The Role of Empiricism

Lutz Prechelt, Freie Universität Berlin

V+Ü "Empirische Methoden im Software Engineering"

- Empirical, method, SW engineering
  - SW engineering is highly socio-technical
  - SW engineering is about usefulness
  - Theory, Construction, Empricism
- The role of empirical methods
- Credibility, Relevance
  - and where they come from
- Quantitative vs. qualitative
- 4 method archetypes
- Some useful study types (+example)
- Epistemological stance
  - Positivist, interpretivist
  - induces different cultures

Die Rolle empirischer Methoden

Lutz Prechelt, Freie Universität Berlin

V+Ü "Empirische Methoden im Software Engineering"

- Empirisch, Methode, SW Engineering
  - SW Engineering ist sehr sozio-technisch
  - Es geht um Nützlichkeit
  - Theorie, Konstruktion, Empirie
- Die Rolle empirischer Methoden
- Glaubwürdigkeit, Relevanz
  - und was sie erzeugt
- Quantitative vs. qualitative Methoden
- 4 Methoden-Archetypen
- Beispiele für nützliche Studienformate
- Epistemologischer Standpunkt
  - Positivistisch, interpretivistisch
  - führt zu verschiedenen Kulturen

"Empirical" / "Empirische"

- Based on observation
  - (greek-latin origin)

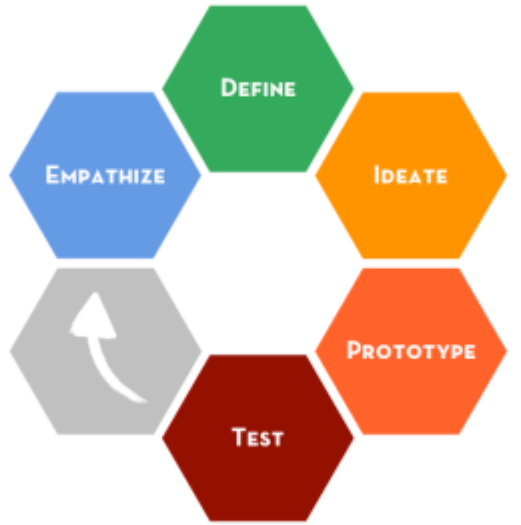
- As opposed to being based on
  - theoretical considerations
    - involving deduction, induction, abduction
  - intuition
  - random selection

http://www.microscopy-uk.org.uk/mag/imgdec04/chPicture010.jpg

"Method" / "Methode"

- 1: "a procedure or process for attaining an object, such as:
  - 1a(1): a systematic procedure, technique, or mode of inquiry employed by or proper to a particular discipline or art […]
  - 1b(1) : a way, technique, or process of or for doing something
  - 1b(2) : a body of skills or techniques"

- https://www.merriam-webster.com/dictionary/method



Example:
Basic steps of Design Thinking

Our definition:

- The body of knowledge of how we produce software
  - as systematically as we can.
  - Some of the knowledge was produced by empirical research.
- The practice of producing software professionally

- SW engineering is the part of computer science which is too difficult for the computer scientist.
  --Friedrich L. Bauer, 1971.
- Einstein argued that there must be simplified explanations of nature, because God is not capricious or arbitrary.
  No such faith comforts the SW engineer.
  --Fred Brooks, "No Silver Bullet", 1987
- The amateur SW engineer is always in search of magic, some sensational method or tool whose application promises to render SW development trivial.
  It is the mark of the professional SW engineer to know that no such panacea exists.
  --Booch, Maksimchuk, Engle, 2007

//spin.atomicobject.com/wp-nt/uploads/softwater-team-meeting.jpg

https://en.wikiquote.org/wiki/Software_engineering

Some ways in which software engineering differs:

1. Almost pure design
   - the building step is automated
2. 1 → Defects correctable cheaply
3. 1 → frequent requirements changes
4. Almost infinite flexibility
   - (no limitations from materials properties)
5. 4 → Easy packaging of standardizable solutions (libraries, frameworks, tools)
   - often Open Source (→no monetary cost)
6. 2+4+5 → Small teams construct systems with very many unique parts
7. 1+6 → High frequency of design decisions
8. 7 → Need for creative work by all team members
9. 1+2+3+4+5+6+7+8 → Strong socio-technical component

Software Engineering is a form of engineering and a social science

Very important slide!

Engineering is about usefulness

- Frederick Brooks: "The Computer Scientist as Toolsmith II", CACM 1996

The scientist *builds in order to study;*
the engineer *studies in order to build.*

- Science is about knowledge
- Engineering is about usefulness
  - Cf. the IEEE's mission statement:
    "IEEE's core purpose is to foster technological innovation and excellence for the benefit of humanity."

Partitioning software engineering: Theory, Construction, Empiricism (T, C, E)

We partition by work method:

- Theory T ("modelling")
  - produces formalisms, derives results about them,
    revolves around logical issues
- Construction C
  - produces systems designs, constructs systems,
    revolves around practical issues
- Empiricism E
  - produces observations of systems and interprets them,
    revolves around behavior in and of the real world

- At any one time, any work in Software Engineering (and in Informatics in general)
  is primarily in only one of these modes
  - whether practitioner work or research work.
  - Good work switches mode frequently (iterative work style)

- The "system" may be
  - a SW product (or part thereof) or
  - a SW development process (or part thereof)
- We talk mostly about the latter case

T,C,E example 1: Developing an algorithm

- Theory
  - Specify the problem to be solved
    - e.g. linear programming: minimize linear function given constraints
  - Specify an algorithm for solving it (e.g. simplex algorithm)
  - Maybe prove the algorithm correct, etc.

- Construction
  - Implement the algorithm as a concrete program
    - often much longer than the theoretical algorithm because of optimizations, input/output, limitations of machine arithmetic, error handling, external interfaces, etc.

- Empiricism
  - Determine actual characteristics of the program for different kinds of inputs
    - execution time, memory behavior, etc.
    - for heuristic or approximation algorithms: quality of results (e.g. in machine learning)

T,C,E example 2: Developing a software design method

- Empiricism
  - Determine the weaknesses of current design methods
- Theory
  - Maybe define some new terminology
  - Maybe pose new design principles
- Construction
  - Formulate a new design method
  - Perhaps construct support tools
- Empiricism
  - Evaluate the behavior of the method for concrete problems in practice
  - Probably in comparison to other methods
    [comparison is difficult (compare A to B) or expensive (compare A to A')!]

T,C,E example 3: Introducing a well-known process element

Freie Universität Berlin

- e.g. introducing Test-Driven Development (TDD) into your local SW development process:

or: any development tool, programming language, framework, etc.

- Construction
  - Learn TDD (each team member)
  - Decide where to use it and where not

- Empiricism
  - Evaluate the impact on (a) development time, (b) defect density, (c) readability and design quality, (d) modifiability.
  - Either by developing some module multiple times [expensive]
  - or by comparing "typical" values [less reliable and convincing]
  - If you expect or find the benefit varies greatly, find out how those benefits materialize (or not) [qualitative research: difficult]

The role of empirical methods

- An empirical method is a template for one type of approach to empirical work
- Each method has
  - different applicability
    - research question, research context
  - different tradeoffs
    - type of work, amount of work, types of results, attributes of results
- Knowing enough methods allows conscious decisions and tradeoffs
- We think about each method in terms of an empirical study:
  - Decide research question or interest
  - Design, execute, and evaluate study
    - set up data collection, collect data, analyze/evaluate data
      - sequential or iterative
  - Formulate conclusions
  - Present study and conclusions
    - write a report, give a presentation
- We think about each use of a method in terms of two outcome quality criteria:

Overall quality applies to the conclusions from a study:

- **Credibility**
  - How trustworthy are the conclusions?
- **Relevance**
  - How interested are we in these conclusions?
    How beneficial is it to have them?

In practice, many studies do not actually formulate conclusions

Where does credibility come from?

1. Study purpose is clear, authors are open towards any result
   - rather than "We will now show that our new X is superior."
2. Study setup is adequate, described in detail, and easy to understand
   - We can see that work has been performed carefully
3. Results are easy to grasp ("anschaulich")
   - rather than abstract or contrived
4. Report convincingly discusses the limitations of the evaluation
   - rather than glossing over its flaws
5. Conclusions clearly follow from the data
   - in particular: sensible operationalizations, no overgeneralization

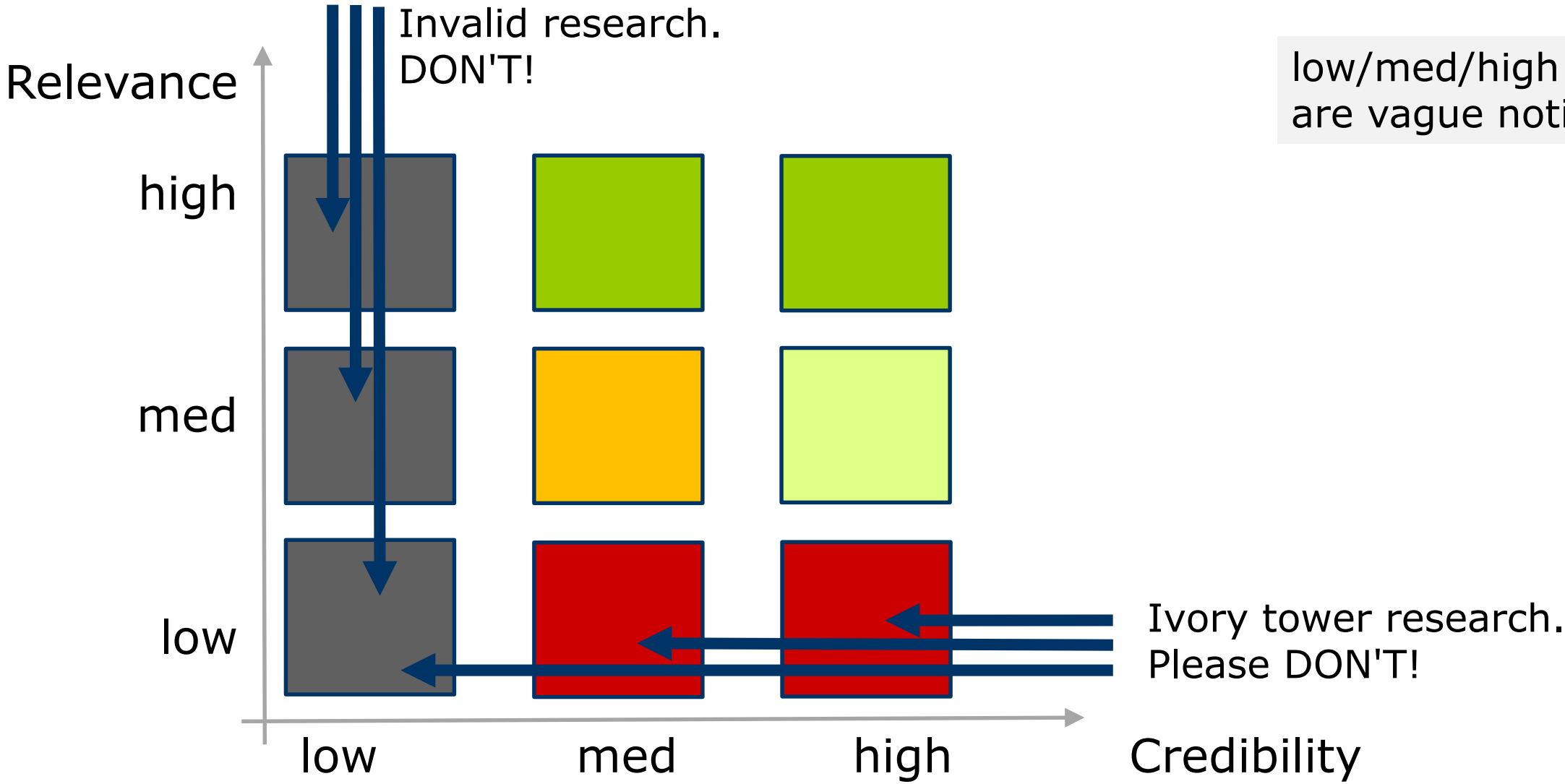
we'll get to these later

Where does relevance come from?

1. The target of the evaluation (i.e. the question asked) is of sufficient interest
   - rather than overly specialized or unimportant

2. The conclusions appear to apply to those situations where we want to apply them
   - "Ecological Validity"

Insist on sufficient credibility and relevance!



low/med/high
are vague notions

Qualitative vs. quantitative

- Empirical evaluations need not always be quantitative
  - i.e. counting and measuring something;
    providing numbers, graphs and calculations

- They can also be qualitative
  - characterizing non-quantifiable characteristics
  - characterizing contexts
  - explaining events and their consequences
  - providing subjective judgements obtained from relevant people

- or can combine both approaches
  - which is almost always a good idea for a quantitative study
    - but often not feasible for qualitative ones

Qualitative vs. quantitative: Examples

E.g. for applying a design method:

- Quantitative questions:
  - A. How long does it take?
    - time in minutes
  - B. How many mistakes are made in the process?
    - number of changes during work
  - C. How good is the result?
    - number of defects

- Corresponding qualitative questions:
  - A. What (types of) activities is the work time spent on?
  - B. Which kinds of mistake happen frequently? Why? How?
  - C. What are the typical kinds of flaws in the result?
    Why do they occur?
    How do they occur? What might be done to prevent them?

Method archetypes

Methods space is spanned by

- Research question nature:
  Howmuch? | Why? How?
- Situation wrt. repeatability:
  Humans | Machines
- Observations wrt. complexity:
  Numbers | Concepts

But not all 8 combinations occur:

4 Method archetypes:

- Quantitative [Numbers]
  - Experiments with groups of humans
    [Howmuch(+reason), Humans, Nums]
  - Repeatable experiments
    [Howmuch(+reason), Machines, Nums]
  - Fact-finding and correlation studies
    [Howmuch, X, Numbers]
- Qualitative [Concepts]
  - Sensemaking
    [Why/How, Humans, Concepts]

Common study type templates follow:

"archetype": a pattern of behavior, 'first' form, inherited pattern of thought

[archetype: repeatable experiments]
Study type "Automated tool benchmarking"

Freie Universität Berlin

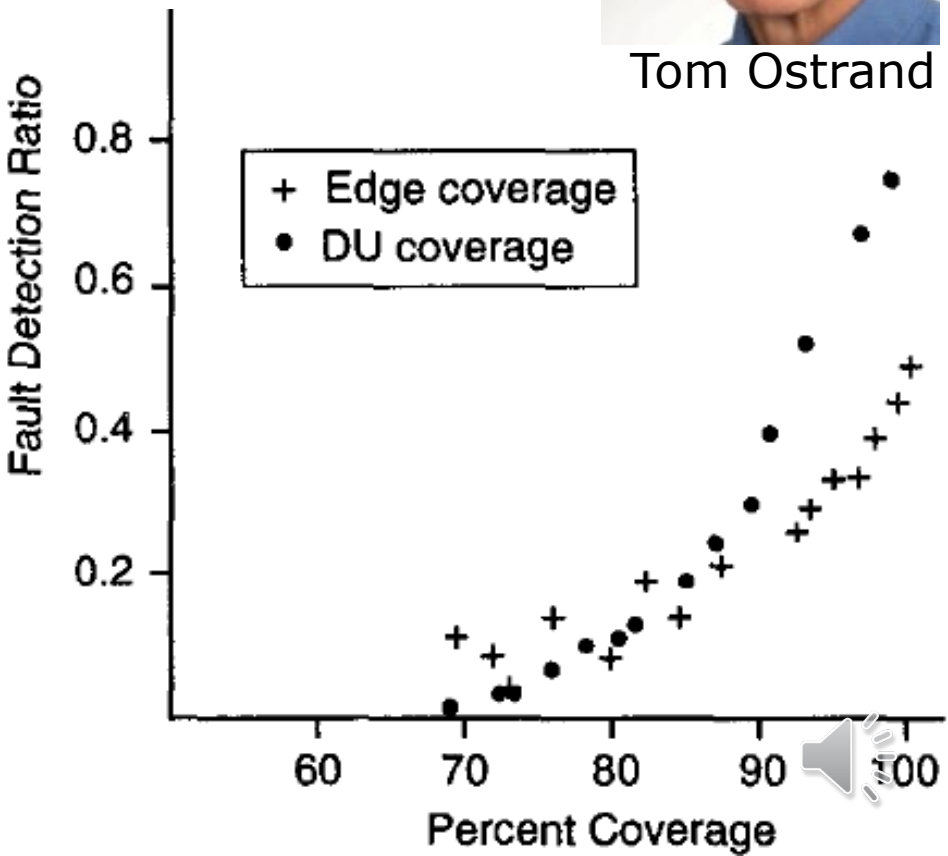
- When:
  - Validate effectiveness of an automated (analysis) tool [Howmuch?]
- What:
  - Collect a suitable corpus of objects; run tool;
    carefully judge each outcome [Machines, Numbers]

- Strengths:
  - Can use broad sets of inputs → Good generalizability [relevance]
  - Easy to understand for readers [credibility]
- Beware of:
  - Not discussing limits of applicability [credibility]
  - Misjudging your own judgment (missed false positives) [credibility]
  - Being optimistic about users' judgment skills [relevance]

Study type "Automated tool benchmarking" example: Comparing test coverage criteria

- M. Hutchins, H. Foster, T. Goradia, T. Ostrand: "Experiments of the effectiveness of dataflow- and controlflow-based test adequacy criteria", ICSE 1994
- Question:
  - How does the defect-detection effectiveness of all edges test coverage compare to all definition-use (DU) pairs?
- Method:
  - Select 7 correct C programs
  - Create and publish 130 single-fault versions of them and 7 thorough test pools (1k-5k tests)
  - Measure defect detection for 5000 test suites per program
- Result:

Tom Ostrand



(We will later discuss example studies in much more detail.)

Study type "Holistic field trial of tool or process"

- When:
  - Validate actual usefulness and usability of a tool or process element [How? Howmuch?]
- What:
  - Convince a team to use it; study their work before and after introduction; analyze effort, benefits, difficulties [Humans, Concepts, Machines, Numbers]

- Strengths:
  - Insights with lots of structure and detail [relevance]
  - Realistic, hence convincing [credibility]
- Beware of:
  - Difficult and lots of effort!
  - Too-idiosyncratic settings → lack of generalization [relevance]
  - Jumping to conclusions [credibility]
  - Impoverished quantitative-only variants of such studies [credibility]

Study type "Holistic field trial of tool or process" example: Web development in Java vs. Perl vs. PHP

- Prechelt: "Plat_Forms: A Web Development Platform Comparison by an Exploratory Experiment Searching for Emergent Platform properties", IEEE TSE 2011.
  - Question:
    What properties emerge from using each platform?
  - Method:
    3 teams-of-3 per platform all build the same web application
    in a two-day contest format (rapid prototyping). [atypical as a field trial]
  - Results:
    - Skill with a platform appears more important than the platform itself
    - Pragmatic ad-hoc approaches (Perl teams) helped maintainability more than standardized approaches (Java teams)
    - PHP teams showed the most uniform performance.
    - …

- When:
  - Measure attitudes and subjective appraisals regarding topic X [Howmuch?]
- What:
  - Interviews to find the relevant aspects of topic area [Humans, Concepts]; representative survey to measure distribution [Humans, Numbers]

- Strengths:
  - Can determine adequate questions and paint a realistic picture
  - Allows correlational analysis
- Beware of:
  - Self-selection bias
  - Ambiguous formulations
  - Respondent biases
  - Interpreting opinions as true statements of facts

Study type "Interviews + Survey" example: Properties of good SE managers

Eirini Kalliamvakou

- Eirini Kalliamvakou et al.:
  "What Makes a Great Manager of Software Engineers?", IEEE TSE 2019.
  - Question: What properties should an SE manager have to be perceived as good?
  - Method:
    37 interviews with SW engineers and managers,
    then questionnaire survey to validate and quantify the results
  - Results:
    is available, is technical, enables autonomy, supports experimentation, grows talent, promotes fairness, builds a relationship, recognizes individuality, clears path to execution, builds team culture, guides the team, maintains positive work environment, inspires, …

Freie Universität Berlin

- When:
  - To understand a relevant SW development process phenomenon [Why? How?]
- What:
  - Collect diverse types of data in the field (not only interviews!); perform sensemaking [Humans, Concepts]

- Strengths:
  - Statements grounded in specific instances → strong credibility [credibility]
  - Captures phenomena that exist → strong generality [relevance]
  - Provides better mental models for research and practice [relevance]
- Beware of:
  - Jumping to conclusions [credibility]
  - Risky: Takes looong, but it's unclear how interesting the results will be

Study type "Open process investigation" example: Quality experience

Freie Universität Berlin

- Lutz Prechelt, Holger Schmeisky, Franz Zieris:
  "Quality Experience: A Grounded Theory of Successful Agile Projects Without Dedicated Testers",
  Int'l. Conf. on SW Engineering 2016
  - Question:
    How come there are successful teams with and without dedicated testers?
  - Method:
    Grounded Theory Methodology; field observations and interviews in 3 agile teams, t1 with testers, t2&t3 without (t1&t2 from same organization)
  - Result:
    Finds that, in suitable application domains,
    quality assurance based on frequent deployments
    (rather than extensive manual testing)
    has many benefits
    - but requires a highly modular architecture and trust from the organization.
    - An example of successful DevOps transformation.

Holger Schmeisky

Other

- Correlational studies of other sorts can be helpful as well [Howmuch?]
  - Mining software repositories
  - Special-purpose process metrics

- Meta-Scientific studies can be helpful as well [Why? How?]
  - Systematic Literature Reviews [X, Concepts/Numbers]
  - Credibility criticism studies [Concepts]
  - Relevance criticism studies [Concepts]

- and certainly more…

Sharp turn ahead!

- Qualitative and quantitative methods tend to have fundamentally different worldviews
  - different "epistemological stance"
- As a result, the communities using each kind of methods tend to have different values
  - even different "culture"
- Not many people work in both camps

Let us define these terms:

Epistemological stance

Very simplified view:

- Epistemology:
  - The philosophical subject of "What can we know?" and "How do we know?"
    - Or: "What is knowledge?" "How do we obtain knowledge?"
- Epistemological stance:
  - My stance is described by the answer I give to the two questions

- There are many epistemological schools of thought:
  - e.g. aitihya/smṛti, Bayesian Epistemology, Constructivism, Critical Rationalism, Empiricism, Fallibilism, Idealism, Interpretivism, Perspectivism, Positivism, Postpositivism, Pragmatism, Rationalism, Relativism, Skepticism, …
  - Not all strictly formulate stances in the above sense
  - The stances and schools overlap greatly
    - but emphasize different things.
    - Most of the names have multiple meanings

This course overall takes a Pragmatist view

https://en.wikipedia.org/wiki/Epistemology
https://plato.stanford.edu/entries/epistemology/

Two epistemological stances (much simplified for V-EMPIR purposes)

Freie Universität Berlin

Positivist stance (V-EMPIR version):

- There is a single, fixed, objective, knowable reality
- A modest number of factors is responsible for what happens
  - reductionist perspective
- These factors can be measured objectively
- Their relationships can be fully understood (laws)

Interpretivist stance (V-EMPIR version):

- Reality is complex and requires interpretation
- A myriad factors are involved in what emerges
  - an emergence perspective
- Many cannot be measured objectively (but characterized *intersubjectively*)
- Many relationships are unique, accidental, ephemeral; understanding will be incomplete

The positivist stance is useful for what computers do, but ridiculous for what people or teams do.

These different stances lead to relevantly different cultures:

Definition "Culture" [ShwBel15]

- "Explicit and implicit patterns of behavior"
  - that constitute achievements of a human group and
- "traditional ideas and […] their attached values".
  - "Conventional understandings manifest in act and artifact."
- **"'culture' refers to community-specific ideas about what is true, good, beautiful, and efficient."**

Discussion:

- Each culture pays much attention to some things, much less to other things.
  - This can lead to blind spots.
  - This can lead to less-than-ideal empirical studies:
    - Positivist-inclined people tend to favor quantitative empirical methods
    - Interpretivist-inclined people tend to favor qualitative empirical methods
    - Few people do both types of studies.
- Keep this in mind when looking for problems with credibility and relevance.

Culture is a valuable *theoretical lens*

Summary

- SW engineering strives for usefulness, but is much more socio-technical than other fields of engineering
  - To obtain usefulness, empiricism is required in engineering and engineering research
  - A good empirical study is credible and relevant.
- An empirical method is a template for one type of approach to empirical work
  - There are 4 method archetypes
    - and many studies patterns (that often combine more than one method)
  - The most striking difference is between quantitative and qualitative methods
  - Underlying are different epistemological stance and different culture
- How we will proceed:
  1. Understand what threatens credibility and relevance
     - next lesson
  2. Look at several methods by example
     - remainder of the course

Thank you!

https://xkcd.com/1095/