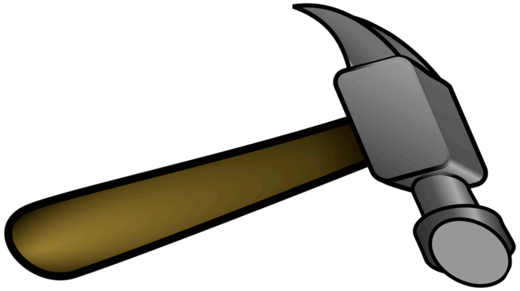
$\int (2x$



Software Architecture Research: Science or Engineering?

Philippe Kruchten

Seattle, May 1st, 2018

ICSA YRF

Philippe Kruchten, Ph.D., P.Eng., FEC, CSDP

Professor of Software Engineering
NSERC Chair in Design Engineering
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC Canada
pbk@ece.ubc.ca

Founder and president
Kruchten Engineering Services Ltd
Vancouver, BC Canada
philippe@kruchten.com

Career path

- 1970: learned Fortran IV
- 1972-1975: developed 2 large apps -> bought a car
  (1975: mechanical engineering degree)
- 1983: very 1st Ada compiler (New York University)
- 1984: System architect at Alcatel Business Systems (Télic)
  (1986: PhD in information systems)
- 1987: joined Rational Software as a consultant in Software Architecture
- 1990: developed Rational architectural method
- 1992-1996: Lead software architect for the Canadian ATC
- 1996-2003: Led development of the Rational Unified Process (aka RUP)
- 2004-now: Professor at UBC, Vancouver, Canada

Caveat - Disclaimer

- This talk is not science

- This talk is not engineering either.

- This is just a bunch of opinions, rooted in my own experience.

Defining success

- So you are working on a PhD…
- … in software architecture (or something softwarish like that)

Defining success

- So you are working on a PhD…
- … in software architecture (or something softwarish like that)

- Why?

Defining success

- So you are working on a PhD…
- … in software architecture (or something softwarish like that)

- Why?
- How do you define success?

- If you met *success* what would it look like?

Defining success

- Pass
- Degree
- Job
- Citation #
- H-Index
- New venture: commercialization
- $$$$$$?
- …
- Impact on our profession:

  # of users

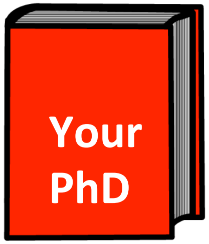
  /= citation count

Evaluation & Judgment

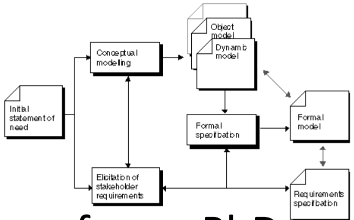
- Conceptual frameworks
  - Pre-existing
  - Home-grown

- Filter
  - For judgment
  - For memorization
  - *For evolution of the framework (self-reflection)*

My own PhD filter



- How much science and how much engineering?
- How valuable is it? (impact)
- How valid is it?
- Is it well communicated?


Initial statement of need
Conceptual modeling
Object model
Dynamic model
Elicitation of stakeholder requirements
Formal specification
Formal model
Requirements specification

My image of your PhD + evaluation

Aside: under the hood: my Filter "engine"

1. Abstract
2. Introduction (… motivation, concept definition)
3. Conclusion (claims)
4. Method(ology)
5. Body, seeing how it support claims
6. References (only as needed)
7. Validation
8. Related work
9. Abstract (2) & Title
10. Conclusion (2)
11. References (all, completeness)

Scientist

Engineer

Software Architect

Researcher in Software Architecture

On the science side

- A better understanding of a phenomenon
  - Model
  - Conceptual, mathematical, ontological, ….
  - Theory

On the engineering side

- Build a tool
- A mousetrap
- A better mousetrap
- Immediate value to mankind,
  - or at least the software practitioner subset of mankind

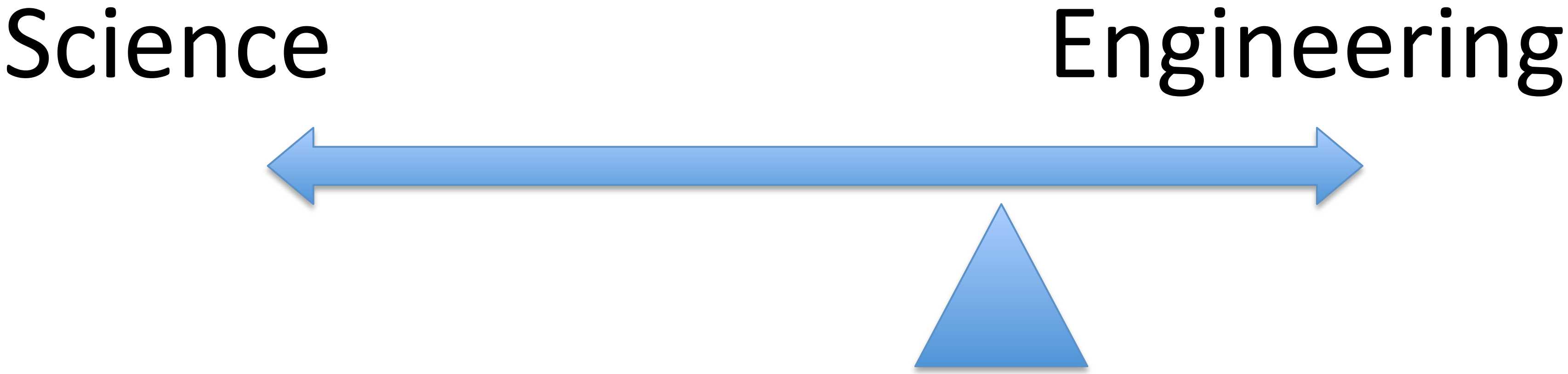
Bad science

- Not a problem
- Overgeneralization
- Not rooted in evidence

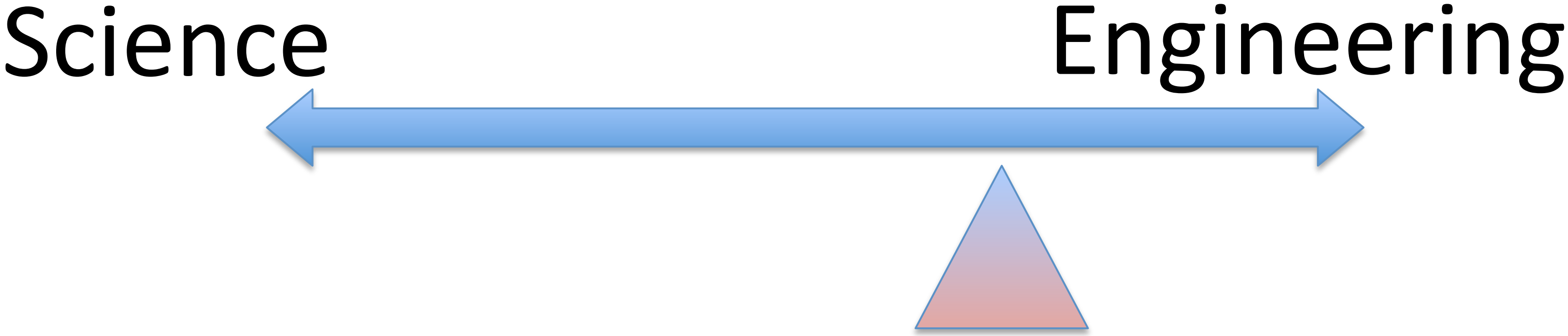
Bad engineering

- Yes another mousetrap
- Does not solve a problem anyone has (anymore)
- Does not scale to real-life problem
- ….

Balance ?



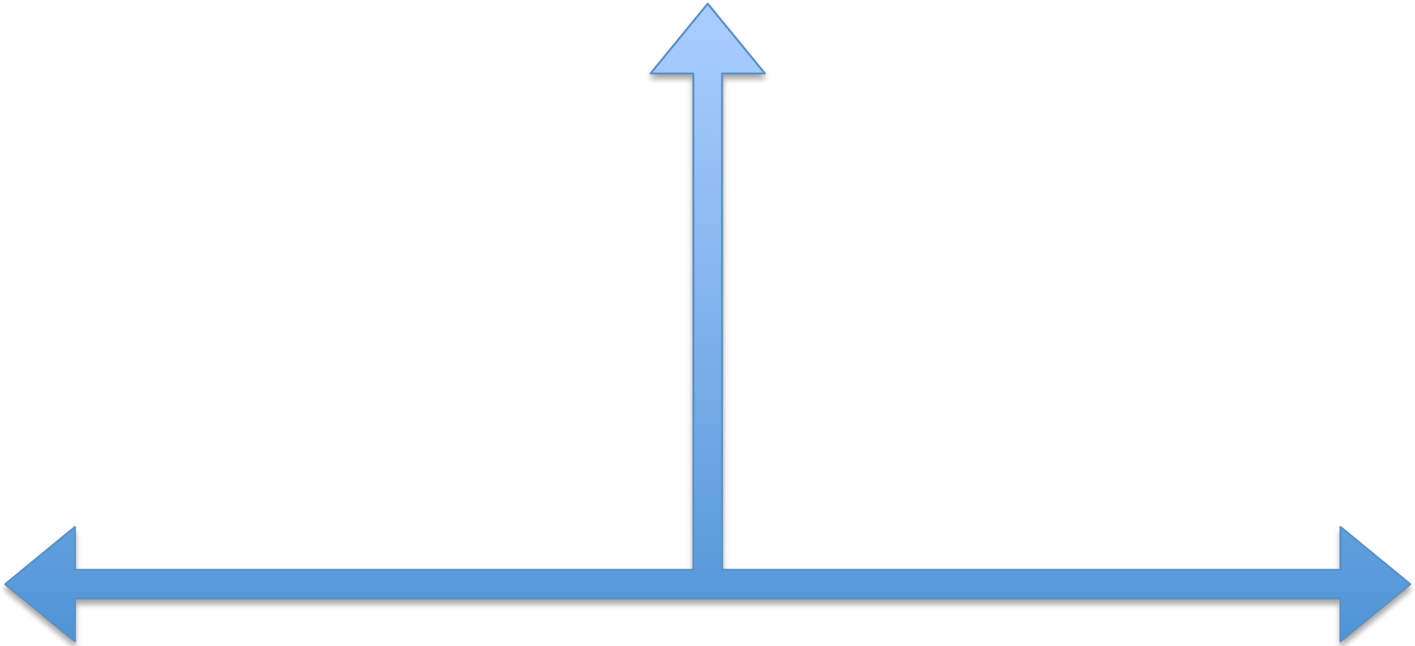
Where do you locate yourself?



Science

Engineering

Impact



Science

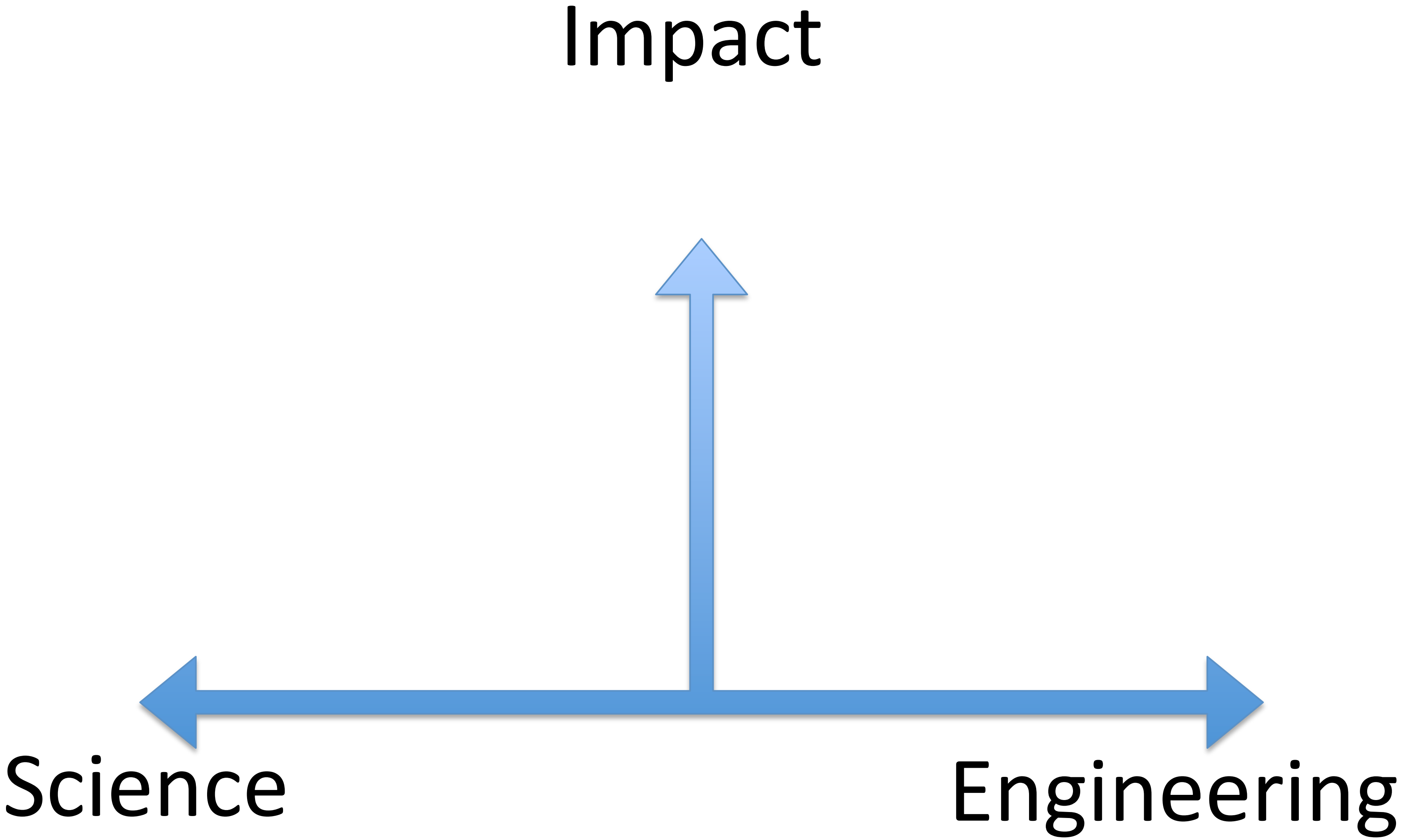
Engineering

Defining success

- Pass
- Degree
- Job
- Citation #
- H-Index
- New venture: commercialization
- $$$$$$?
- …
- Impact on our profession:

  # of users

  /= citation count

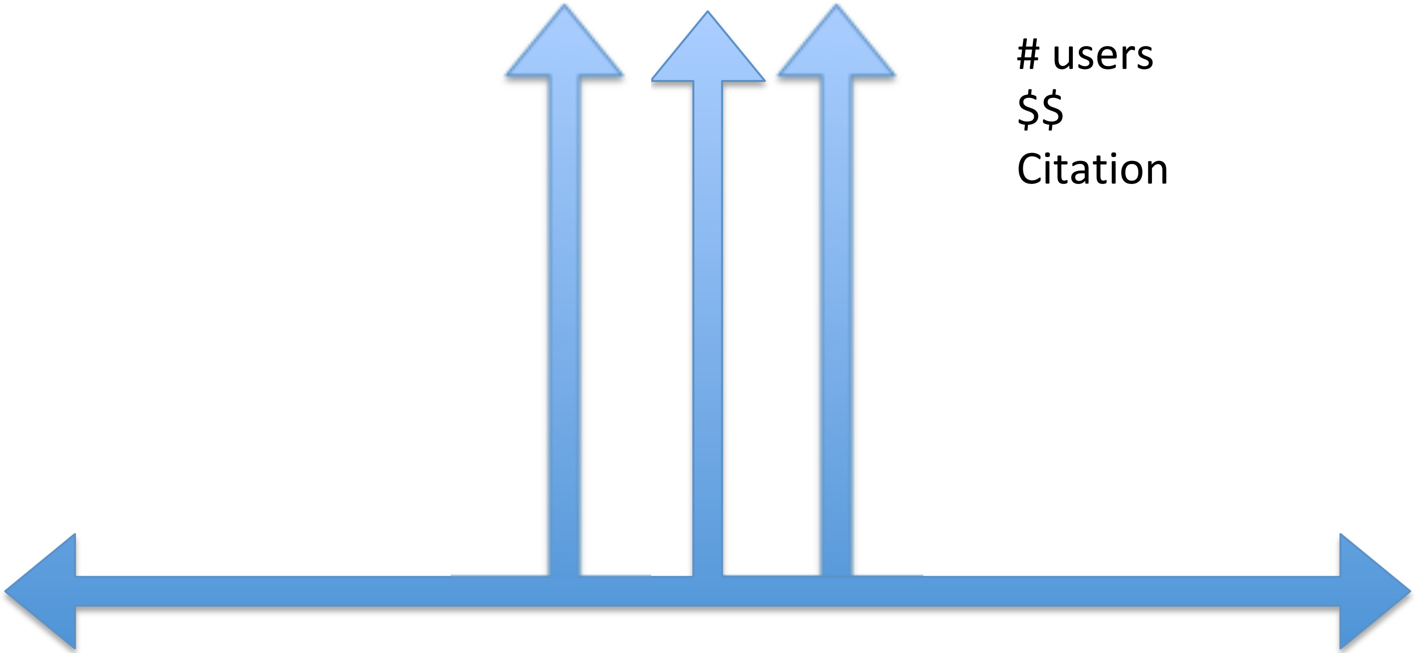


Impact

Science

Engineering

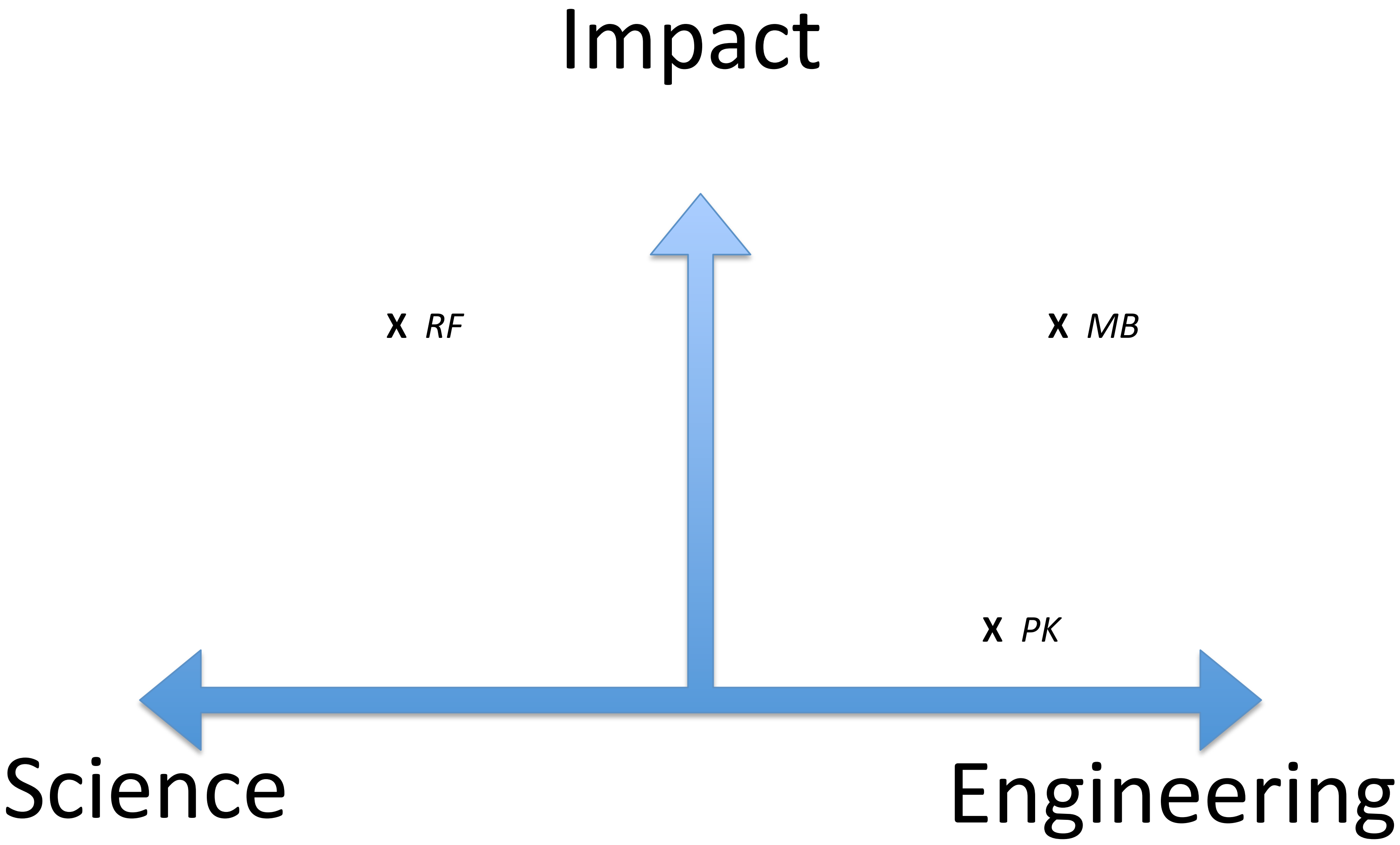
Impact



# users
$$
Citation

Science

Engineering



Impact

X *RF*

X *MB*

X *PK*

Science

Engineering

Research method

Scientific method

Engineering ‘method’

Empirical method

Analytical method

Explain what you do, and why

- Case study
- Survey
- Experiment
- Ethnographic study
- Secondary study

…

…

- Action research

- Grounded Theory

Science AND engineering

- Tool building:
  - Articulate what you do based on some theory
  - Clean, clear concepts

- Science:
  - Validate the usefulness by some practical and realistic implementation; usage

Traps in tool building

- Good tool => lots of work
  - Only a tiny fraction is PhD material
- Validation is hard (experiment)
  - Better mouser trap ? Really?
- Isolation

  => Feed the beast
- Dissemination (commercialization?)
- Sustainability

Traps in science appraoch

- “Everything software folks do can be reduced to a huge graph.”
- “Everything can be reduced to second order logic.”

- And then what?

(OK, I am an engineer, I have some biases, here)

Validity

- … Or how much do I believe what your claim

- Construct validity
  - Flimsy foundation, not a real problem, too much bias
- Internal validity
  - Research not conducted properly; lack of rigour
- Conclusion validity
  - Gee, I would have concluded something different
- External validity
  - Cannot be generalized

Contribution

- Make a contribution; is it valuable, for whom?
- *Do not just "go through the ropes"*

- Be clear in your head (and with your supervisor)
  - Motivation (why are you doing this)
  - Impact
  - Approach (method & validation)

- Ethical behaviour

Success

- Take the means to really achieve success the way you defined success.
  - Optimize for that form of success.

- Communication
  - Socialize your ideas, contributions…
- Secrecy
  - Patent, trade secret, etc…

More personal gripes

- Not all human knowledge is in journal papers indexed by IEEExplore (or web of sci, or scopus)
- Renaming old concepts is not innovation, just obfuscation
- Software engineering students are not representative of the software development population
- This is not a battlefield

More gripes

- Try your survey with 5 people (not close friends); then fix it and try with 20 different people; then fix it and only then send it to 4,000 people. Indicate typical duration.
- Qualitative research (e.g., ethnographic study) is not just a name for “I did not collect any data so I am going to wing it based on my notes”

Slides at philippe.kruchten.com/talks/

Aside: under the hood: my Filter "engine"

1. Abstract
2. Introduction (… motivation, concept definition)
3. Conclusion (claims)
4. Method(ology)
5. Body, seeing how it support claims
6. References (only as needed)
7. Validation
8. Related work
9. Abstract (2) & Title
10. Conclusion (2)
11. References (all, completeness)