



Ivan Beschastnikh

Computer Science
University of British Columbia
Vancouver, Canada



Software Practices



Gail
Murphy



Gregor Kiczales



Reid Holmes



William
Bowman



Ron Garcia



Ivan Beschastnikh

Computer Science
University of British Columbia
Vancouver, Canada



Networks Systems Security



Margo
Seltzer



Mathias
Léuyer



Norm
Hutchinson



Alan
Wagner



Mike Feeley



Software Practices

Networks Systems Security



Gail
Murphy



Gregor Kiczales



Margo
Seltzer



Mathias
Léuyer



Norm
Hutchinson



Ron Garcia



William
Bowman



Reid Holmes



Ivan Beschastnikh



Alan
Wagner



Mike Feeley

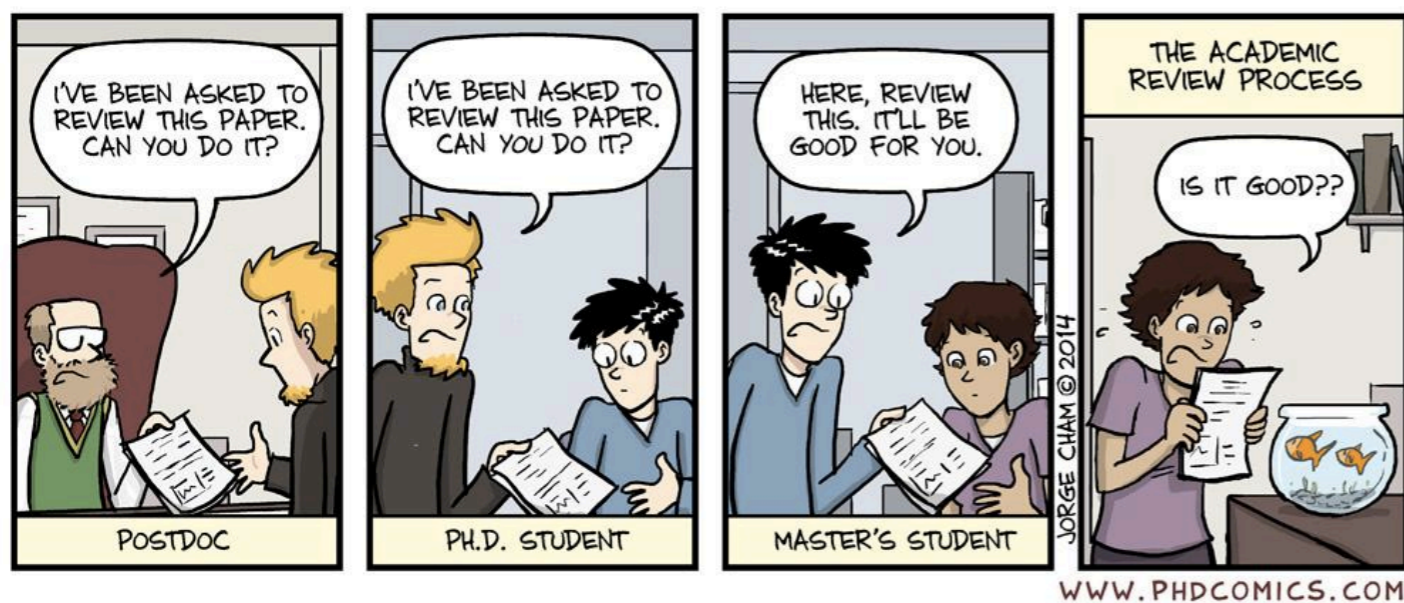


Computer Science
University of British Columbia
Vancouver, Canada



Successfully Publishing your Research

Think about peer review, frequently!



Ivan Beschastnikh

Networks, Systems and Security Lab

Software Practices Lab

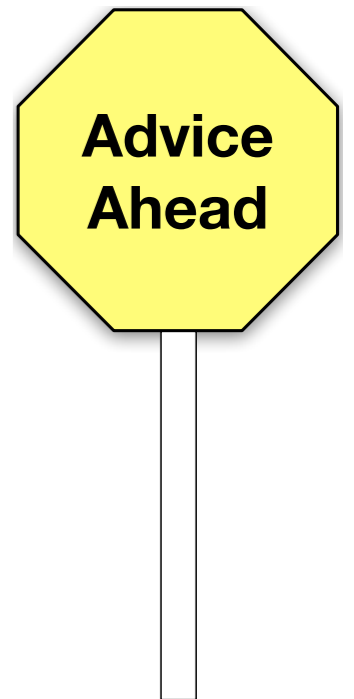


With special thanks to my
collaborators and countless
anonymous reviewers

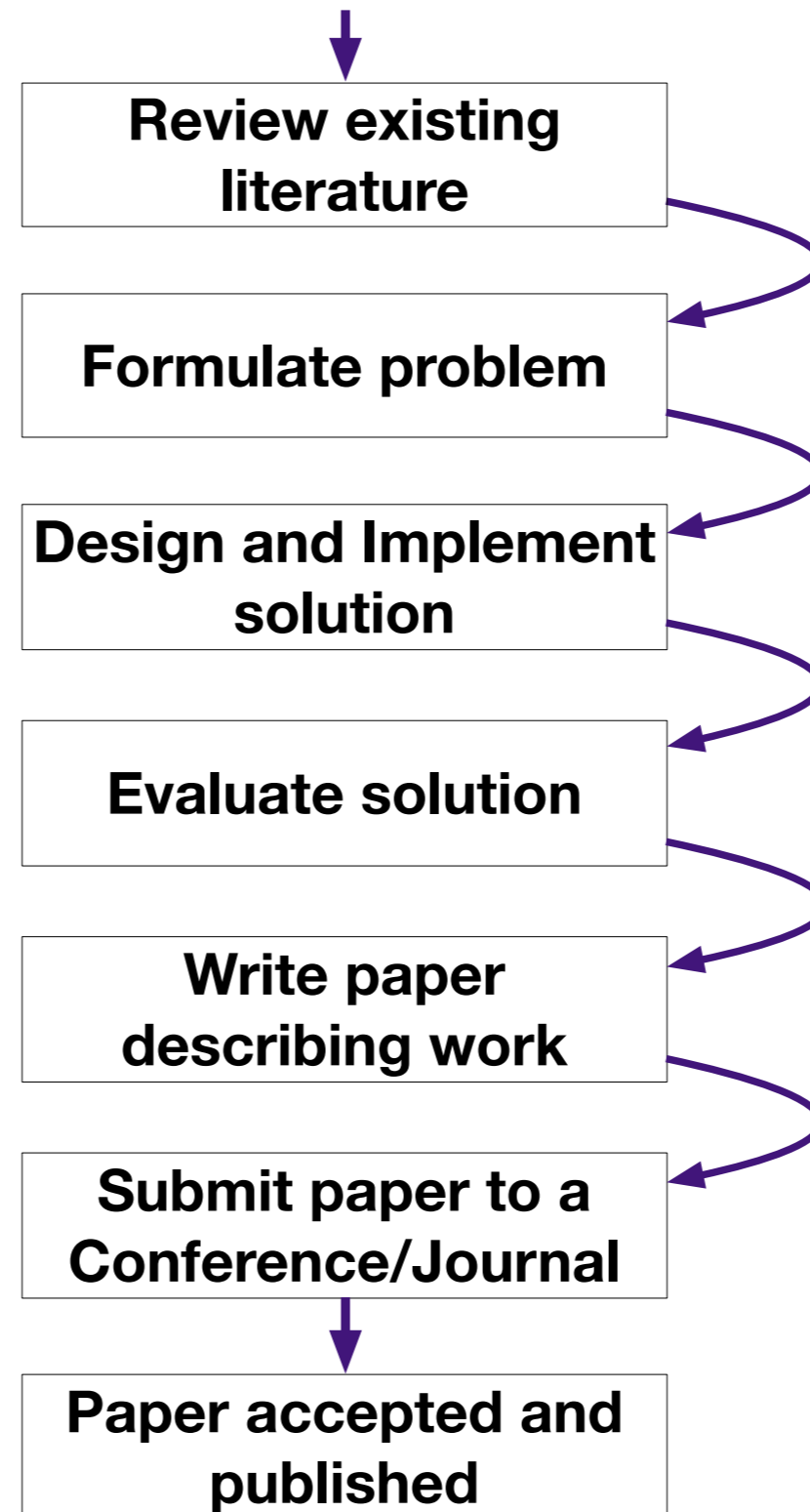


Warning: meta-talk with advice ahead

- Talk with *advice*: how you/we should do research, not a talk about some research I've done
- Question my advice + solicit others' viewpoints:
 - Ask me questions after the talk
 - Talk to others in this room! They can share their experiences, views, and advice
- *Short* talk: particular focus with simplifications and omissions
Writing quality papers takes years to learn, I have an hour

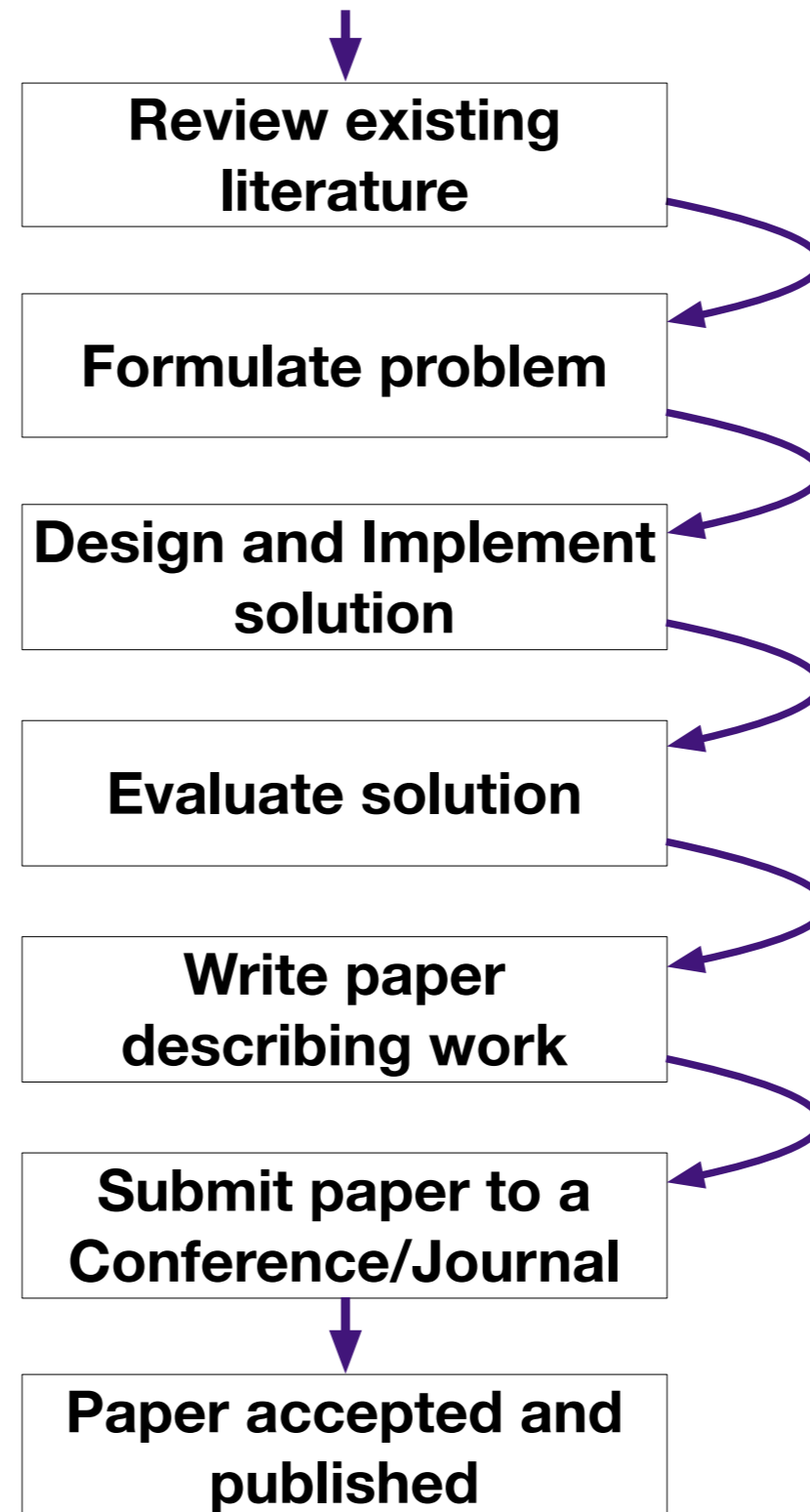


Idealized research process



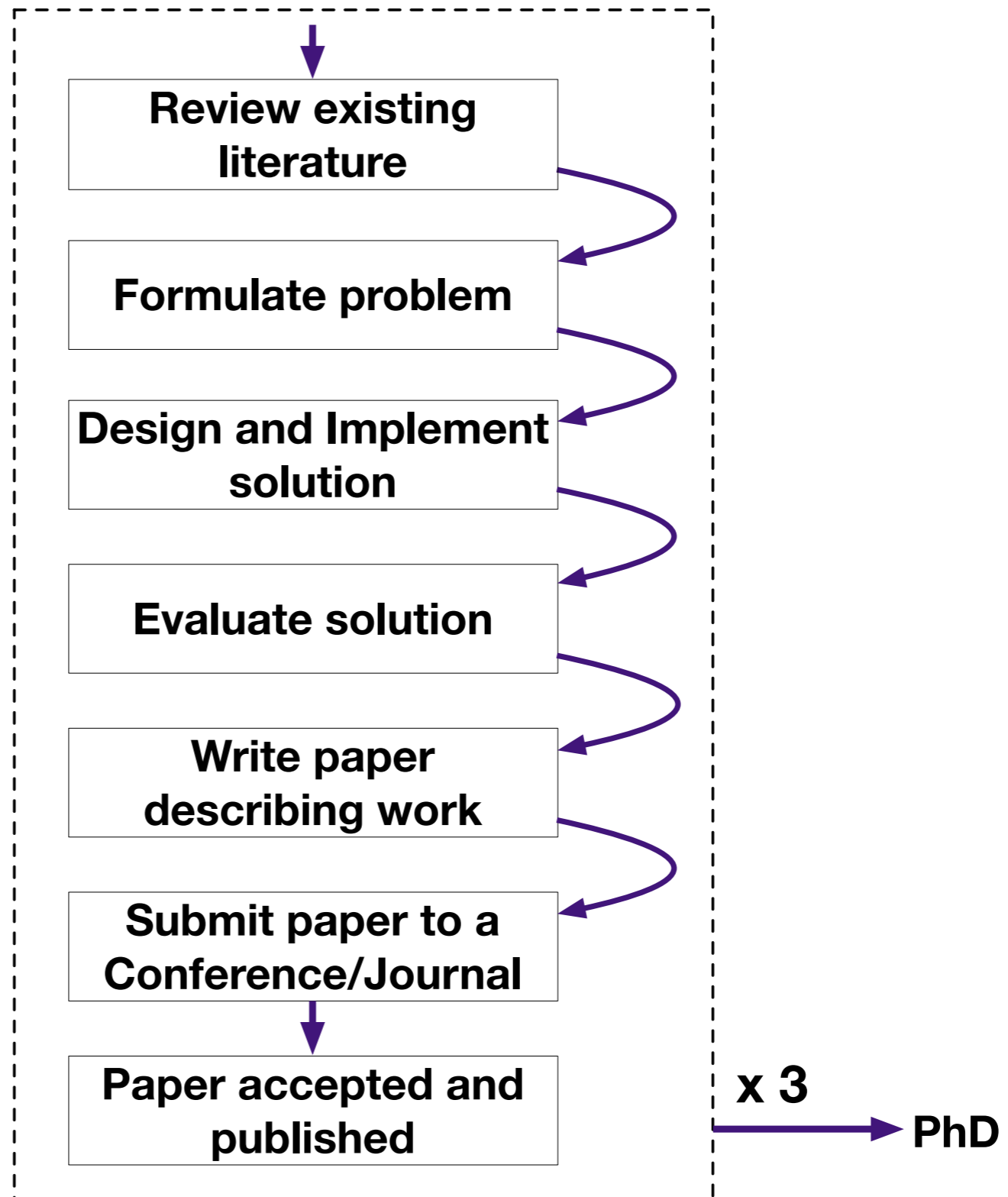
Idealized research process

The *happy path* sequence in research



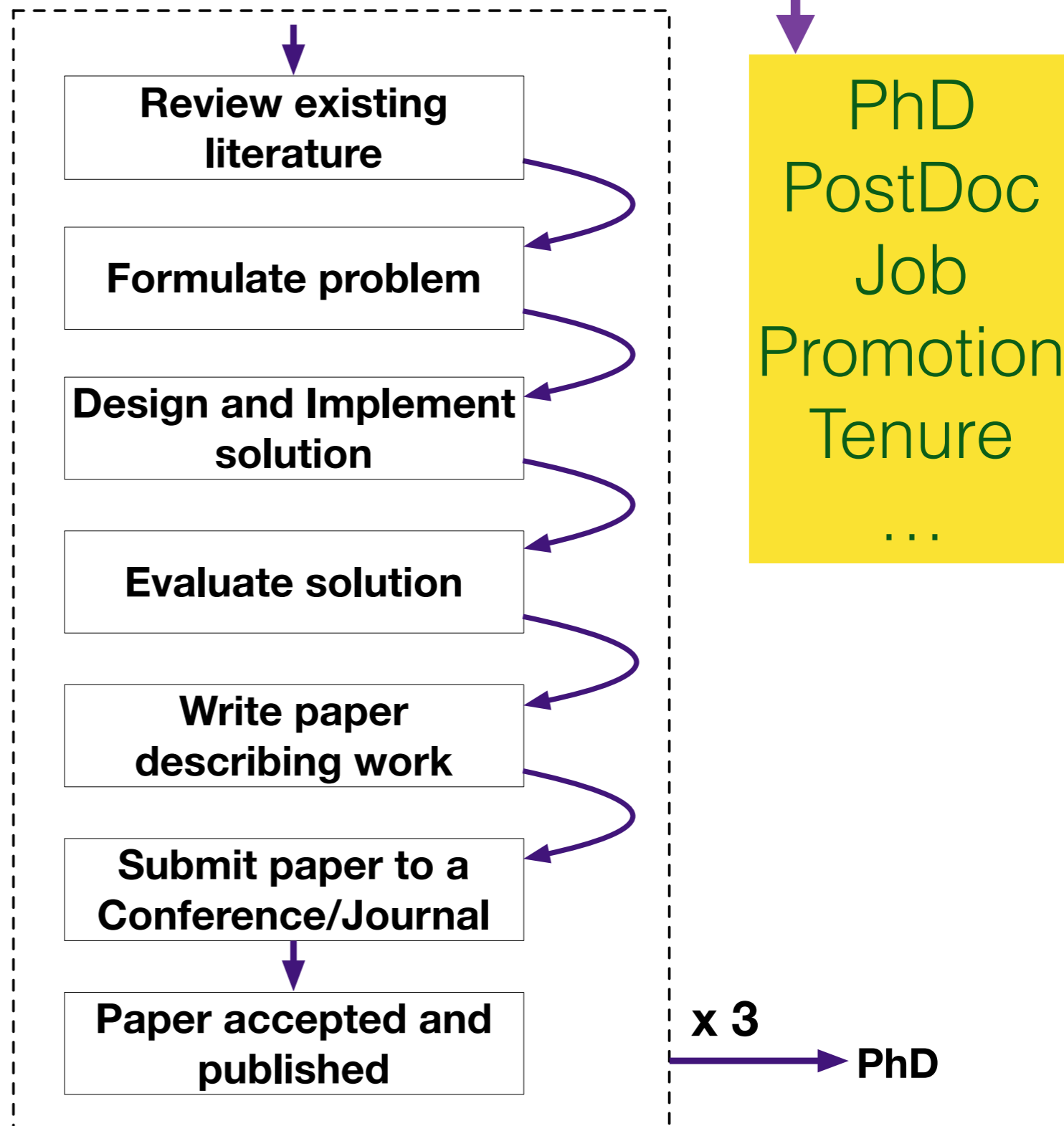
If you do this well, you'll get a PhD

The *happy path*
sequence in
research



If you do it well, you'll get

The *happy path* sequence in research

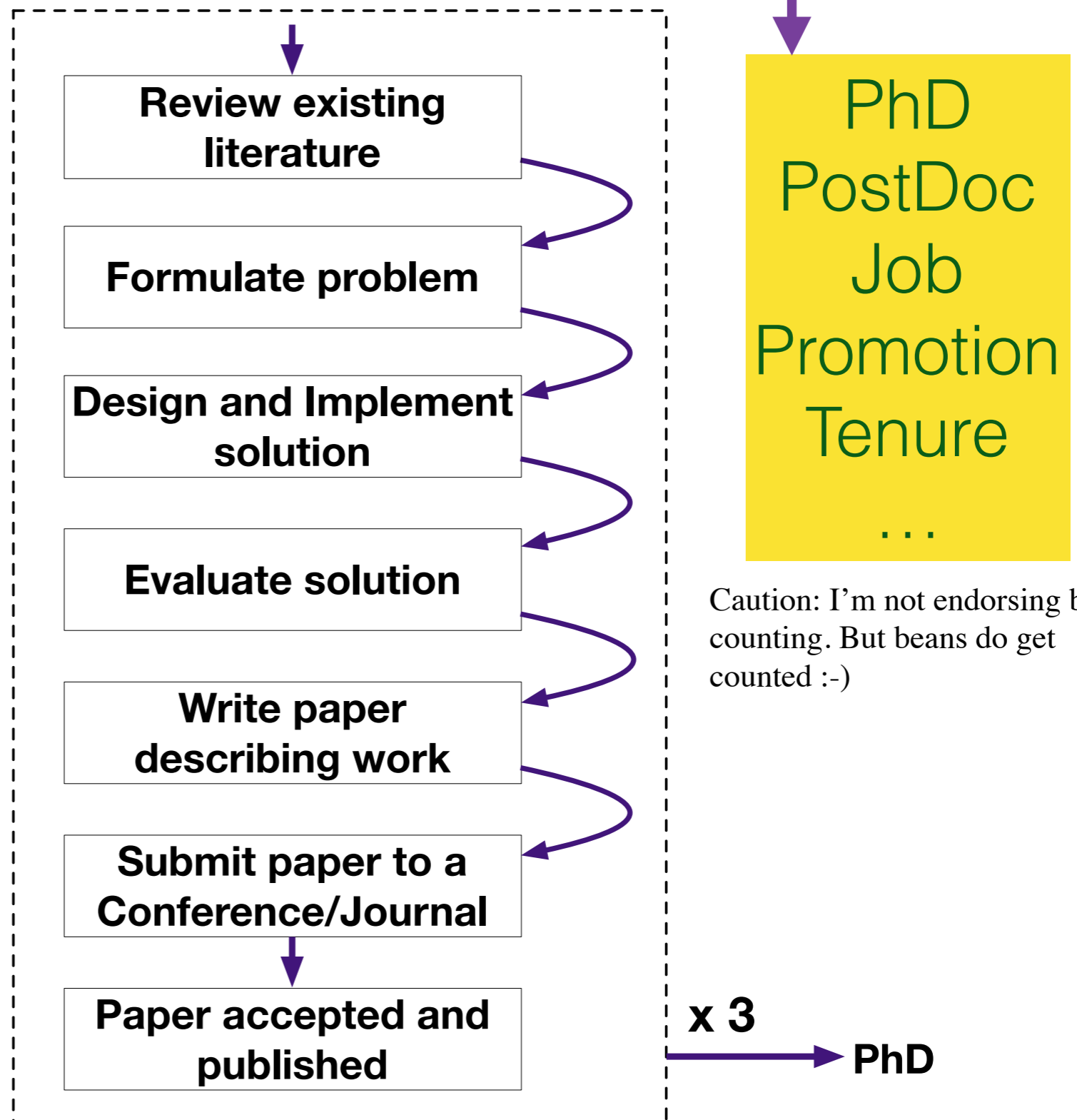


If you do it well, you'll get

The *happy path* sequence in research

Caution: I focus on publishing, but research is so much more than paper writing!

[1] Research should not stop with the research paper
<https://lemire.me/blog/2020/02/07/research-should-not-stop-with-the-research-paper/>

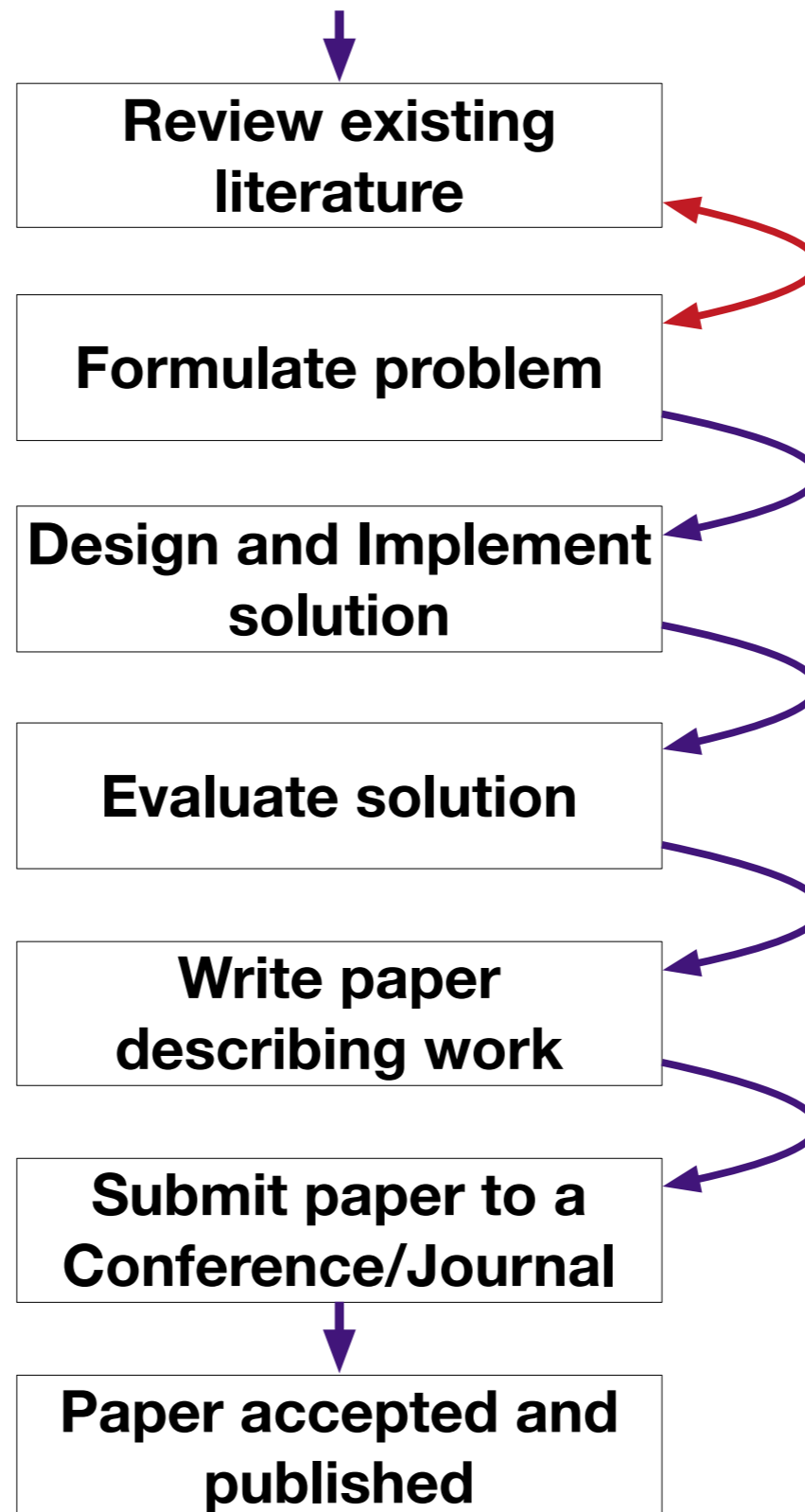


Caution: I'm not endorsing bean counting. But beans do get counted :-)



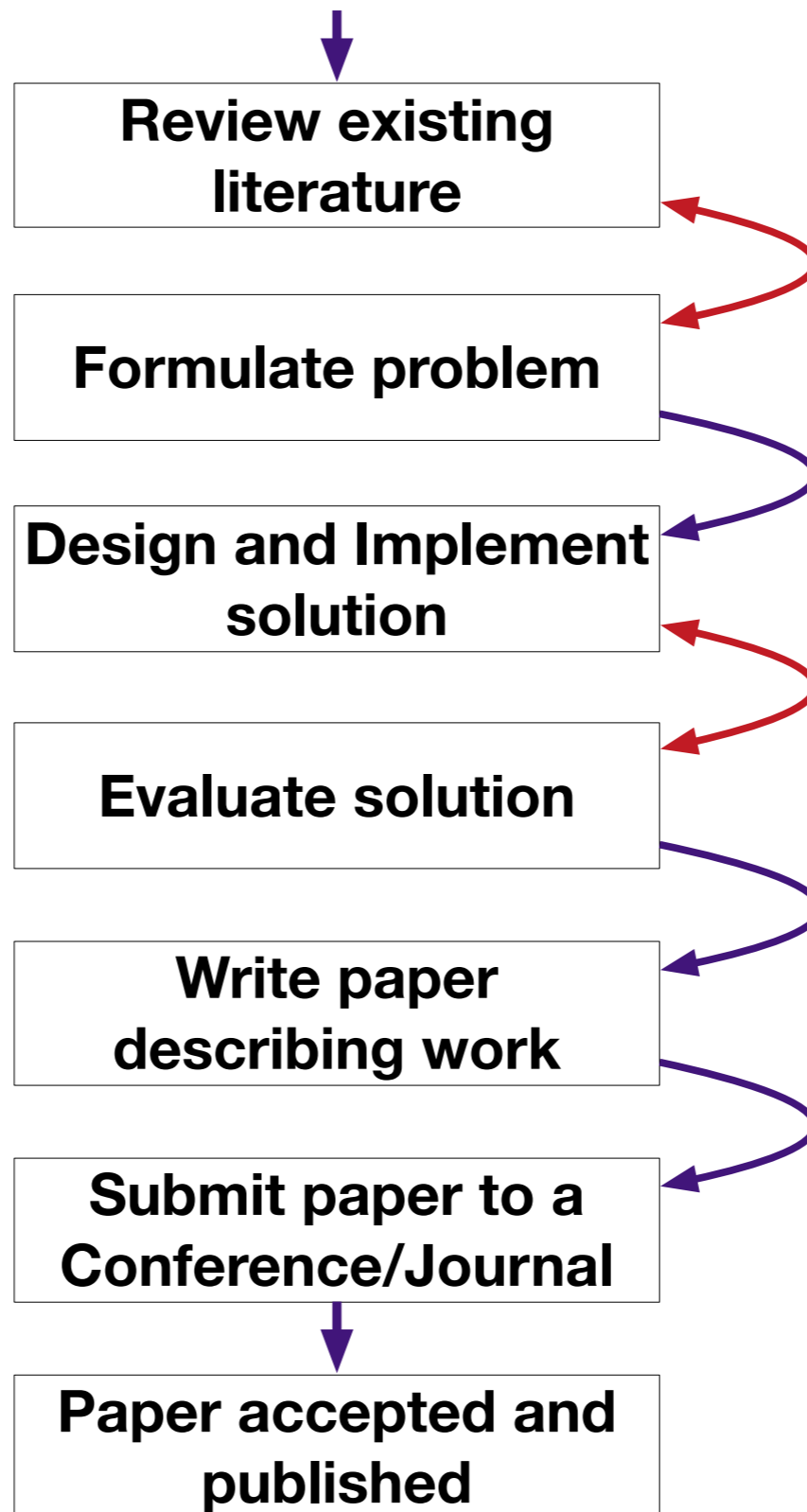
The problem has been solved by many!

**More complicated
in practice, lots
that could go
wrong**

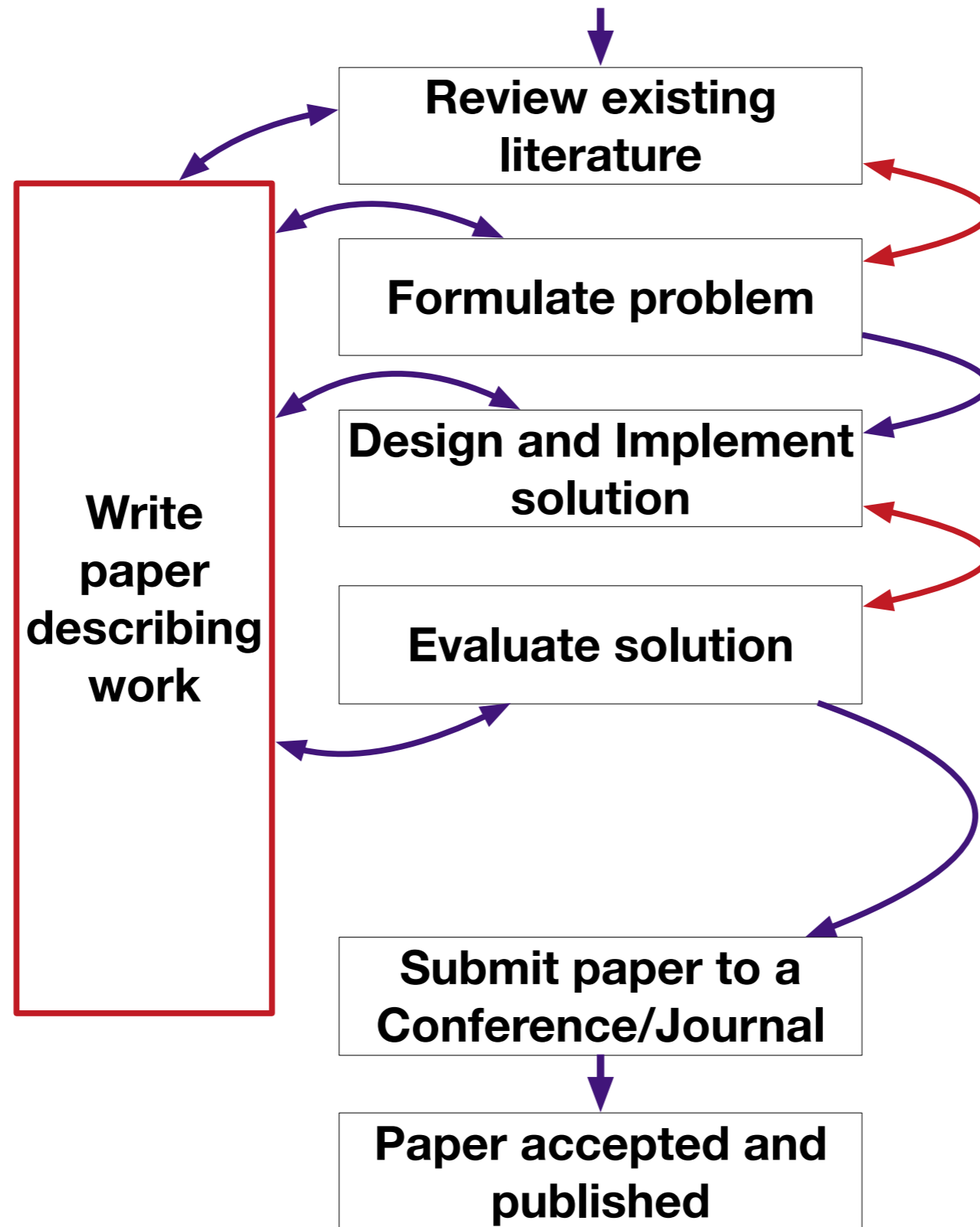


The approach doesn't work

**More complicated
in practice, lots
that could go
wrong**



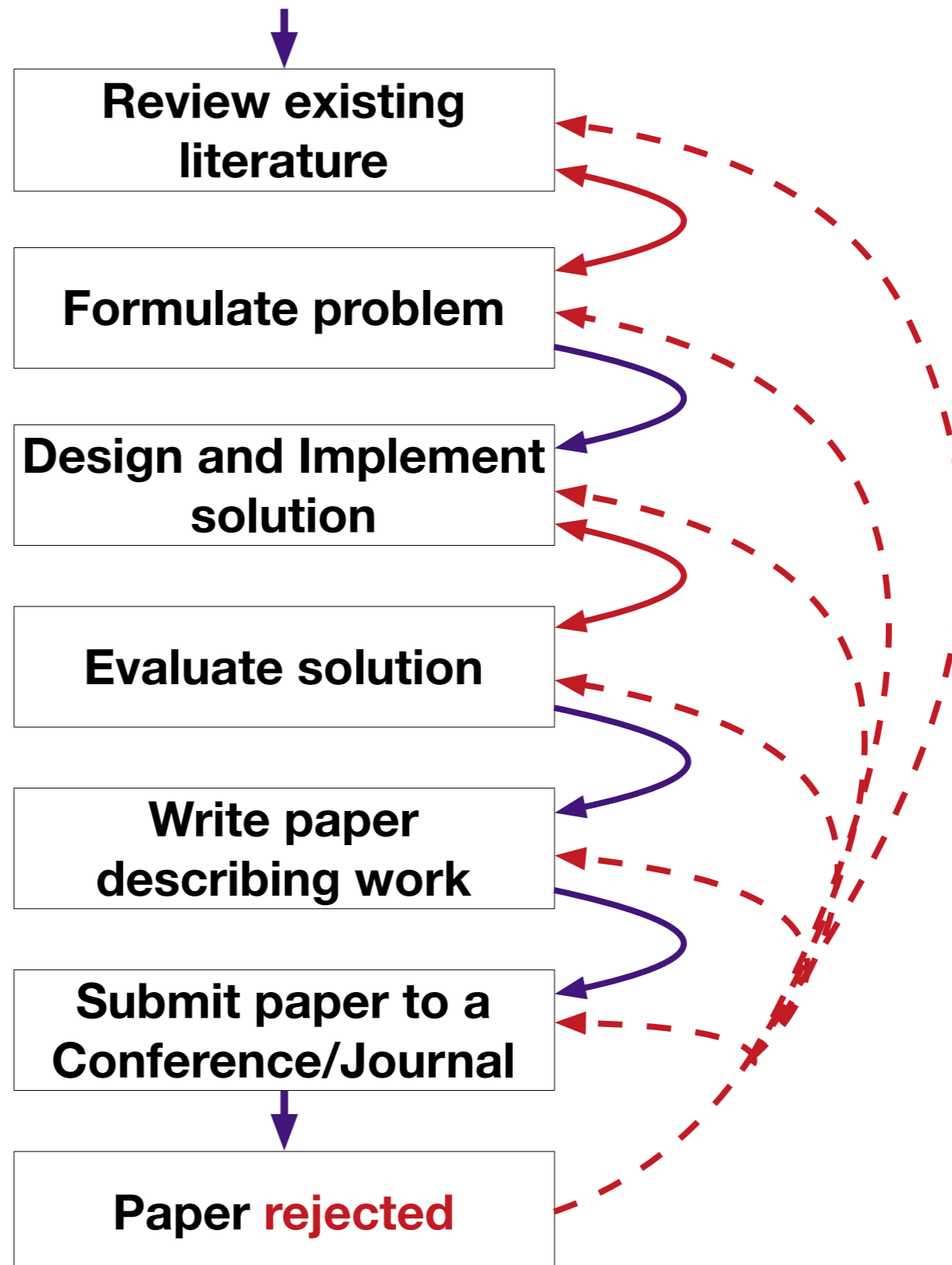
(Good) paper writing in practice



**Many ways to
organize the
research
process!**

“The program committee is sorry to inform you...”

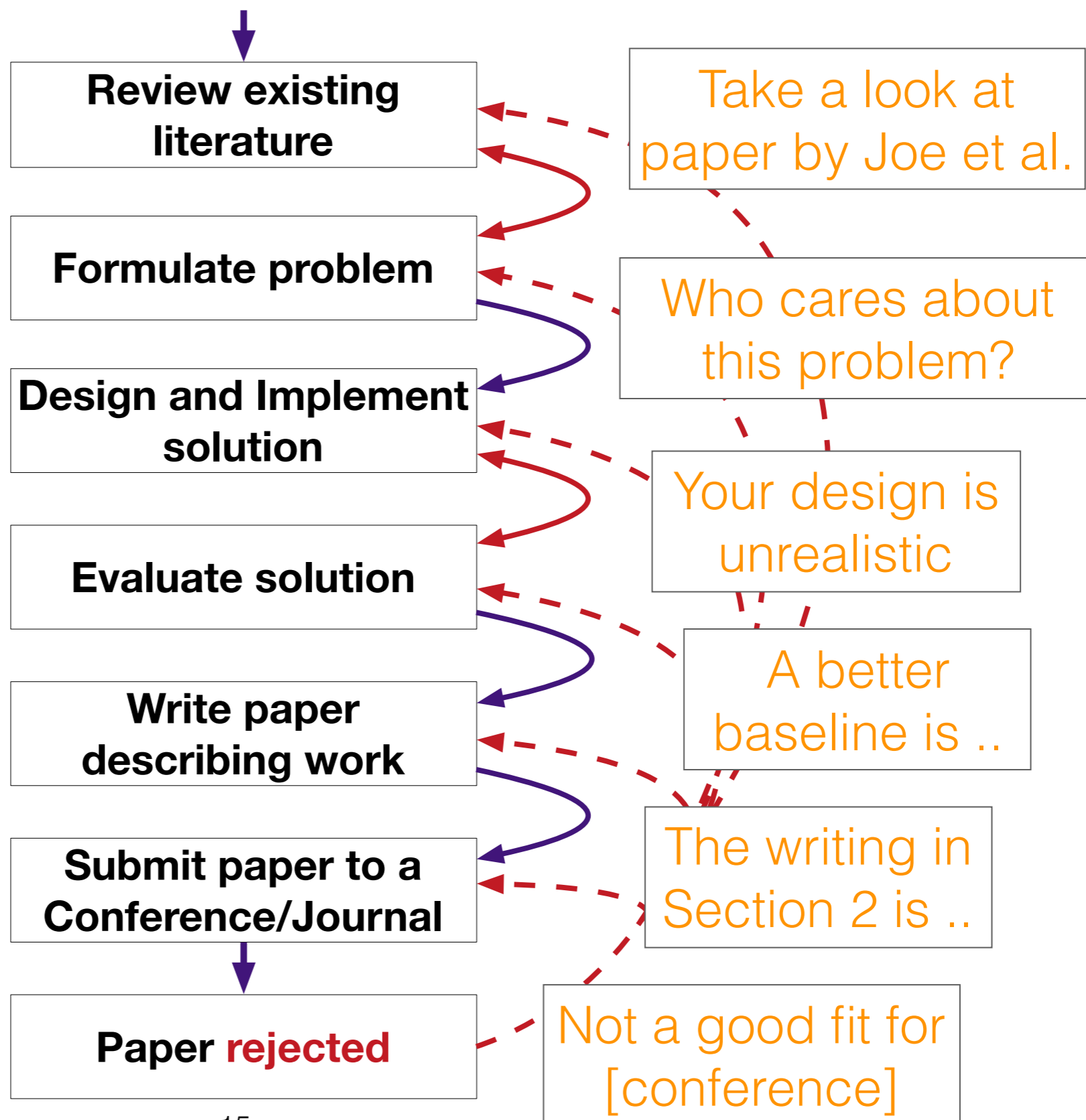
Rejection is the norm for majority of papers submitted to top venues



“...we hope that you find the reviews helpful”

Rejection is the norm for majority of papers submitted to top venues

“ In most cases, the reviews offer an opportunity to improve the work, and so you should be very grateful for a rejection! It is much better for your career if a good paper appears at a later date, rather than than a poor paper earlier or a sequence of weak papers.” — Mike Ernst, my advisor



Key omission: the peer-review

Easy to ignore initially:

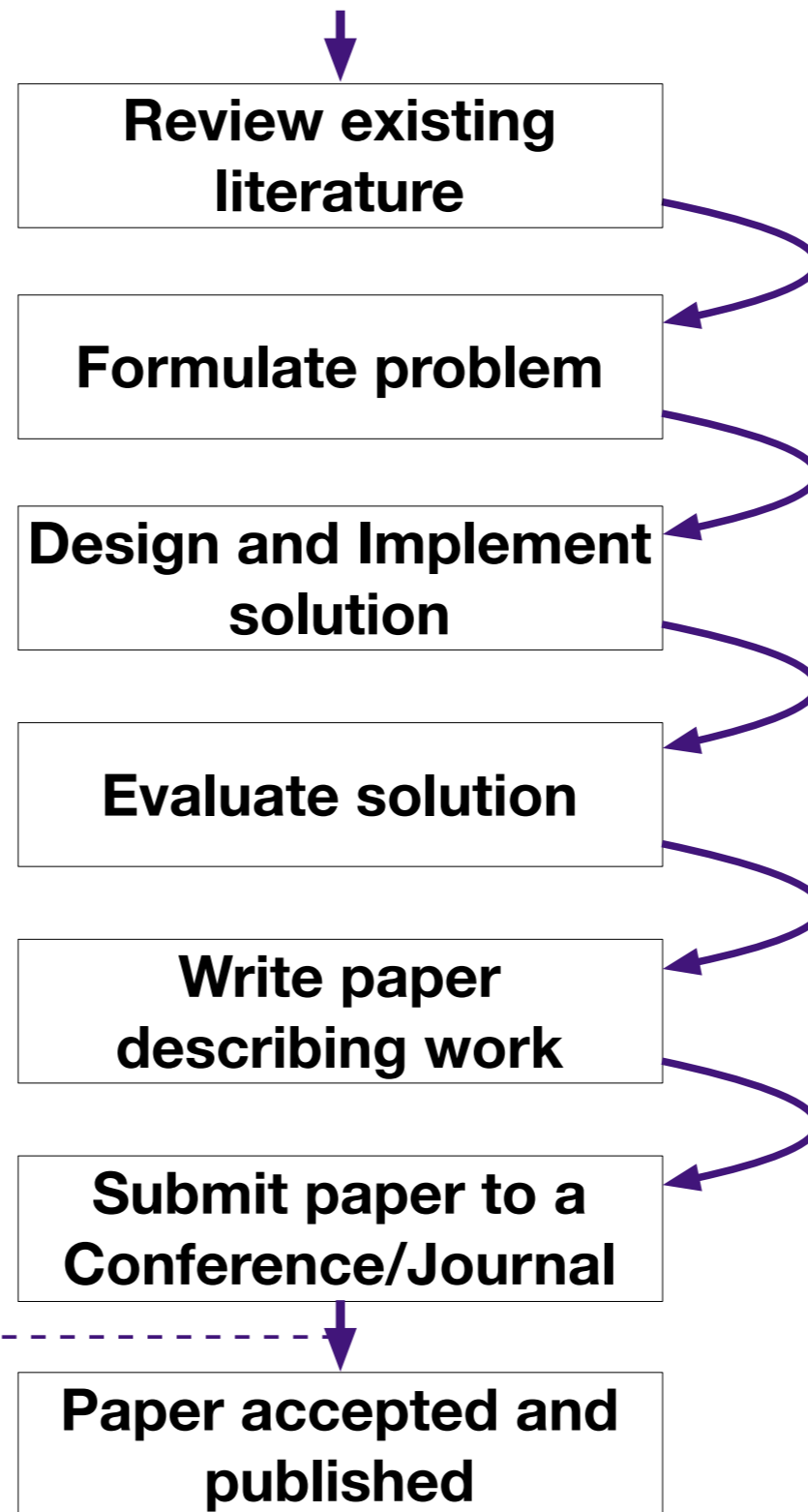
- It's those older people I don't know
- They are wise, I trust them
- I can't control what they will say
- How does a PC work anyway?

Ignore at your own risk!



Peer-review

Program Committee (PC)



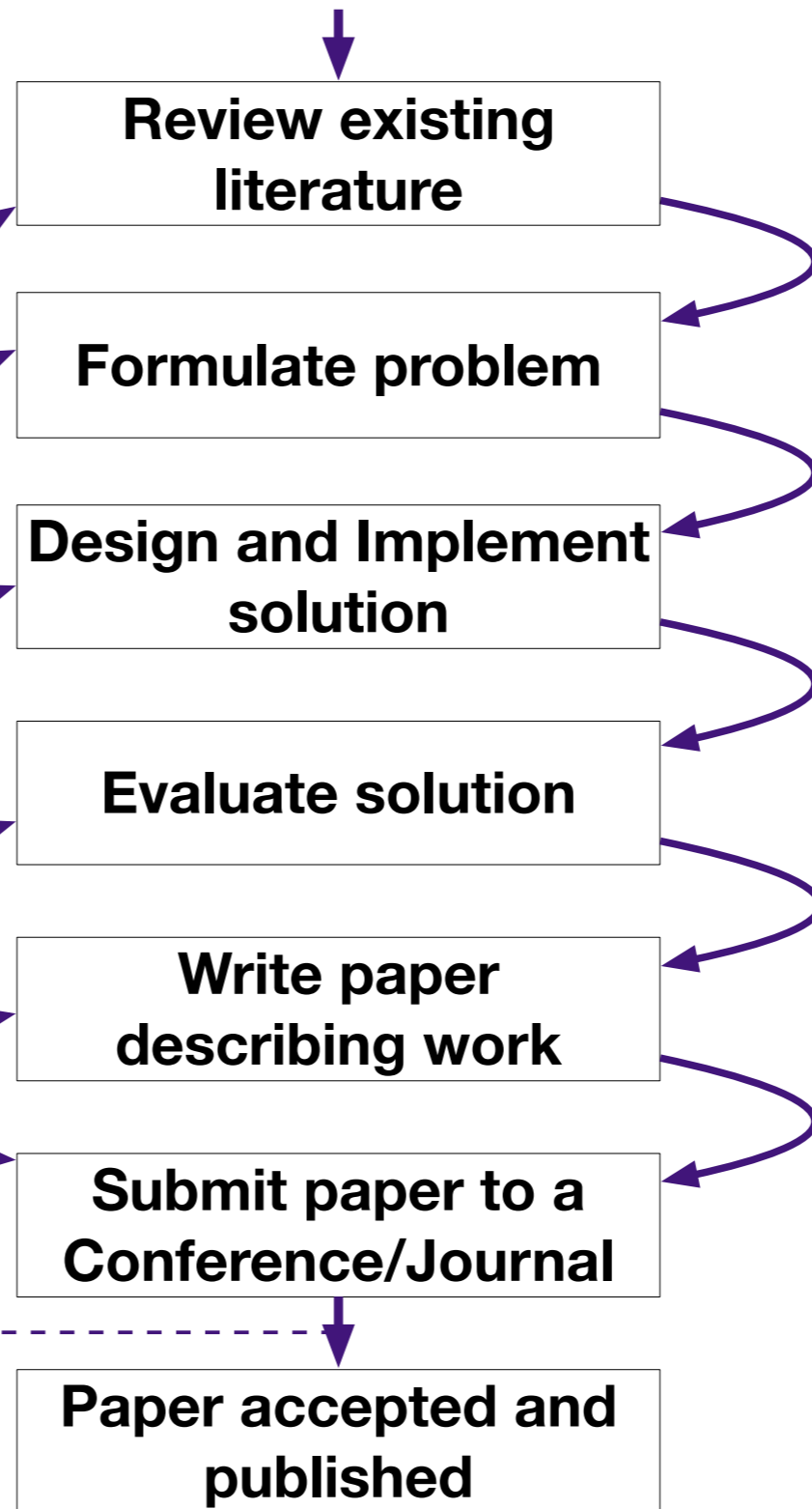
Talk take-away: Keep peer-review in mind



Talk take-away: Keep peer-review in mind

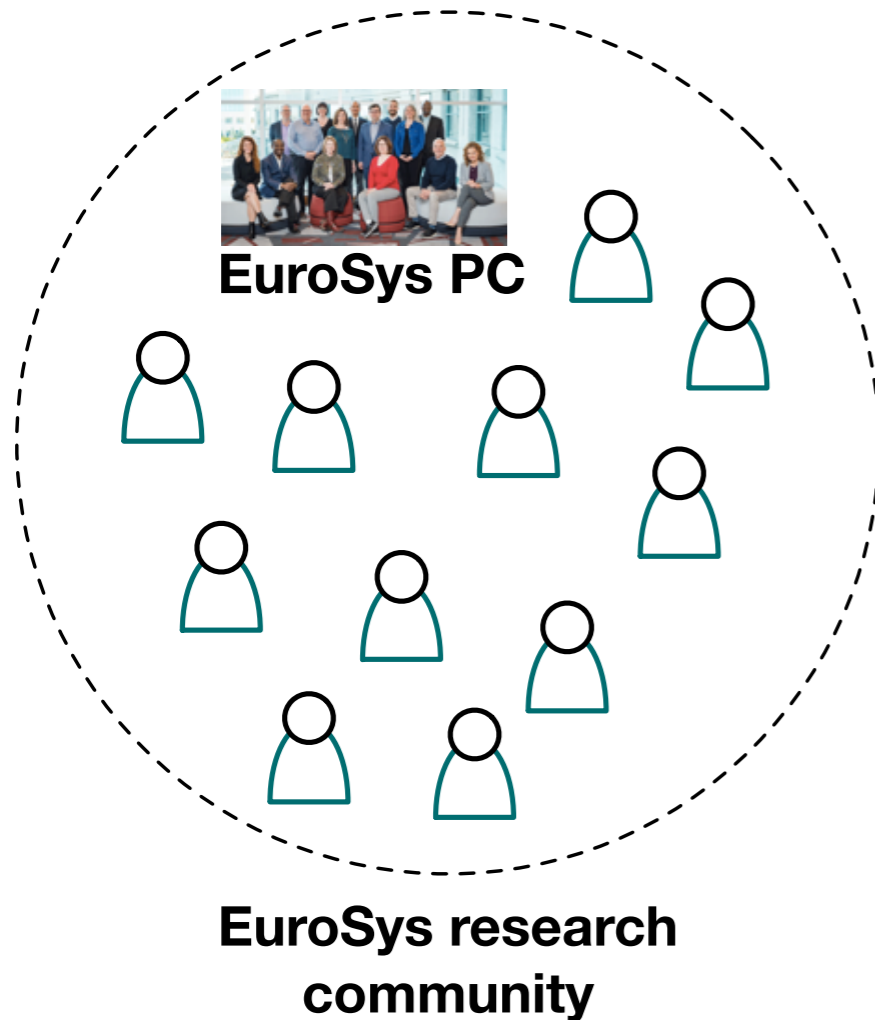


Peer-review



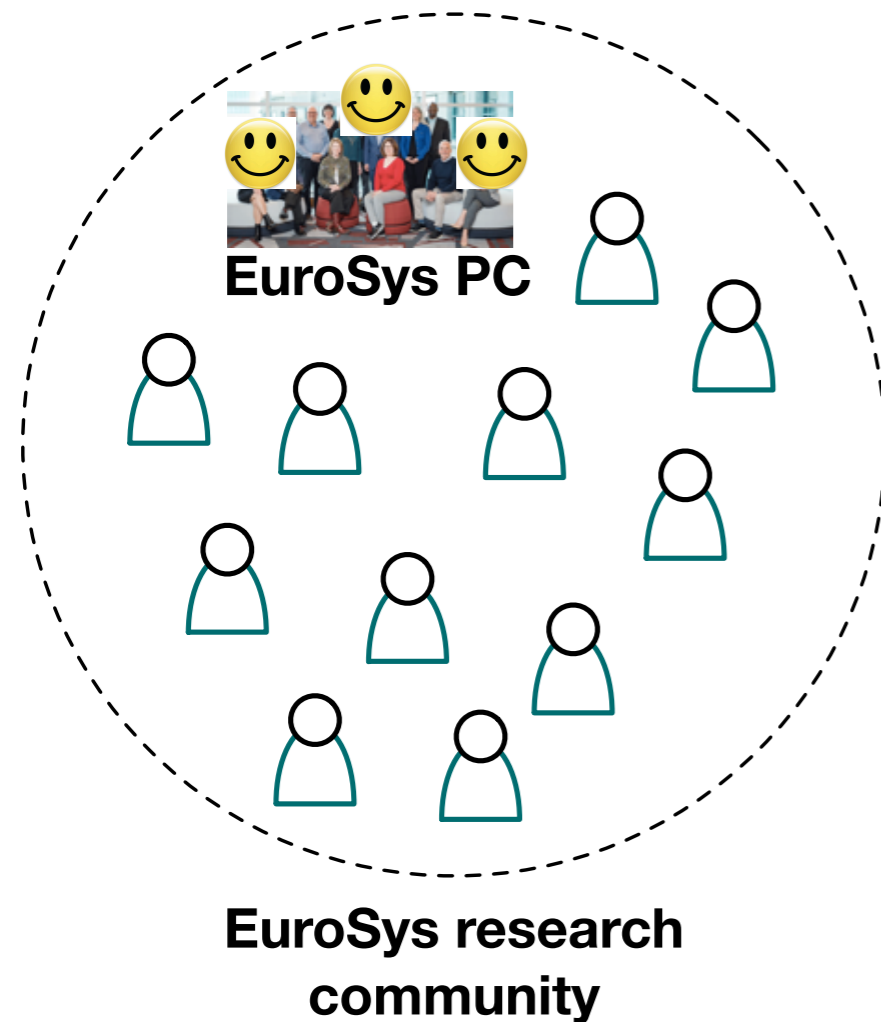
Program Committee (PC)

Who is on the PC anyway?



- Typically 20-40 people
- Varies from year to year
- Selected to represent various sectors of the community: geography, gender, topics, seniority, etc.
- Led by PC chair(s) who lead the peer-review process

How does the PC do its work?



- Usually several rounds of reviewing
- In each round, a reviewer is assigned 10-25 papers
- **Each paper reviewed by ~3 reviewers per round**
- Paper moves from round to round if there is enough support
- Final paper decisions made at a PC meeting (online or offline)
- **Accepted papers usually require a champion on the PC**

Views on role of the PC

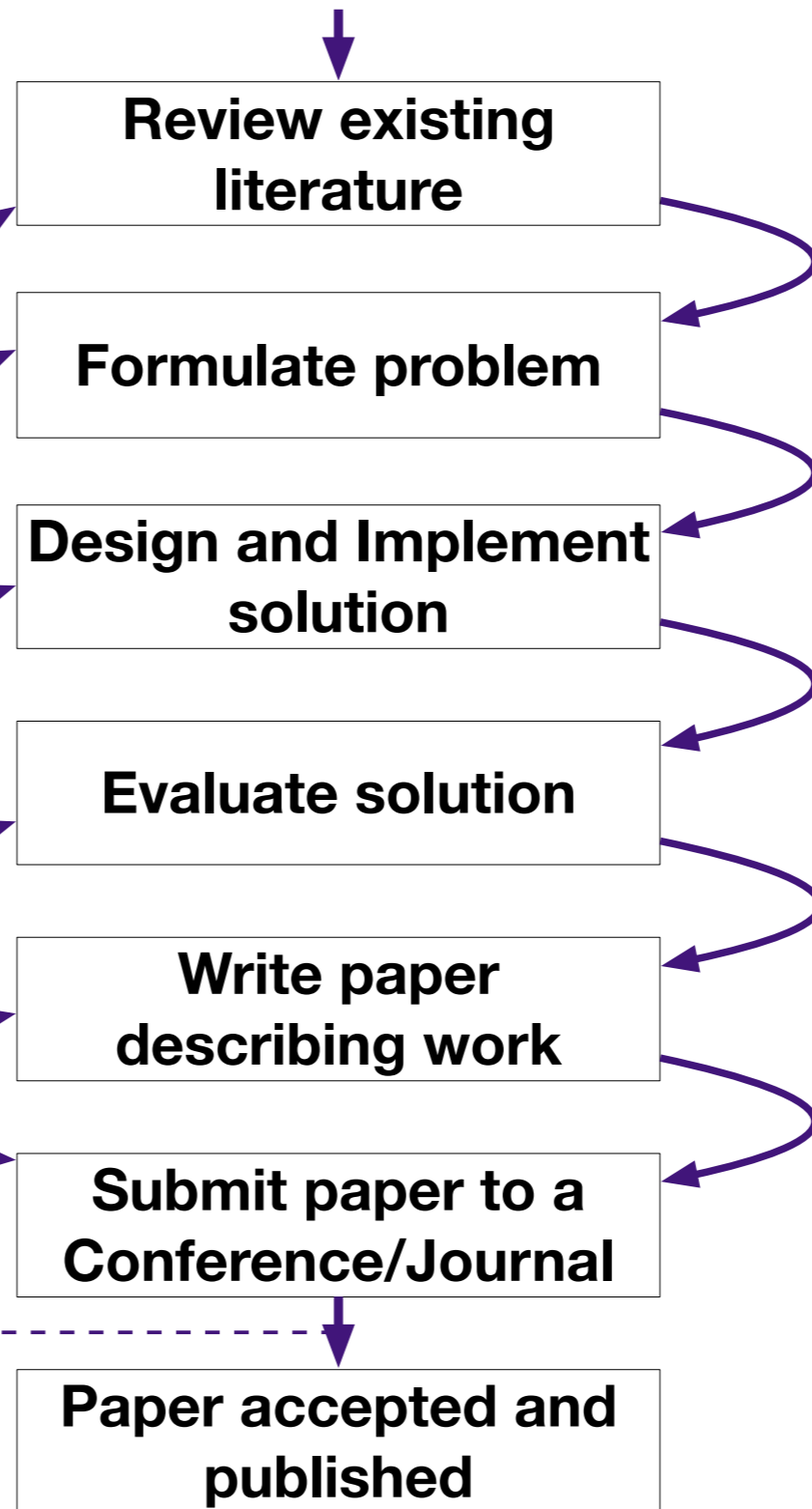


- **Naive view:** PC is a set of experts who judge my work
 - Usually the authors are more expert than the PC!
- **Pessimistic view:** PC is a gatekeeper. Conference can only accept X papers, so someone has to select them.
 - Usually conference organizers *want* to accept more papers
- **More accurate view:** PC is the audience for your work! They are *representative* of the broader community. A rejection is valuable info!
 - Rejection = the work is not ready for broad dissemination, e.g., will not be understood, appreciated, have as much impact without more work. Generally: if you address the concerns, then paper will be accepted

Talk take-away: Keep peer-review in mind



Peer-review



Program Committee (PC)

Talk take-away: Keep peer-review in mind



Peer-review

Review existing literature

But, how?

- Strategies (high-level advice)
- Tactics (low-level advice)

Solution

Evaluate solution

Write paper describing work

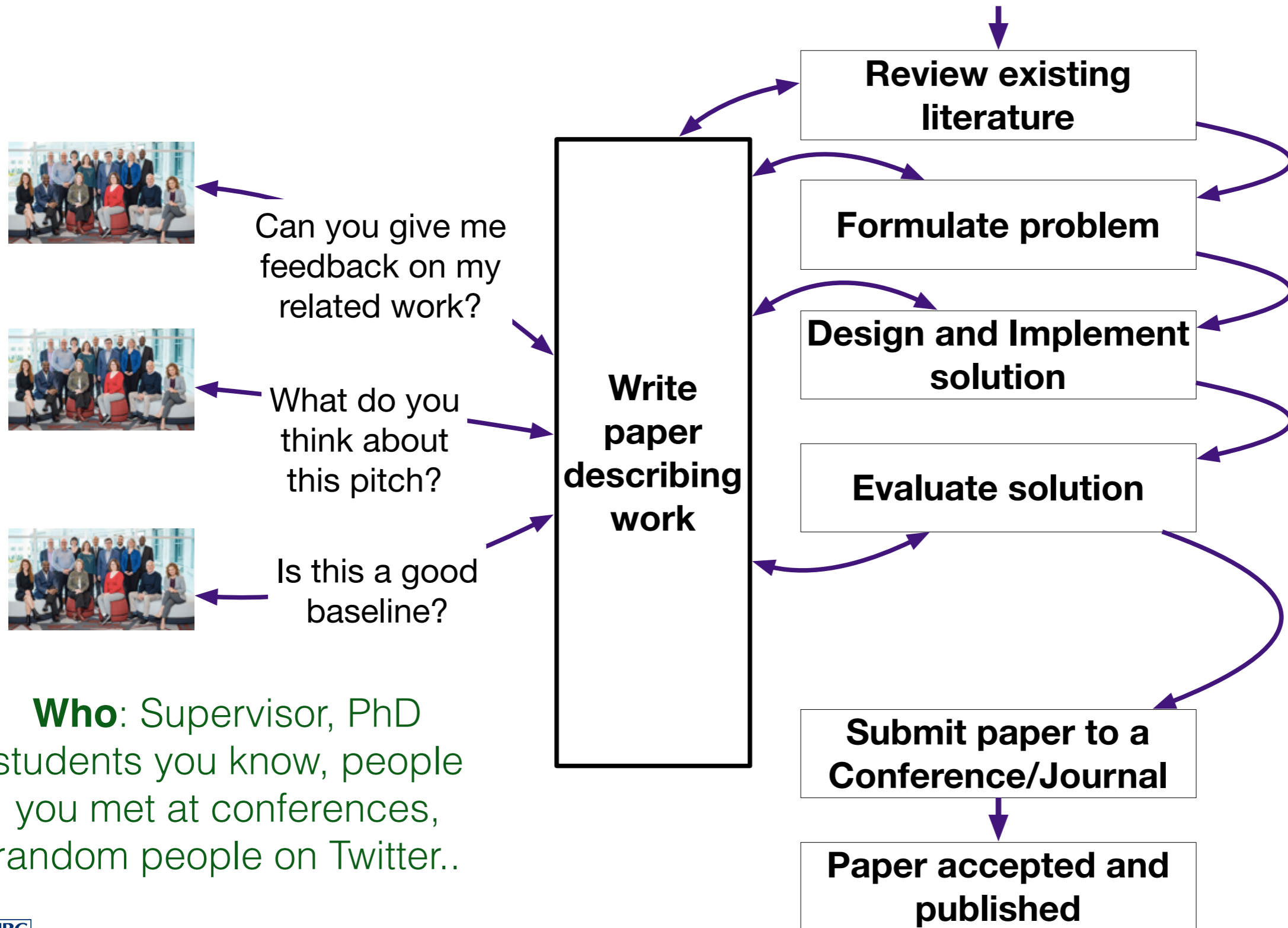
Submit paper to a Conference/Journal

Paper accepted and published

Program Committee (PC)



Strategy 1: Consult your own program committee during the research process



Who: Supervisor, PhD students you know, people you met at conferences, random people on Twitter..

Strategy 2: Read many (good) papers

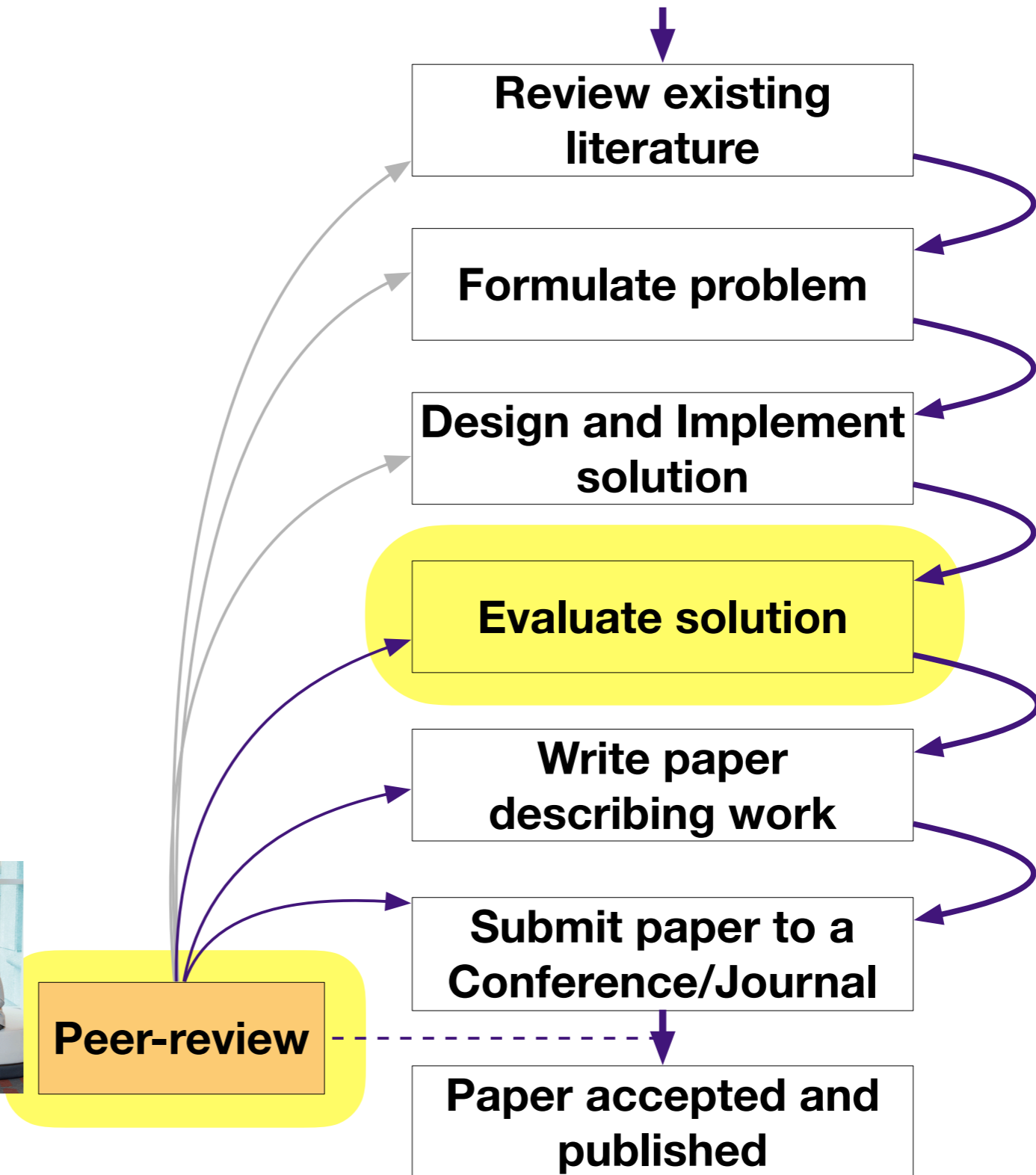
- The PC is selected based on their representation of the community
- PC members have published many papers: read those!
- Even if the paper is not in your domain, it will help you to *assimilate the norms of the community*
 - Hot/cold topics, problems community cares about, history!
 - Accepted versus niche experimental methods
 - Benchmarks and evaluation criteria
 - Writing style: “*This paper is more appropriate for NSDI*”

Aside: know your community!

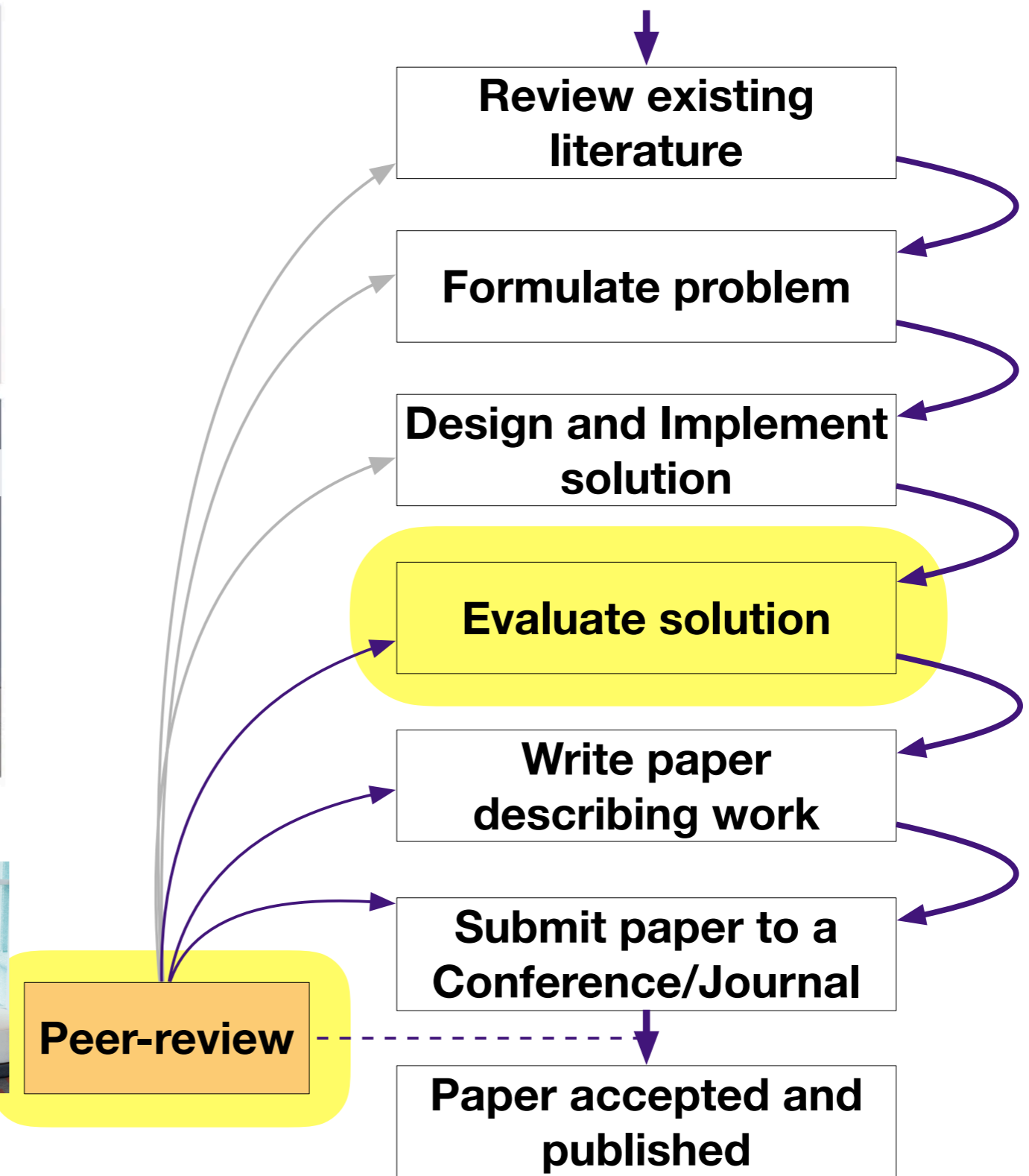
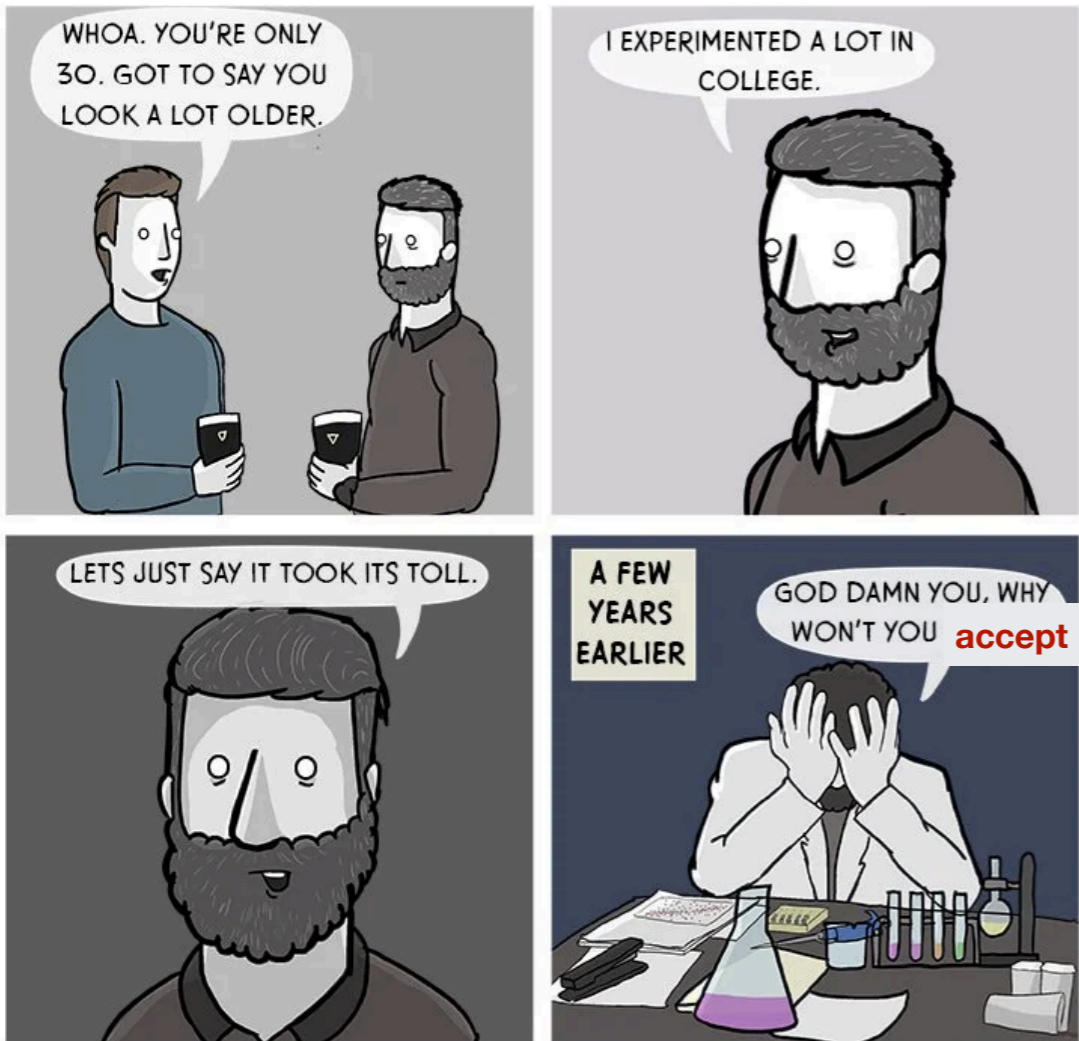
- Academia is clique-ish: many overlapping communities, typically identified by a conference/journal, topic, or methodology
- Read papers in venues where you want to publish them
 - By publishing in venues X , you are implicitly joining community X !
- Try to attend conferences in your community (even if you don't have a paper)
- Different communities have different paper norms/practices
 - SE community (e.g., ICSE): Explicit RQs, Threats to Validity section, care with user studies, deployment of prototypes is rare
 - Sys community (e.g., EuroSys): Perf and benchmarks focus, working + deployed prototypes, evaluation emphasis on trade-offs



Rest of talk: focus on evaluation



Rest of talk: focus on evaluation

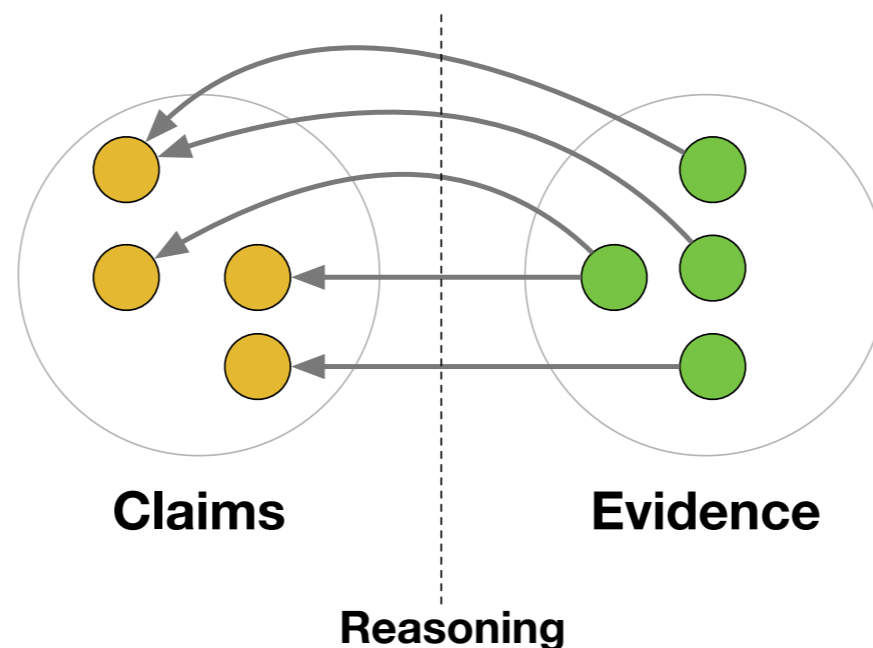


Evaluation: it's about **confidence**

Most research papers in our field make **claims** and provide **evidence** for those claims

- **Claims:** statements about the world (your system) that can be empirically validated (i.e., refutable)
 - *System X has higher throughput than system Y*
- **Evidence:** material to convince the reader about claims
 - *Measurement results that show that system X is faster than system Y*

Each claim must have corresponding evidence



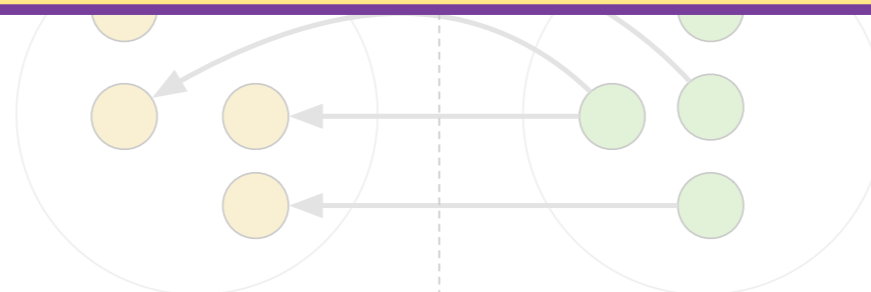
Each evidence must correspond to some claim

Evaluation: it's about **confidence**

Most research papers in our field make **claims** and provide **evidence** for those claims

- **Claims**
 - **As empiricists we obsess over evidence**
 - Numerous methodological choices
 - Many technical skills [e.g., stats]
 - **Strategy 3: think hard about paper claims!**
 - As with paper pitches: the why and who cares are frequently more important than how
 - Claims often provide much more flexibility
- **Evidence**

Each claim must have corresponding evidence



Each evidence must correspond to some claim

Evaluation: it's about **confidence**

Most research papers in our field make **claims** and provide **evidence** for those claims

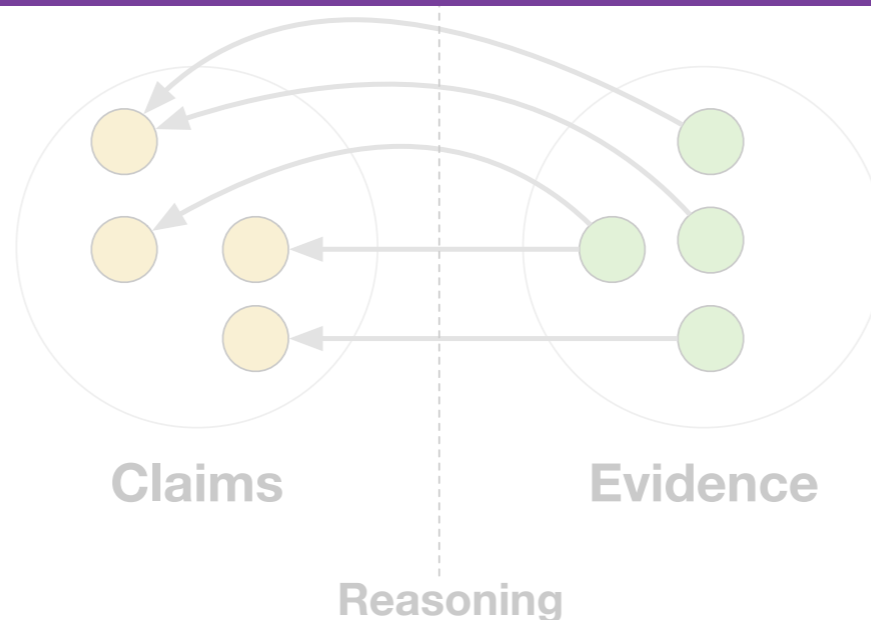
- **Claims:** statements about the world (your system) that can be empirically validated (i.e. refutable)

Bottom line: You want your evaluation to be *convincing*.

- **Evidence**

Help the PC accept your paper by giving them *confidence* in your work.

Each claim must have corresponding evidence



Each evidence must correspond to some claim

Tighten those claims

Reviewer (and author!) claims check-list

- Claims are precise and explicit (no implicit claims)
 - *What are you promising and not promising?*
- Claims match the problem and design
 - *Does the claim make sense in this context?*
- Claims are interesting and non-trivial
 - *Will I learn something if I find out the answer to this claim?*
- Claims do not over-promise
 - *Will any amount of evidence convince me of this claim?*

Think about how PC thinks about your claims!

Reviewer (and author!) claims check-list

- **Claims are precise and explicit (no implicit claims)**

- *What are you promising and not promising?*
- *“Performance” means different things to different sys researchers: throughput, goodput, scalability, MTTR, etc*

- **Claims match the problem and design**

- *Does the claim make sense in this context?*
- *Reviewers associate certain claims with certain contexts: mobile and IoT ~ energy claims*

- **Claims are interesting and non-trivial**

- *Will I learn something if I find out the answer to this claim?*
- *Depends on what reviewers already know and what they care about!*

- **Claims do not over-promise**

- *Will any amount of evidence convince me of this claim?*
- *Perceptions of strength of claim and what evidence is expected: “large deployment”, utility ~ company use*



Think about how PC thinks about your claims!

Reviewer (and author!) claims check-list

- **Claims are precise and explicit (no implicit claims)**

- *What are you promising and not promising?*
- *“Performance” means different things to different sys researchers: throughput, goodput, scalability, MTTR, etc*

- **Claims match the problem and design**

- *Does the claim make sense in this context?*
- *Reviewers associate certain claims with certain contexts: mobile and IoT ~ energy claims*

- **Claims are interesting and non-trivial**

- *Will I learn something if I find out the answer to this claim?*
- *Depends on what reviewers already know and what they care about!*

- **Claims do not over-promise**

- *Will any amount of evidence convince me of this claim?*
- *Perceptions of strength of claim and what evidence is expected: “large deployment”, utility ~ company use*



Think about how PC thinks about your claims!

Reviewer (and author!) claims check-list

- **Claims are precise and explicit (no implicit claims)**

- *What are you promising and not promising?*
- *“Performance” means different things to different sys researchers: throughput, goodput, scalability, MTTR, etc*

- **Claims match the problem and design**

- *Does the claim make sense in this context?*
- *Reviewers associate certain claims with certain contexts: mobile and IoT ~ energy claims*

- **Claims are interesting and non-trivial**

- *Will I learn something if I find out the answer to this claim?*
- *Depends on what reviewers already know and what they care about!*

- **Claims do not over-promise**

- *Will any amount of evidence convince me of this claim?*
- *Perceptions of strength of claim and what evidence is expected: “large deployment”, utility ~ company use*



Think about how PC thinks about your claims!

Reviewer (and author!) claims check-list

- **Claims are precise and explicit (no implicit claims)**

- *What are you promising and not promising?*
- *“Performance” means different things to different sys researchers: throughput, goodput, scalability, MTTR, etc*

- **Claims match the problem and design**

- *Does the claim make sense in this context?*
- *Reviewers associate certain claims with certain contexts: mobile and IoT ~ energy claims*

- **Claims are interesting and non-trivial**

- *Will I learn something if I find out the answer to this claim?*
- *Depends on what reviewers already know and what they care about!*

- **Claims do not over-promise**

- *Will any amount of evidence convince me of this claim?*
- *Perceptions of strength of claim and what evidence is expected: “large deployment”, utility ~ company use*



Think about how PC thinks about your claims!

Reviewer (and author!) claims check-list

- **Claims are precise and explicit (no implicit claims)**

- *What are you promising and not promising?*
- *“Performance” means different things to different sys researchers.*

- **Claims match the problem and design**

- *Does the claim make sense in this context?*
- *Reviewers associate certain claims with certain contexts: mobile and IoT ~ energy claims*

- **Claims are interesting and non-trivial**

- *Will I learn something if I find out the answer to this claim?*
- *Depends on what reviewers already know and what they care about!*

- **Claims do not over-promise**

- *Will any amount of evidence convince me of this claim?*
- *Perceptions of strength of claim and what evidence is expected: “large deployment”, utility ~ company use*

In writing:

Claims are frequently presented as *contributions* or *research questions*



Spotting evidence “crimes”

Many evaluation evidence mistakes to avoid

- Selective benchmarking
- Improper handling of benchmark results
- Using the wrong benchmarks
- Improper comparison of benchmark results
- Missing crucial information

[1] Systems Benchmarking Crimes by Gernot Heiser
<https://www.cse.unsw.edu.au/~gernot/benchmarking-crimes.html>



Spotting evidence “crimes”

Many evaluation evidence mistakes to avoid

- **Selective benchmarking**
- Improper handling of benchmark results
- Using the wrong benchmarks
- **Improper comparison of benchmark results**
- Missing crucial information

[1] Systems Benchmarking Crimes by Gernot Heiser
<https://www.cse.unsw.edu.au/~gernot/benchmarking-crimes.html>



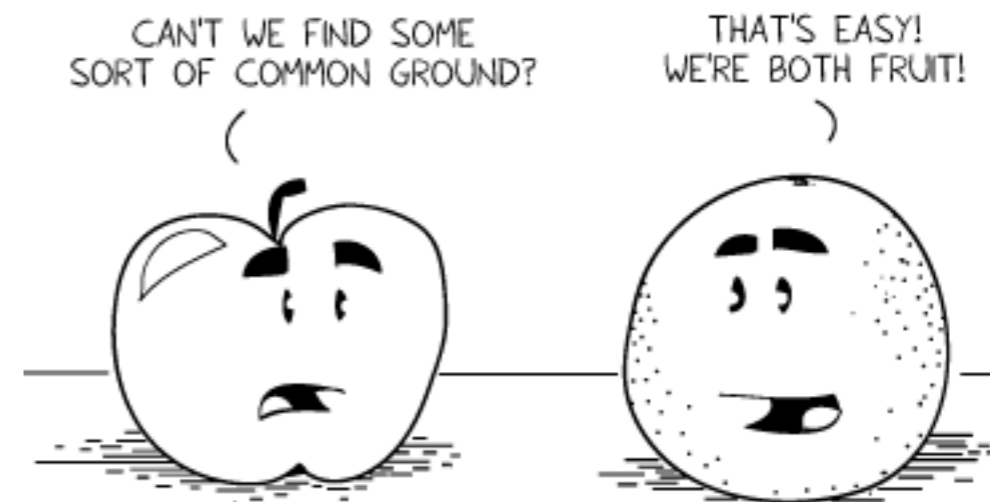
Selective benchmarking: What are you not showing me?

- You can't evaluate your system on all possible scenarios
- But you must convince reviewers that your eval covers sufficiently many scenarios [generalization]
 - Aim for *meaningful* diversity
 - Explain *why* the diversity you have is sufficient
 - Be careful with what you claim

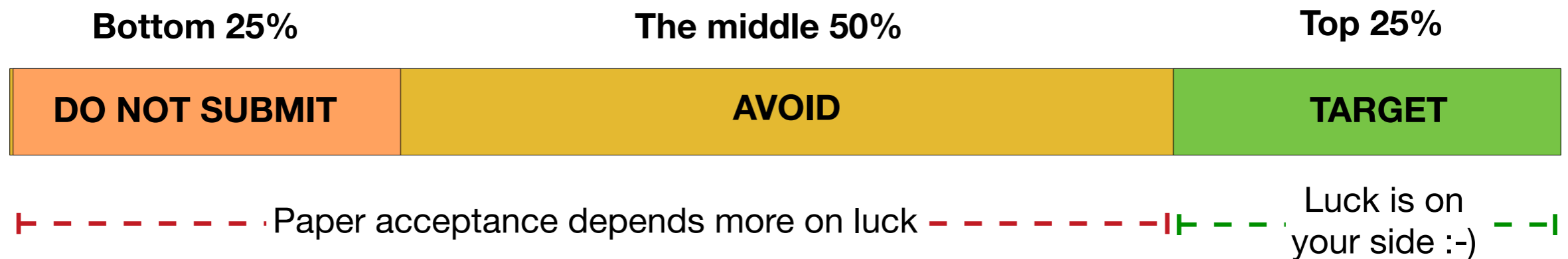


Improper benchmark comparison: Can I trust what you are showing me?

- **Select baseline carefully**, avoid non-baseline comparisons
 - Existing state of the art solution: previous year's paper
 - Optimal (or theoretically best) solution: assume zero soft overhead
- **Aim for accepted standard that others trust**
 - Avoid comparing to your paper from last year
 - Re-use (widely available) benchmarks from previous work
- **Instantiate competitor system fairly**
 - Did you configure your competitor with same care as your own system?
 - Ask competing system's authors for advice!



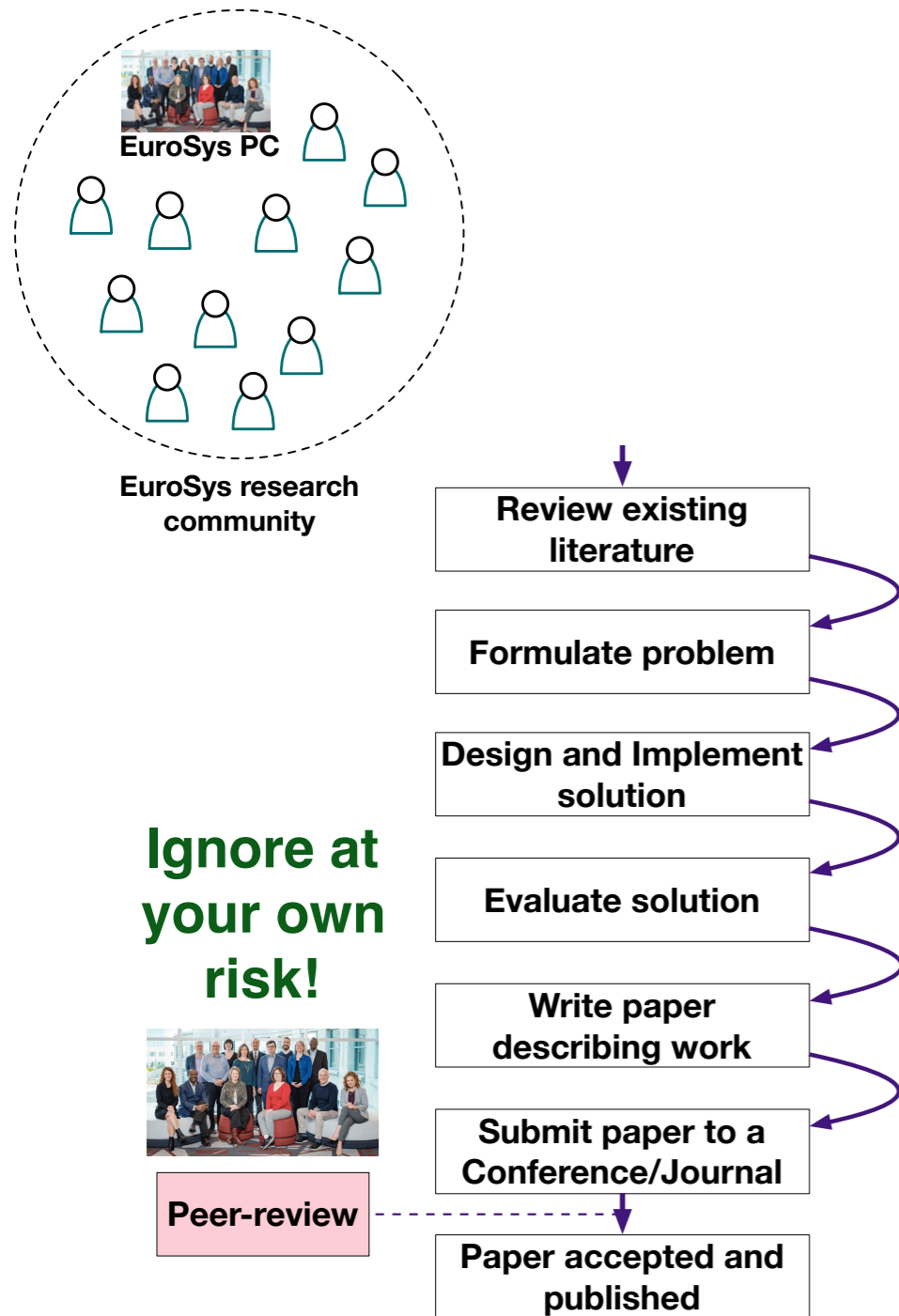
Caution: peer-review is fairly random!



- There is evidence that peer-review is more random for papers outside of the top/bottom 25%
- Strategy: if your paper is not in the top 25%, don't submit
 - Consult your program committee
 - Consult yourself (and be honest with yourself)

[1] Conference Reviewing Considered Harmful, Tom Anderson, OSR April 2009
<https://homes.cs.washington.edu/~tom/support/confreview.pdf>

Successfully publishing your research



- Consult *your personal PC* during research
- Read many (good) papers (learn norms)
- Reflect on your research community
- Reviewers as proxies for your community
- Match claims to evidence
- Think positively in the face of rejection

Don't be afraid to modify your research process

Keep peer-review in mind