A practical guide to controlled experiments of software engineering tools with human participants

Andrew J. Ko, Thomas D. LaToza, Margaret M. Burnett

ESE Feb 2015

Summary by Prof. Thomas LaToza

SWE 795, Spring 2017

Software Engineering Environments

Motivation

- Evaluate the **usability** of a programming language feature or tool for developers
  - usually productivity effects
- Given a context, what is effect on developer productivity

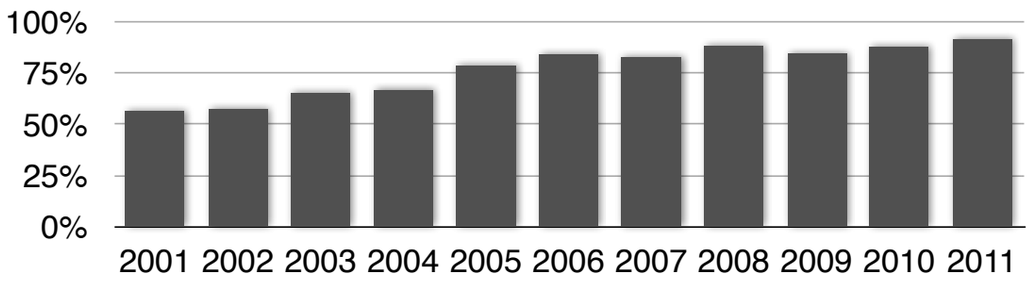
Challenges

- How many **participants** do I need?
- Which participants to recruit?
- What do I **measure**? How do I measure it?
- Should I train participants?
- What **tasks** should I pick?

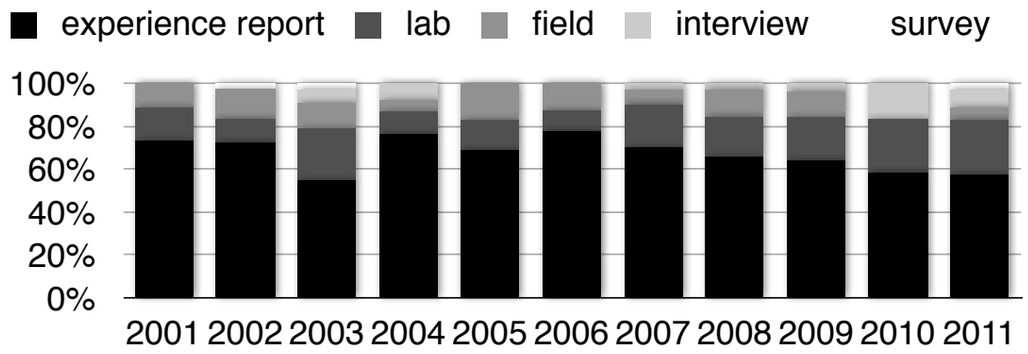
Evaluations of software engineering tools w/ humans are rare

- Systematic review of 1701 software engineering articles
  - All papers published at ICSE, FSE, TSE, TOSEM 2001 - 2011

82%	63%	17%
1392	1065	289
described tool	empirical eval	empirical eval w/ humans



% of papers w/ tools w/ empirical evals



% of human evals by method

Controlled experiment

- Only way to argue **causality** - change in var x causes change in var y

- Manipulate **independent** variables
  Creates “conditions” that are being compared
  Can have >1, but # conditions usually exponential in # ind. variables

- Measure dependent variables (a.k.a measures)
  Quantitative variable you calculate from collected data
  E.g., time, # questions, # steps, ...

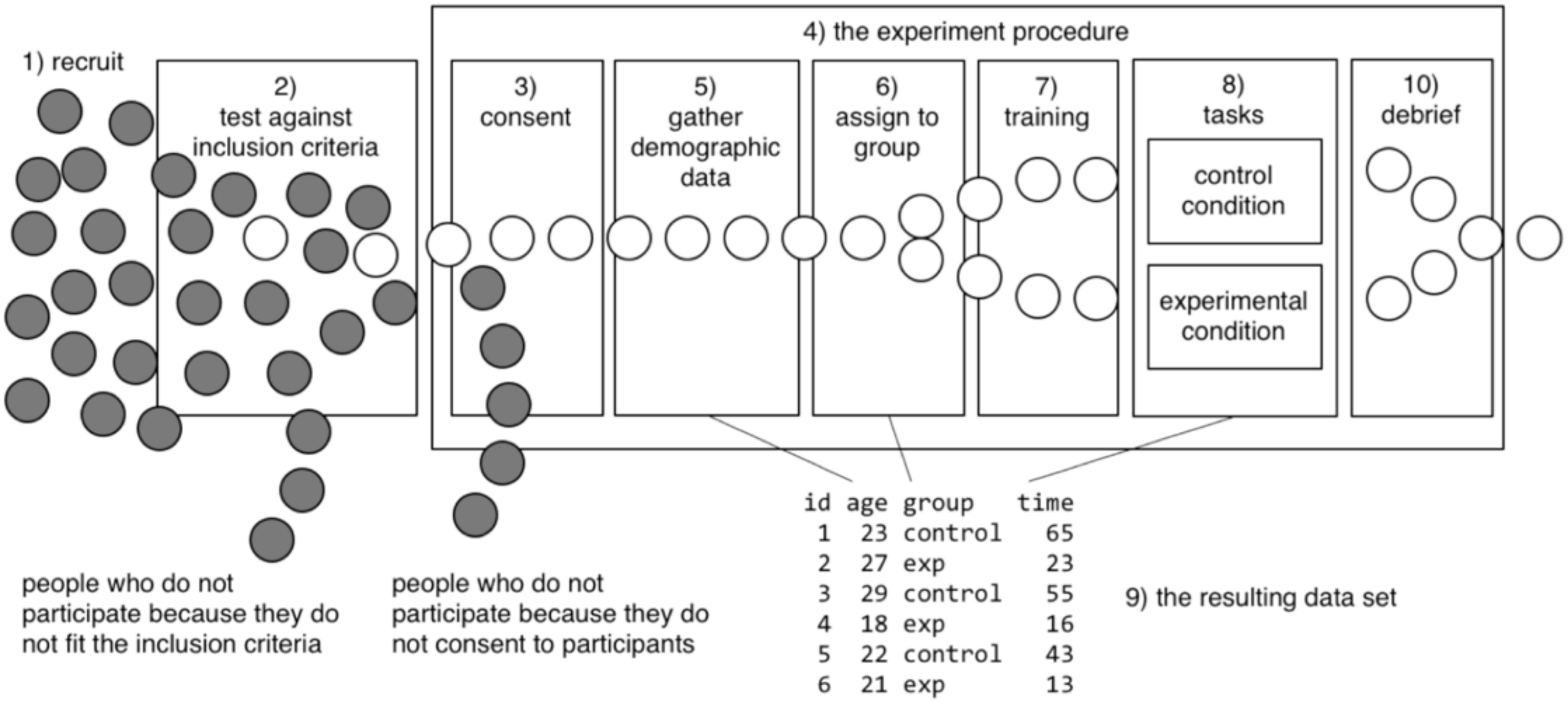
- **Randomly** assign participants to condition
  Ensure that participants only differ in condition
  Not different in other **confounding** variables

- Test hypotheses
  Change in independent variable causes dependent variable change
  e.g., t-test, ANOVA, other statistical techniques

Terminology

- "Tool" — any **intervention** manipulating a software developer's work environment
  - e.g., programming language, programming language feature, software development environment feature, build system tool, API design, documentation technique, …
- Data — what you collected in study
- Unit of analysis — individual **item** of data
- Population — **all** members that exist
- Construct — some **property** about member
- Measure — **approximation** of construct computed from data

Anatomy of controlled experiment w/ humans



Deciding who to recruit

- **Inclusion criterion**: attributes participants must have to be included in study
- Goal: reflect characteristics of those that researchers believe would benefit
- Example - Nimmer & Ernst (2002)
  - Support those w/ out experience w/ related analysis tools
  - Chose graduate students
  - Developed items to assess (1) did not have familiarity w/ tool (2) Java experience (3) experience writing code

Tasks

- Goal: design tasks that have **coverage** of work affected by tool

- Key tradeoff: realism vs. control
  - How are real, messy programming tasks **distilled** into brief, accessible, actionable activities?

- More realism —> messier, fewer controls

- More control —> cleaner, less realism

- Tradeoff often takes the form of tradeoff between bigger tasks vs. smaller tasks

Discussion Questions

- Overall reaction to the paper
- What aspect of evaluating tools was the most confusing?
- What aspect seems the most challenging?
- When (if ever) is a controlled experiment the wrong evaluation for a tool?
- How much evaluation is enough?