control everything
    - Focus on independent variable, but at a cost of external validity
    - For example, invite a small subgroup of a population (student programmers), but then cannot generalize results to all programmers
  - External validity: general setting
    - Invite a large sample of all programmers, but may miss relevant differences factors

Relationship between Internal and External Validity

- What to focus on with your first experiment?

- Generally, conduct experiments with high internal validity first
  - An experiment about static and dynamic type systems, Stefan Hanenberg
  - Extremely high internal validity with made-up language, no IDE, etc.
  - Then, conducted experiments with actual programming languages, in IDEs, etc.

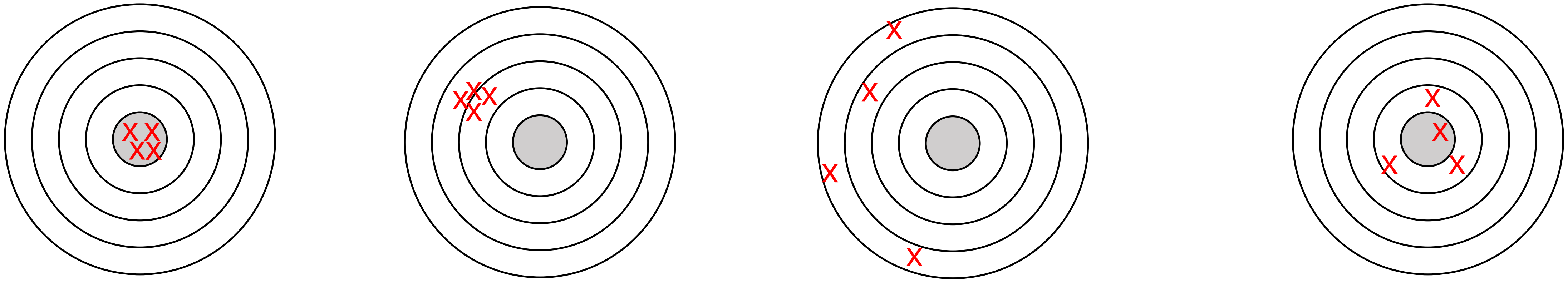
- Then, with experience, increase external validity step by step
  - If internal validity is reduced too much, it is not a *controlled* experiment anymore

Quality Criteria of Empirical Studies (1)

- Relevance
  - Does the experiment target an important question, and can it provide answers?
- Objectivity
  - Are the results independent from the specific researcher who conducted the experiment?
  - Does not have to be intentional (e.g., Asian women math study)
  - Typically, can be tested by replications (but they often vary other factors as well)

Quality Criteria of Empirical Studies (2)

- Validity
- Reliability
  - How accurate are the results?



Motivation: Cache Example

Imagine you work as a software engineer at an online retail company.

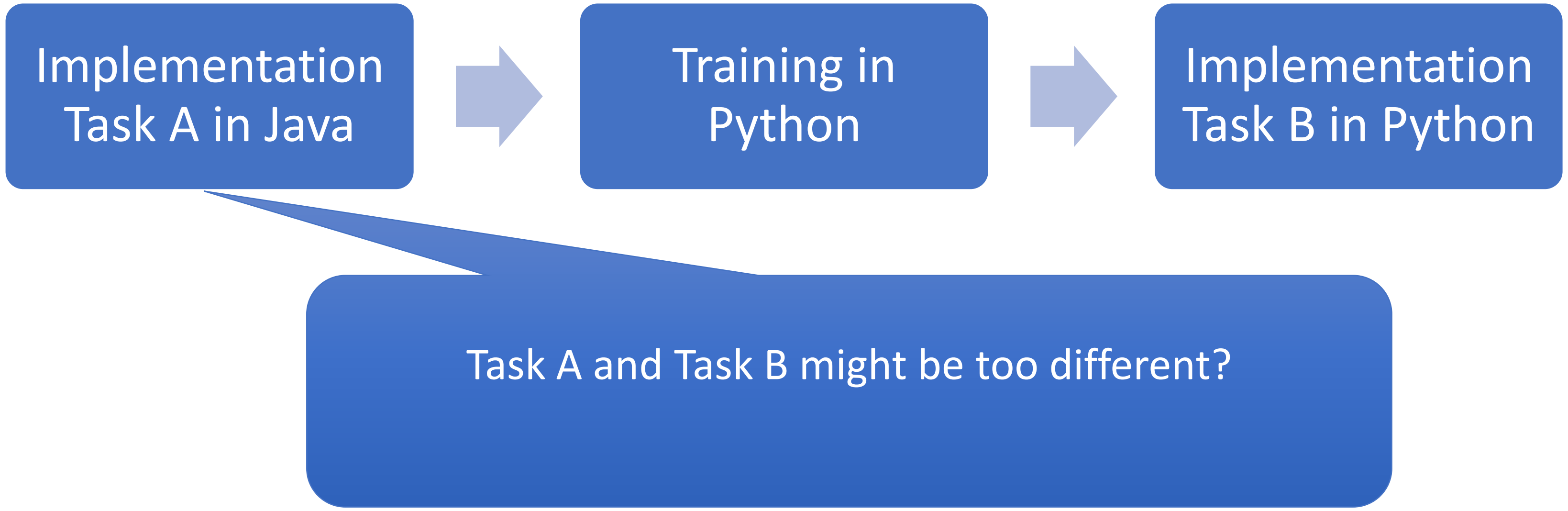
Your boss comes to you with the following task: „Our website currently does not have a caching system and our response time is too slow. We‘re losing revenue! Evaluate whether a cache system would make our website faster“

Again, what would you do?

Examples of a ~~Controlled~~ Experiment

What are and what are not controlled experiments?

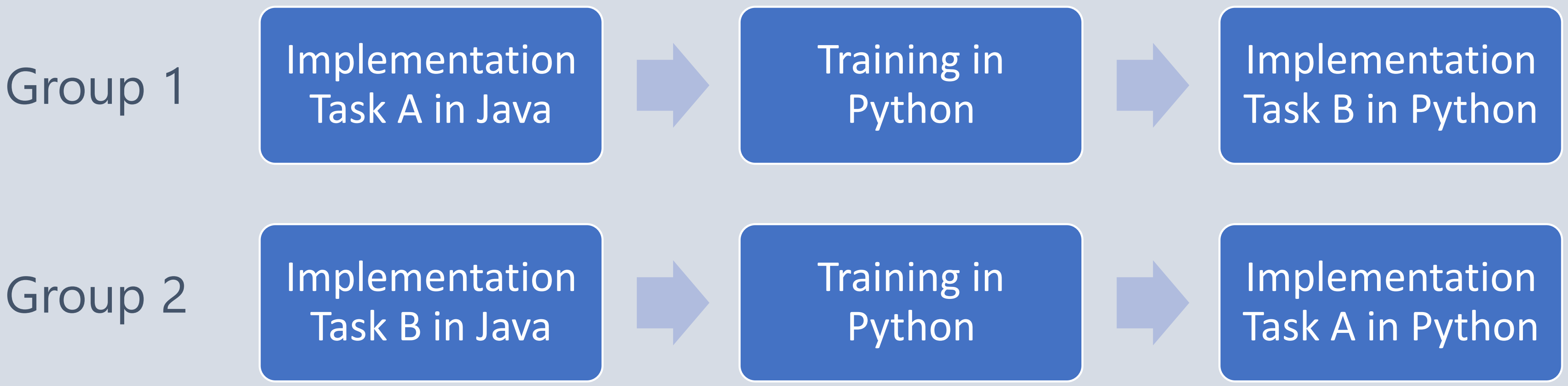
- A team lead intends to evaluate the effectiveness of a technical training:


Implementation Task A in Java
Training in Python
Implementation Task B in Python
Task A and Task B might be too different?

Examples of a ~~Controlled~~ Experiment

What are and what are not controlled experiments?

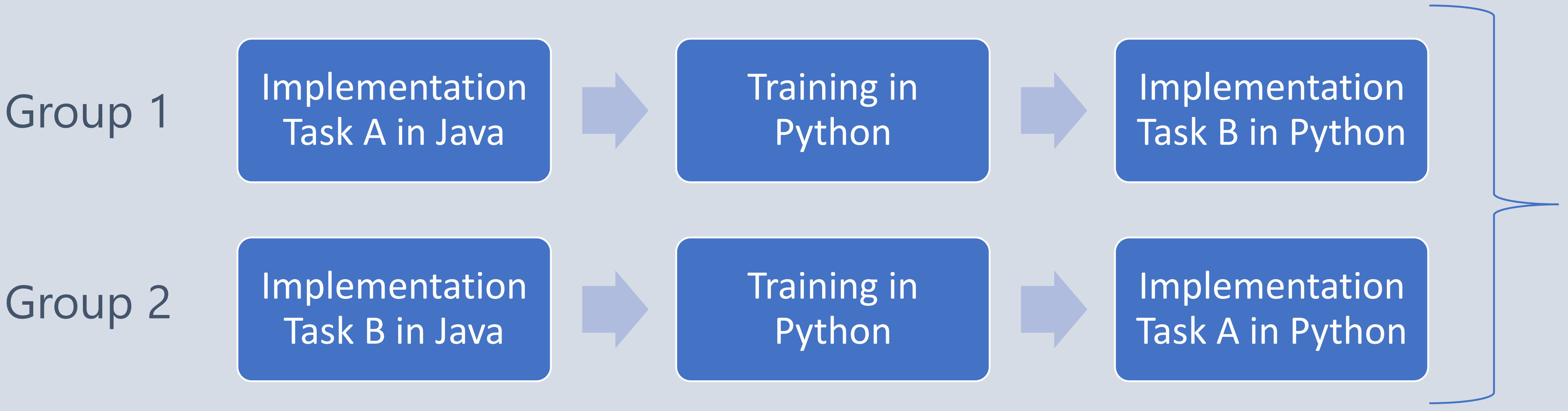
- A team lead intends to evaluate the effectiveness of a technical training:


Group 1
Implementation Task A in Java
Training in Python
Implementation Task B in Python
Group 2
Implementation Task B in Java
Training in Python
Implementation Task A in Python

Examples of a ~~Controlled~~ Experiment

What are and what are not controlled experiments?

- A team lead intends to evaluate the effectiveness of a technical training:

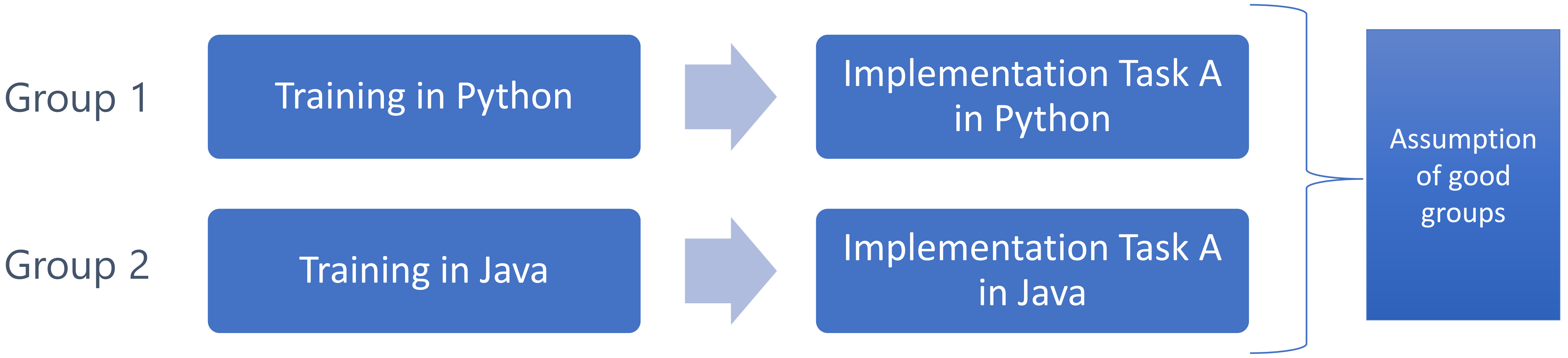

Group 1
Implementation Task A in Java
Training in Python
Implementation Task B in Python
Group 2
Implementation Task B in Java
Training in Python
Implementation Task A in Python

Fails to control for many factors: Prior knowledge, individual differences, …

Examples of a ~~Controlled~~ Experiment

What are and what are not controlled experiments?

- A team lead intends to evaluate the effectiveness of a technical training:


Group 1
Training in Python
Implementation Task A in Python
Group 2
Training in Java
Implementation Task A in Java
Assumption of good groups