CSC2130: Empirical Research Methods for Software Engineering

Steve Easterbrook

sme@cs.toronto.edu

www.cs.toronto.edu/~sme/CSC2130/

Course Goals

- Prepare students for advanced research in SE:
  - Learn how to plan, conduct and report on empirical investigations.
  - Understand the key steps of a research project:
  - formulating research questions, theory building, data analysis (using both qualitative and quantitative methods), building evidence, assessing validity, and publishing.
- Motivate the need for an empirical basis for SE
- Cover all principal empirical methods applicable to SE:
  - controlled experiment, case studies, surveys, archival analysis, action research, ethnographies,…
- Relate these methods to relevant metatheories in the philosophy and sociology of science.

Intended Audience

- This is an advanced software engineering course:
  - assumes a strong grasp of the key ideas of software engineering and the common methods used in software practice.
- Focus:
  - how do software developers work?
  - how do new tools and techniques affect their ability to construct high quality software efficiently?
  - qualitative and quantitative techniques from behavioural sciences
- The course is aimed at students who:
  - …plan to conduct SE research that demands some form of empirical validation
  - …wish to establish an empirical basis for an existing SE research programme
  - …wish to apply these techniques in related fields (e.g. HCI, Cog Sci)
- Note: we will *not* cover the kinds of experimental techniques used in CS systems areas.

Format

- Seminars:
  - 1 three-hour seminar per week
  - Mix of discussion, lecture, student presentations
- Readings
  - Major component is discussion of weekly readings
  - Please read the set papers before the seminar
- Assessment:
  - 10% Class Participation
  - 20% Oral Presentation - critique a published empirical study
  - 70% Written paper - design an empirical study for a SE research question

Course Outline

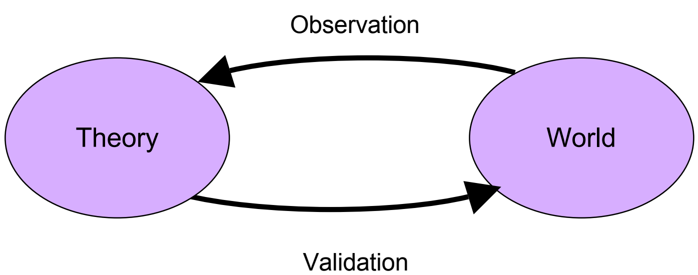
1. **Introduction & Orientation**
2. **What is Science?**
   - Philosophy of Science
   - Sociology of Science
   - Metatheories
3. **What is software engineering?**
   - Engineering & Design
   - Disciplinary Analogies for SE
   - Evidence-based software engineering
4. **Basics of Doing Research**
   - Finding good research questions
   - Theory building
   - Research Design
   - Ethics
   - Evidence and Measurement
   - Sampling
   - Peer Review Process

Course Outline (cont)

5. **Experiments**
   - Controlled Experiments
   - Quasi-experiments
   - Replication
6. **Case Studies**
   - Single and Multi-case
   - Longitudinal Case Studies
   - Approaches to Data Collection
7. **Reading Week -No seminar**
8. **Histories and Simulations**
   - Artifact Analysis
   - Archival Analysis and Post-mortems
   - Simulation Techniques
9. **Survey and Observation**
   - Surveys
   - Focus Groups
   - Field Studies / Ethnographies
10. **Interventions**
    - Action Research
    - Pilot Studies
    - Benchmarking
11. **Analysis Methods**
    - Qualitative, Quantitative and Mixed approaches
    - Statistical Analysis
    - Grounded Theory
12. **Generalisation and Validity**
    - Threats to Validity
    - Power and Reliability
    - Replication
13. **Reporting and Publishing**
    - Displaying data
    - Writing up results
    - Where to publish

A Scientific Approach (?)

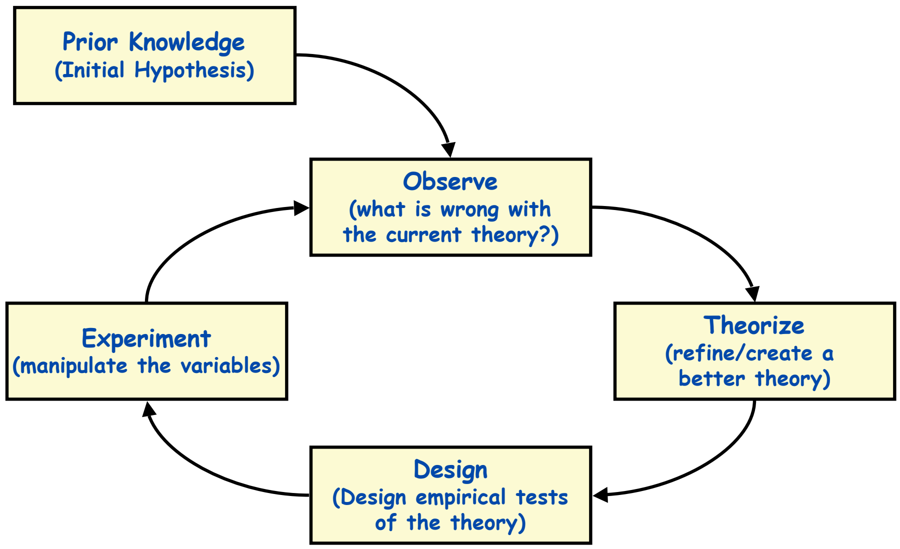
- No single "official" scientific method
  http://dharma-haven.org/science/myth-of-scientific-method.htm
- However, there are commonalities



High School Science Version

1. Observe some aspect of the universe.
2. Invent a tentative description, called a *hypothesis*, that is consistent with what you have observed.
3. Use the hypothesis to make predictions.
4. Test those predictions by experiments or further observations and modify the hypothesis in the light of your results.
5. Repeat steps 3 and 4 until there are no discrepancies between theory and experiment and/or observation.

Inquiry Cycle



Some Characteristics of Science

- **Explanations are based on observations**
  - A way of thinking
  - Relationships are perceptible in a way that has to make sense given accepted truths
- **Creativity is as important as in art**
  - Hypotheses, experimental designs
  - Search for elegance, simplicity

Some Definitions

- A model is an abstract representation of a phenomenon or set of related phenomena
  - Some details included, others excluded
- A theory is a set of statements that explain a set of phenomena
  - Serves to explain and predict
- A hypothesis is a testable statement derived from a theory
  - A hypothesis is not a theory!
- In software engineering, there are few capital-T theories
  - Many small-t theories, philosophers call these *folk theories*

Science and Theory

- A (scientific) theory is:
  - more than just a description - it explains and predicts
  - Logically complete, internally consistent, falsifiable
  - Simple and elegant.
- Components of a theory:
  - concepts, relationships, causal inferences
    - E.g. Conway's Law- structure of software reflects the structure of the team that builds it. A theory should explain why.
- Theories lie at the heart of what it means to do science.
  - Production of generalizable knowledge
  - Scientific method ⇔ Research Methodology ⇔ Proper Contributions for a Discipline
- Theory provides orientation for data collection
  - Cannot observe the world without a theoretical perspective

Meta-Theories

- Logical Positivism:
  - Separates discovery from validation
  - Logical deduction, to link theoretical concepts to observable phenomena
  - Scientific truth is absolute, cumulative, and unifiable
- Popper:
  - Theories can be refuted, not proved;
  - only falsifiable theories are scientific
- Campbell:
  - Theories are underdetermined;
  - All observation is theory-laden & biased
- Quine:
  - Terms used in scientific theories have contingent meanings
  - Cannot separate theoretical terms from empirical findings
- Kuhn:
  - Science characterized by dominant paradigms, punctuated by revolution
- Lakatos:
  - Not one paradigm, but many competing research programmes
  - Each has a hard core of assumptions immune to refutation
- Feyerabend:
  - Cannot separate scientific discovery from its historical context
  - All scientific methods are limited;
  - Any method offering new insight is okay
- Toulmin:
  - Evolving Weltanschauung determines what is counted as fact;
  - Scientific theories describe ideals, and explain deviations
- Laudan:
  - Negative evidence is not so significant in evaluating theories.
  - All theories have empirical difficulties
  - New theories seldom explain everything the previous theory did

What is a research contribution?

- A better understanding of how software engineers work?
- Identification of problems with the current state-of-the-art?
- A characterization of the properties of new tools/techniques?
- Evidence that approach A is better than approach B?

How will you validate your claims?

Meet Stuart Dent

- **Name:**
  - Stuart Dent (a.k.a. "Stu")
- **Advisor:**
  - Prof. Helen Back
- **Topic:**
  - Merging Stakeholder views in Model Driven Development
- **Status:**
  - 2 years into his PhD
  - Has built a tool
  - Needs an evaluation plan

Stu's Evaluation Plan

- **Formal Experiment**
  - Independent Variable: Stu-Merge vs. Rational Architect
  - Dependent Variables: Correctness, Speed, Subjective Assessment
  - Task: Merging Class Diagrams from two different stakeholders' models
  - Subjects: Grad Students in SE
  - H_1: "Stu-Merge produces correct merges more often than RA"
  - H_2: "Subjects produce merges faster with Stu-Merge than with RA"
  - H_3: "Subjects prefer using Stu-Merge to RA"
- **Results**
  - H_1 accepted (strong evidence)
  - H_2 & H_3 rejected
  - Subjects found the tool unintuitive

Threats to Validity

- Construct Validity
  - What do we mean by a merge? What is correctness?
  - 5-point scale for subjective assessment - insufficient discriminatory power
    - (both tools scored very low)
- Internal Validity
  - Confounding variables: Time taken to learn the tool; familiarity
  - Subjects were all familiar with RA, not with Stu-merge
- External Validity
  - Task representativeness: class models were of a toy problem
  - Subject representativeness: Grad students as sample of what population?
- Theoretical Reliability
  - Researcher bias: subjects knew Stu-merge was Stu's own tool

More on validity in the backup slides at the end of the talk

What went wrong?

- What was the research question?
  - "Is tool A better than tool B?"
- What would count as an answer?
- What use would the answer be?
  - How is it a "contribution to knowledge"?
- How does this evaluation relate to the existing literature?

Experiments as Clinical Trials

Why would we expect it to be better?

Why do we need to know?

What will we do with the answer?

Is drug A better than drug B?

Better at doing what?

Better in what way?

Better in what situations?

Why would we expect it to be better?

You gotta have a theory!

The Role of Theory Building

- Theories allow us to compare similar work
  - Theories include precise definition for the key terms
  - Theories provide a rationale for which phenomena to measure
- Theories support analytical generalization
  - Provide a deeper understanding of our empirical results
  - …and hence how they apply more generally
  - Much more powerful than statistical generalization
- …but in SE we are very bad at stating our theories
  - Our vague principles, guidelines, best practices, etc. could be strengthened into theories
  - Every tool we build represents a theory

Stu's Theory

- Background Assumptions
  - Large team projects, models contributed by many actors
  - Models are fragmentary, capture partial views
  - Partial views are inconsistent and incomplete most of the time
- Basic Theory
  - (Brief summary:)
  - Model merging is an exploratory process, in which the aim is to discover intended relationships between views. 'Goodness' of a merge is a subjective judgment. If an attempted merge doesn't seem 'good', many need to change either of the models, or the way in which they were mapped together.
- Derived Hypotheses
  - Useful merge tools need to represent relationships explicitly
  - Useful merge tools need to be complete (work for any models, even if inconsistent)

What type of question are you asking?

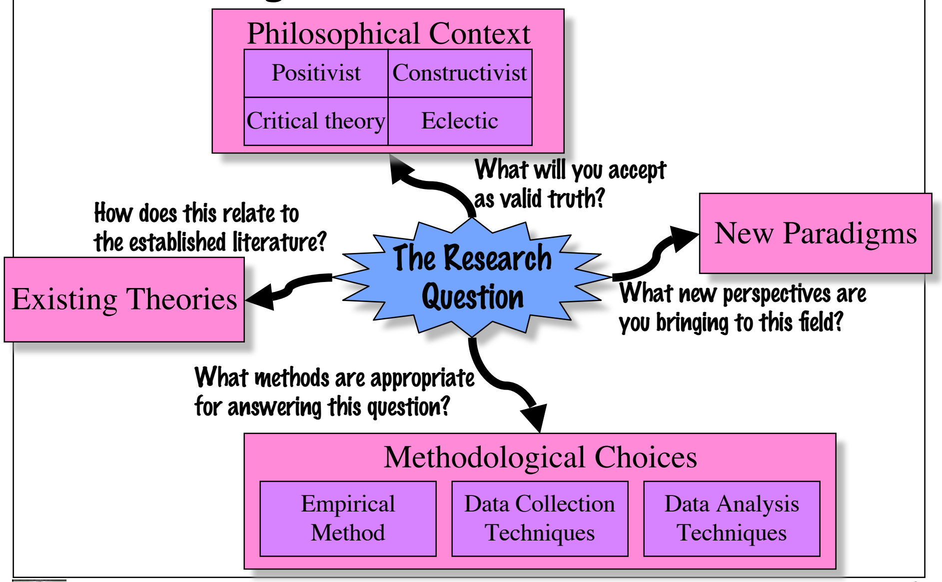
- Existence:
  - Does X exist?
- Description & Classification
  - What is X like?
  - What are its properties?
  - How can it be categorized?
  - How can we measure it?
  - What are its components?
- Descriptive-Process
  - How does X work?
  - What is the process by which X happens?
  - In what are the steps as X evolves?
  - How does X achieve its purpose?
- Descriptive-Comparative
  - How does X differ from Y?
- Relationship
  - Are X and Y related?
  - Do occurrences of X correlated with occurrences of Y?
- Causality
  - Does X cause Y?
  - Does X prevent Y?
  - What causes X?
  - What effect does X have on 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?

Stu's Research Question(s)

Pick just one for now...

- Existence
  - Does model merging ever happen in practice?
- Description/Classification
  - What are the different types of model merging that occur in practice on large scale systems?
- Descriptive-Comparative
  - How does model merging with explicit representation of relationships differ from model merging without such representation?
- Causality
  - Does an explicit representation of the relationship between models cause developers to explore different ways of merging models?
- Causality-Comparative
  - Does the algebraic representation of relationships in Stu's tool lead developers to explore more than do pointcuts in AOM?

Putting the Question in Context



How do we evaluate our tools?

Common “in the lab” Methods	Common “in the wild” Methods
→ Rational Reconstructions → Exemplars → Benchmarks → Simulations → Controlled Experiments	→ Case Studies → Surveys → Ethnographies → Artifact/Archive Analysis (“mining”!) → Action Research

Rational Reconstructions

a demonstration of a tool or technique on data taken from a real case study, but applied after the fact to demonstrate how the tool/technique would have worked

→ good for

initial validation before expensive pilot studies

checking the researcher's intuitions about what the tool/technique can do

→ limitations

potential bias (you knew the findings before you started)

easy to ignore "signal-to-noise ratio"

→ examples

In RE: LAS; BART; … etc.

See:

Shaw, M.; Writing good software engineering research papers. Proceedings. 25th International Conference on Software Engineering (ICSE 2003). p726-736

Exemplars

self-contained, informal descriptions of a problem in some application domain; exemplars are to be considered immutable; the [researcher] must do the best she can to produce a [solution] from the problem statement.

→ Good for:

Setting research goals,

Understanding differences between research programs

→ Limitations:

No clear criteria for comparing approaches

Not clear that "immutability" is respected in practice

→ Examples:

Meeting Scheduler; Library System; Elevator Control System; Telephones;…

see:

M. S. Feather, S. Fickas, A. Finkelstein, and A. van Lamsweerde, "Requirements and Specification Exemplars," Automated Software Engineering, vol. 4, pp. 419-438, 1997.

Benchmarks

A test or set of tests used to compare alternative tools or techniques. A benchmark comprises a motivating comparison, a task sample, and a set of performance measures

→ good for

- making detailed comparisons between methods/tools
- increasing the (scientific) maturity of a research community
- building consensus over the valid problems and approaches to them

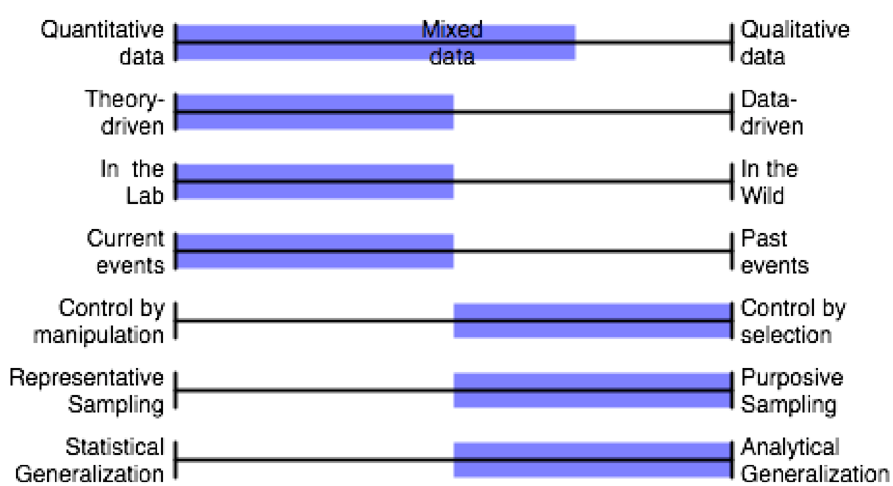
→ limitations

- can only be applied if the community is ready
- become less useful / redundant as the research paradigm evolves

See:

S. Sim, S. M. Easterbrook and R. C. Holt "Using Benchmarking to Advance Research: A Challenge to Software Engineering". Proceedings, ICSE-2003

Benchmarking



Simulations

An executable model of the software development process, developed from detailed data collected from past projects, used to test the effect of process innovations

→ Good for:

- Preliminary test of new approaches without risk of project failure
- [Once the model is built] each test is relatively cheap

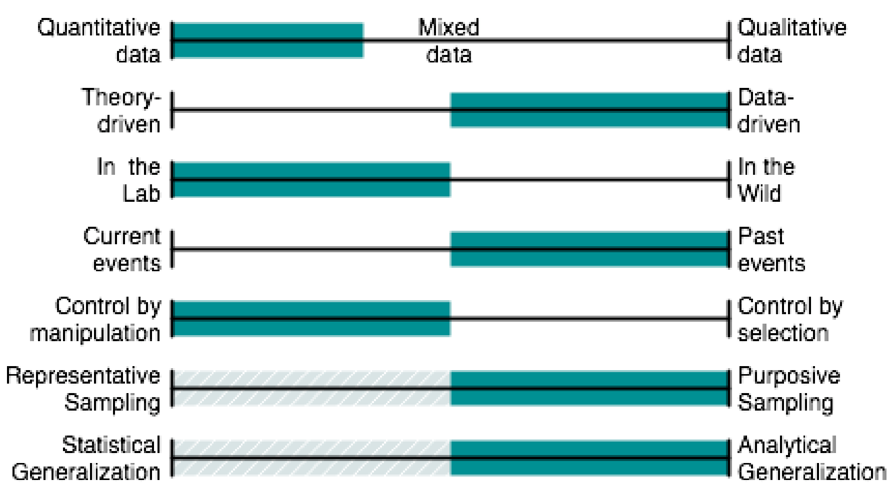
→ Limitations:

- Expensive to build and validate the simulation model
- Model is only as good as the data used to build it
- Hard to assess scope of applicability of the simulation

See:

Kellner, M. I.; Madachy, R. J.; Raffo, D. M.; Software Process Simulation Modeling: Why? What? How? Journal of Systems and Software 46 (2-3) 91-105, April 1999.

Simulations



Controlled Experiments

experimental investigation of a testable hypothesis, in which conditions are set up to isolate the variables of interest ("independent variables") and test how they affect certain measurable outcomes (the "dependent variables")

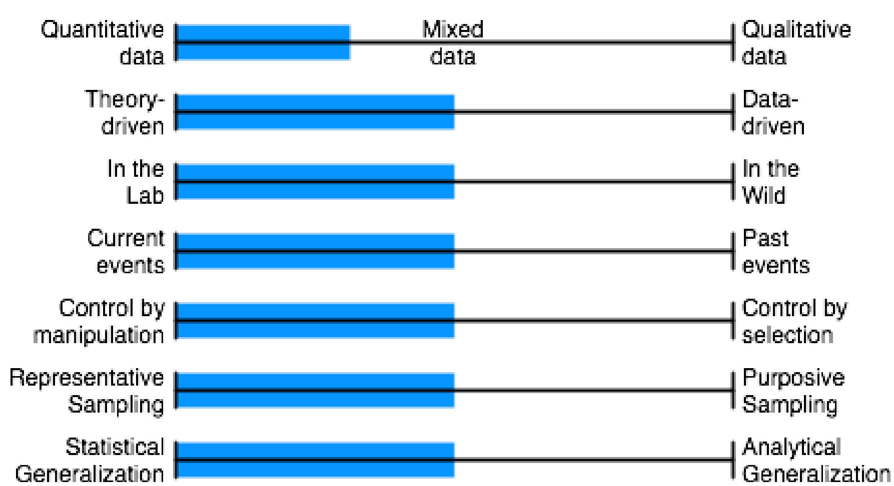
→ **good for**
- quantitative analysis of benefits of a particular tool/technique
- establishing cause-and-effect in a controlled setting
- (demonstrating how scientific we are!)

→ **limitations**
- hard to apply if you cannot simulate the right conditions in the lab
- limited confidence that the laboratory setup reflects the real situation
- ignores contextual factors (e.g. social/organizational/political factors)
- extremely time-consuming!

See:

Pfleeger, S.L.; Experimental design and analysis in software engineering. *Annals of Software Engineering* 1, 219-253. 1995

Controlled Experiments



Case Studies

"A technique for detailed exploratory investigations, both prospectively and retrospectively, that attempt to understand and explain phenomenon or test theories, using primarily qualitative analysis"

→ good for

- Answering detailed how and why questions
- Gaining deep insights into chains of cause and effect
- Testing theories in complex settings where there is little control over the variables

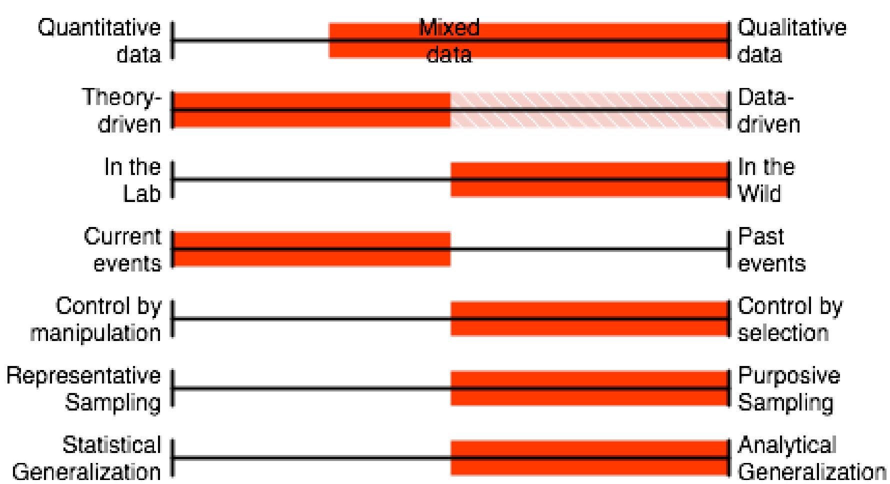
→ limitations

- Hard to find appropriate case studies
- Hard to quantify findings

See:

Flyvbjerg, B.; Five Misunderstandings about Case Study Research. Qualitative Inquiry 12 (2) 219-245, April 2006

Case Studies



Survey Research

"A comprehensive system for collecting information to describe, compare or explain knowledge, attitudes and behaviour over large populations"

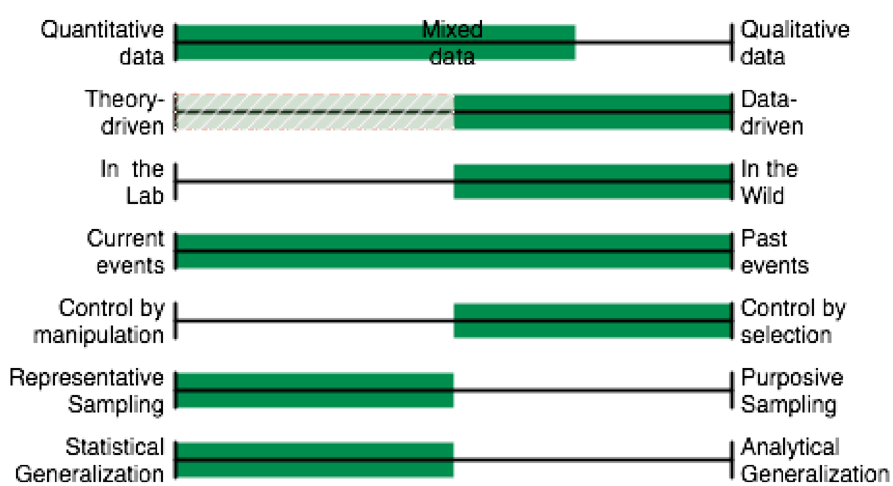
- **good for**
  - Investigating the nature of a large population
  - Testing theories where there is little control over the variables
- **limitations**
  - Relies on self-reported observations
  - Difficulties of sampling and self-selection
  - Information collected tends to subjective opinion

See:

Shari Lawarence Pfleeger and Barbara A. Kitchenham, "Principles of Survey Research," Software Engineering Notes, (6 parts) Nov 2001 - Mar 2003

University of Toronto
Department of Computer Science

Survey Research



Ethnographies

Interpretive, in-depth studies in which the researcher immerses herself in a social group under study to understand phenomena though the meanings that people assign to them

→ Good for:

- Understanding the intertwining of context and meaning
- Explaining cultures and practices around tool use
- Deep insights into how people perceive and act in social situations

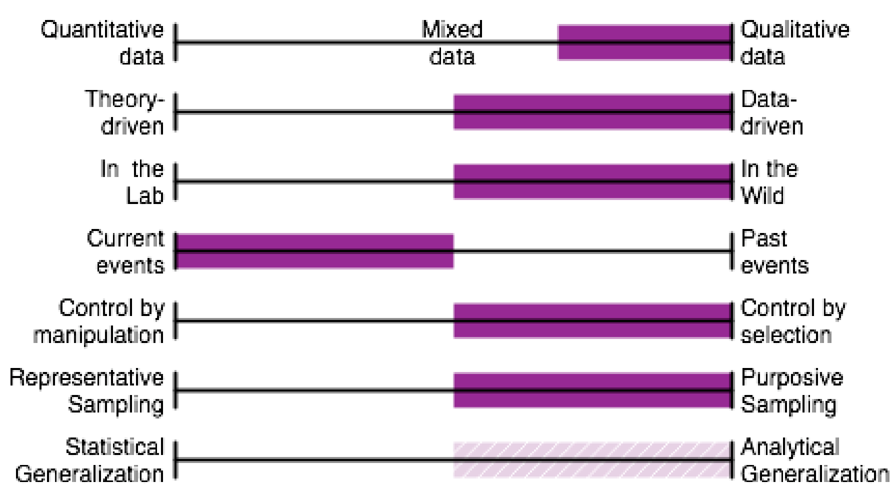
→ Limitations:

- No generalization, as context is critical
- Little support for theory building
- Expensive (labour-intensive)

See:

Klein, H. K.; Myers, M. D.; A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly 23(1) 67-93. March 1999.

Ethnographies



Artifact / Archive Analysis

Investigation of the artifacts (documentation, communication logs, etc) of a software development project after the fact, to identify patterns in the behaviour of the development team.

➜ good for

- Understanding what really happens in software projects
- Identifying problems for further research
- Collecting data to build or validate simulations

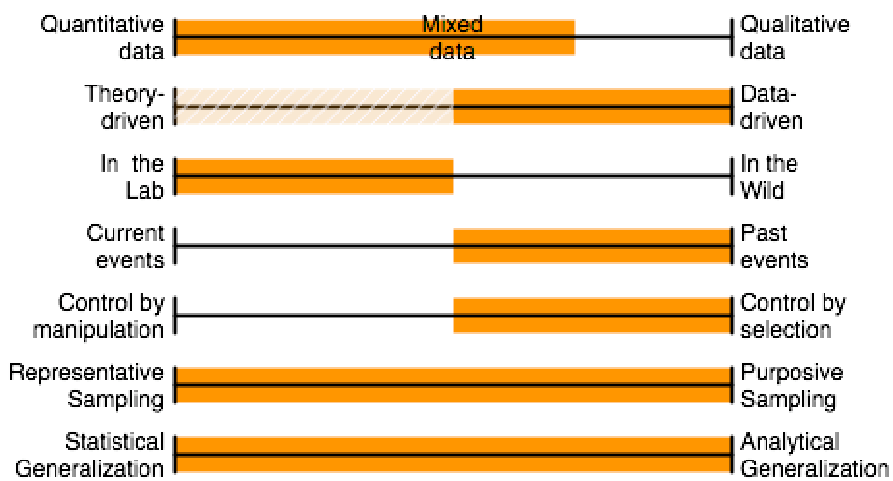
➜ limitations

- Hard to build generalizations (results may be project specific)
- Incomplete data
- Ethics: how to get consent from participants

See:

Audris Mockus, Roy T. Fielding, and James Herbsleb. Two case studies of open source software development: Apache and mozilla. ACM Transactions on Software Engineering and Methodology, 11(3):1-38, July 2002.

Artifact / Archive Analysis



Action Research

"research and practice intertwine and shape one another. The researcher mixes research and intervention and involves organizational members as participants in and shapers of the research objectives"

→ good for

- any domain where you cannot isolate {variables, cause from effect, …}
- ensuring research goals are relevant
- When effecting a change is as important as discovering new knowledge

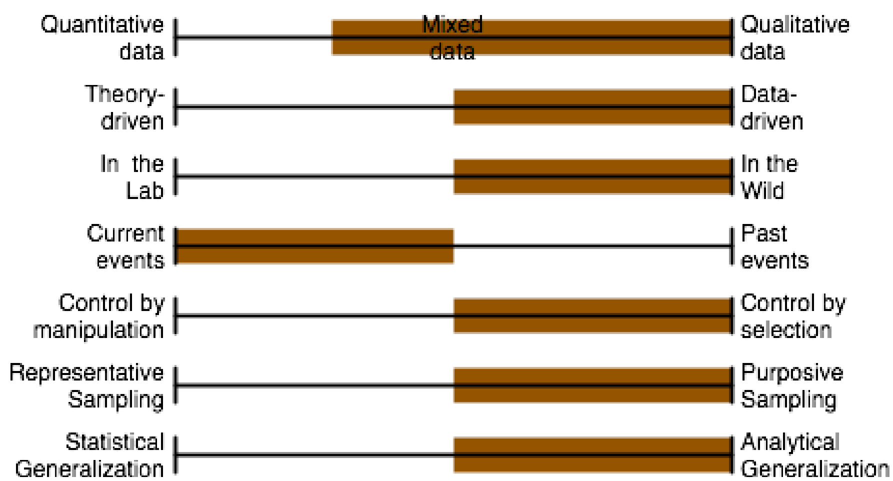
→ limitations

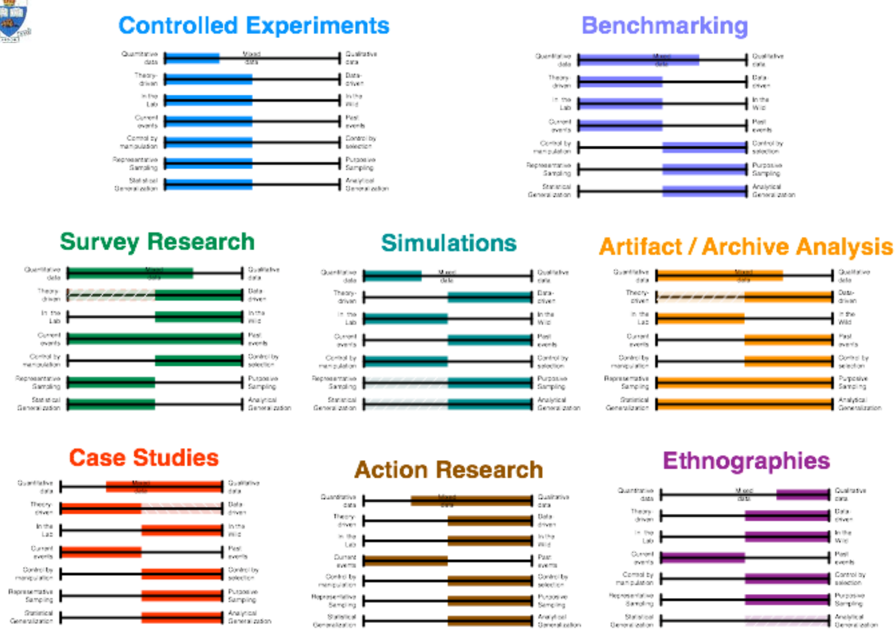
- hard to build generalizations (abstractionism vs. contextualism)
- Strongly tied to philosophy of critical theory - won't satisfy the positivists!

See:

Lau, F; Towards a framework for action research in information systems studies. Information Technology and People 12 (2) 148-175. 1999.

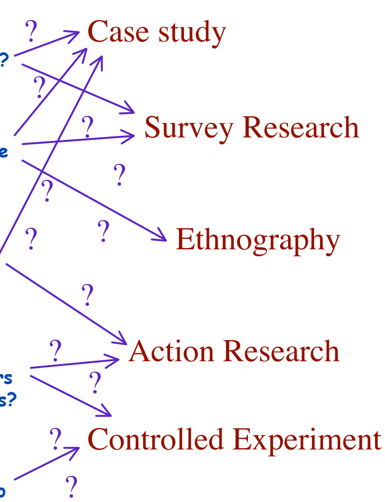
Action Research





Stu's Method(s) Selection…

- **Existence**
  - Does model merging ever happen in practice?
- **Description/Classification**
  - What are the different types of model merging that occur in practice on large scale systems?
- **Descriptive-Comparative**
  - How does model merging with explicit representation of relationships differ from model merging without such representation?
- **Causality**
  - Does an explicit representation of the relationship between models cause developers to explore different ways of merging models?
- **Causality-Comparative**
  - Does the algebraic representation of relationships in Stu's tool lead developers to explore more than do pointcuts in AOM?



Warning

No method is perfect

Don't get hung up on methodological purity

Pick something and get on with it

Some knowledge is better than none

Okay, but…

Why Build a Tool?

- Build a Tool to Test a Theory
  - Tool is part of the experimental materials needed to conduct your study
- Build a Tool to Develop a Theory
  - Theory emerges as you explore the tool
- Build a Tool to Explain your Theory
  - Theory as a concrete instantiation of (some aspect of) the theory

Take home messages

Articulate the theory(s) underlying your work

Be precise about your research question

Be explicit about your philosophical stance

Use the theory to guide the study design

Test the Theory not the Tool