

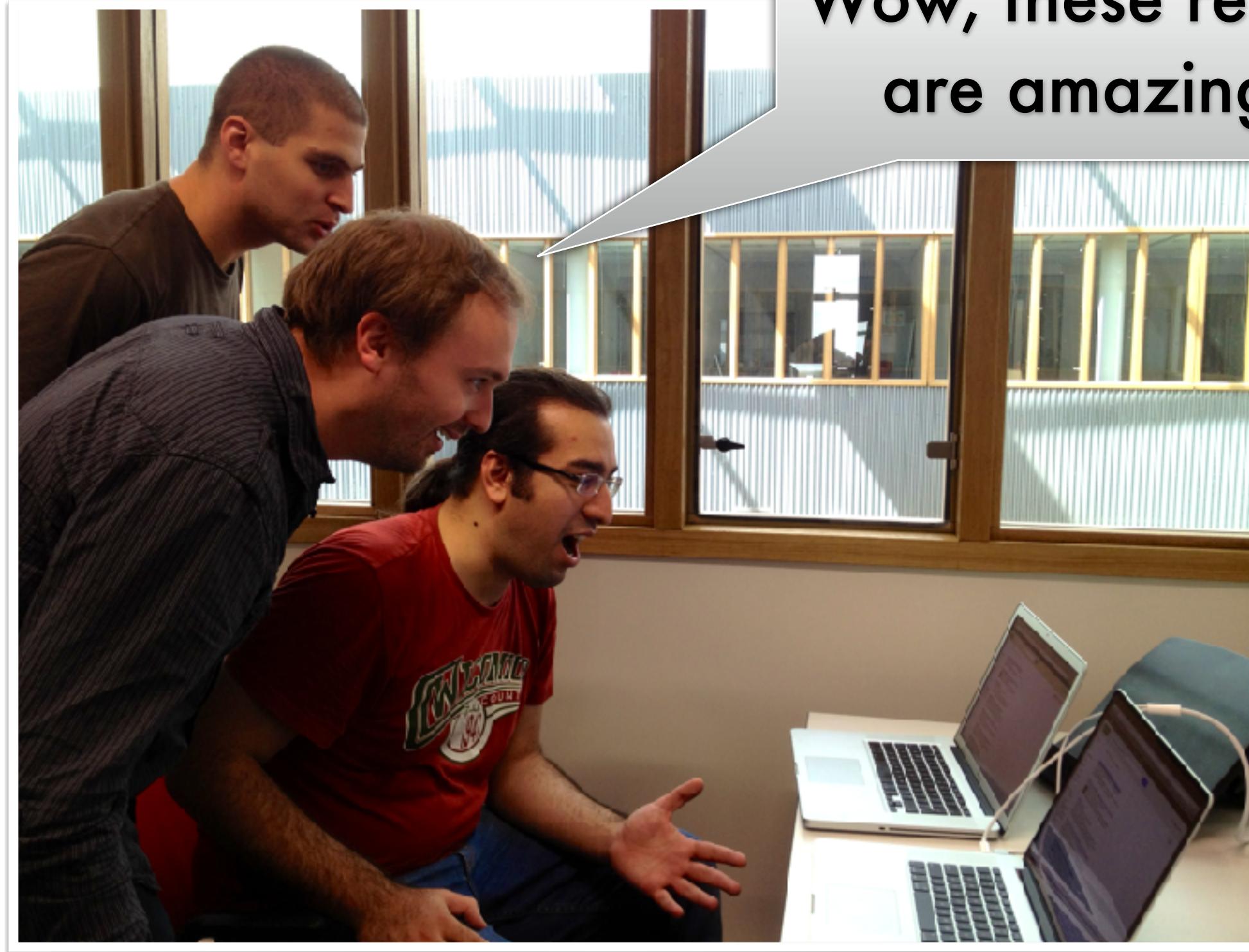
Studying People in Software Engineering Research: We're There, But Let's Go Back Again

Bogdan Vasilescu
@b_vasilescu

In an atrium-facing office in Metaforum there lived
a PhD student ...



In an atrium-facing office in Metaforum there lived a PhD student ...



In an atrium-facing office in Metaforum there lived a PhD student ...



Dear Andrea, Alexander and Bogdan:

Thank you for your recent submission to ICSE 2013 Technical Research Track.

After review from both of the PC Co-Chairs, we have determined that your paper will not undergo further review by the Program Committee due to the scope of the paper not fitting within the Technical Research track of ICSE.

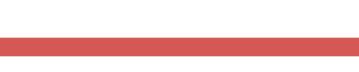
...

Best Regards,

ICSE PC Co-Chairs



Fast forward 10 years



Human aspects of Software Engineering take up considerable space in the ICSE program

Entire sessions on:

- “Developers' behaviors”
- “Studies on gender in SE”
- “Diversity and inclusion in SE”
- ...



We're there.

We got here through: visionaries



Alexander Serebrenik photo by Marike van Pagée
<https://www.agconnect.nl/artikel/superscience-betere-software-door-inclusieve-teams>

We got here through: empiricism

A circular portrait of Victor Basili, a man with white hair and a beard, wearing glasses and a black turtleneck, is positioned on the left. To his right, the word "Awards" is written in large, bold, orange letters. Below the portrait, the text "IEEE TCSE Lifetime Achievement Award" is displayed in a dark font. Underneath that, the name "Victor Basili" is written in a larger, bold, orange font. A detailed description of the award follows: "in recognition of formalizing the concepts around measurement programs and process improvement to support decision making in software development organizations, and founding the field of Empirical Software Engineering." At the bottom of the slide, there is a logo for "ICSE 2023" featuring a stylized kangaroo and cityscape, along with the text "45th International Conference on Software Engineering" and "Melbourne Convention and Exhibition Centre 14-20 May 2023". The bottom left corner contains the hashtag "#icse2023".

We've seen an explosion of data (and access to data) about people in SE the last 10 years

Let's build from here

The complete developer platform to build, scale, and deliver secure software.

100+ million Developers 4+ million Organizations 330+ million Repositories 90% Fortune 100

Follow us on [Twitter](#) [Facebook](#) [YouTube](#) [LinkedIn](#)



GHTorrent: Github's Data from a Firehose

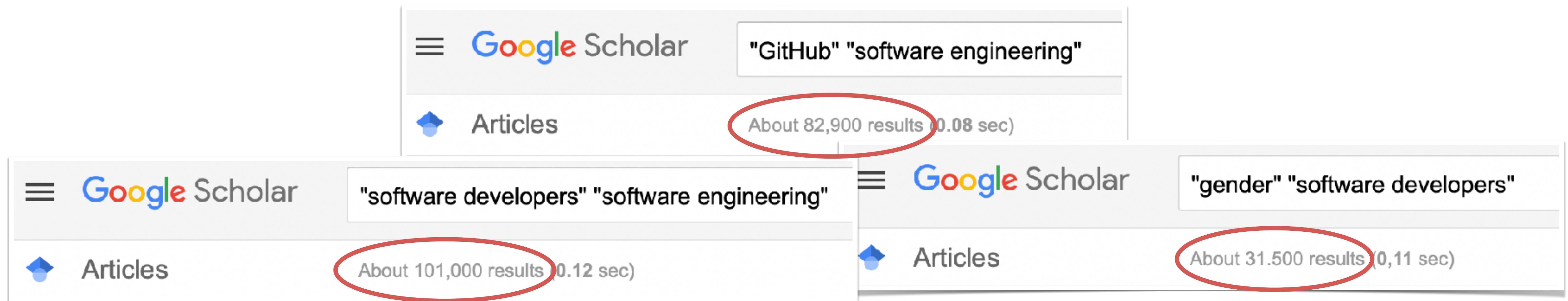
Georgios Gousios and Diomidis Spinellis
Department of Management Science and Technology



GHTorrent received the foundational contribution award at MSR 2018! #msr18

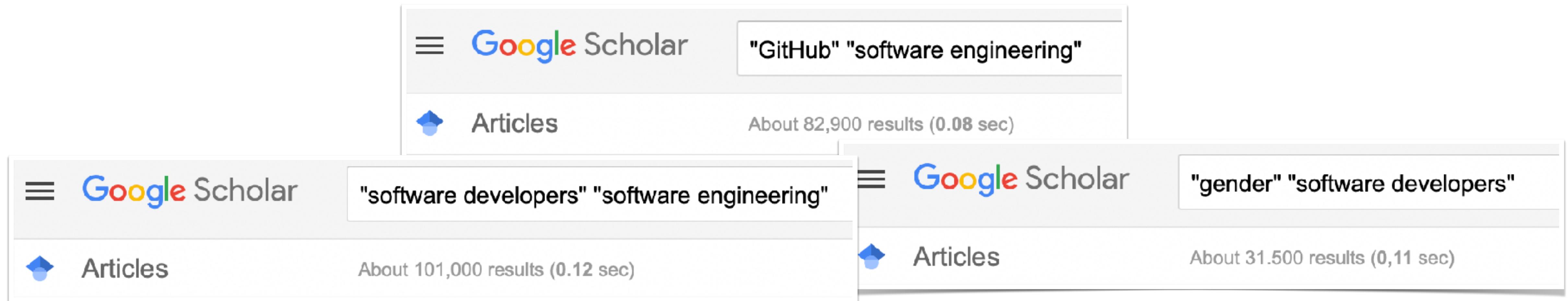


Pro: It is relatively easy to study people in Software Engineering



With great power comes great responsibility

Con Pro: It is relatively easy to study people in Software Engineering



Let's go back again

to disciplines and research methods where people are more of a tradition.

Aside: People have always been a topic in SE research

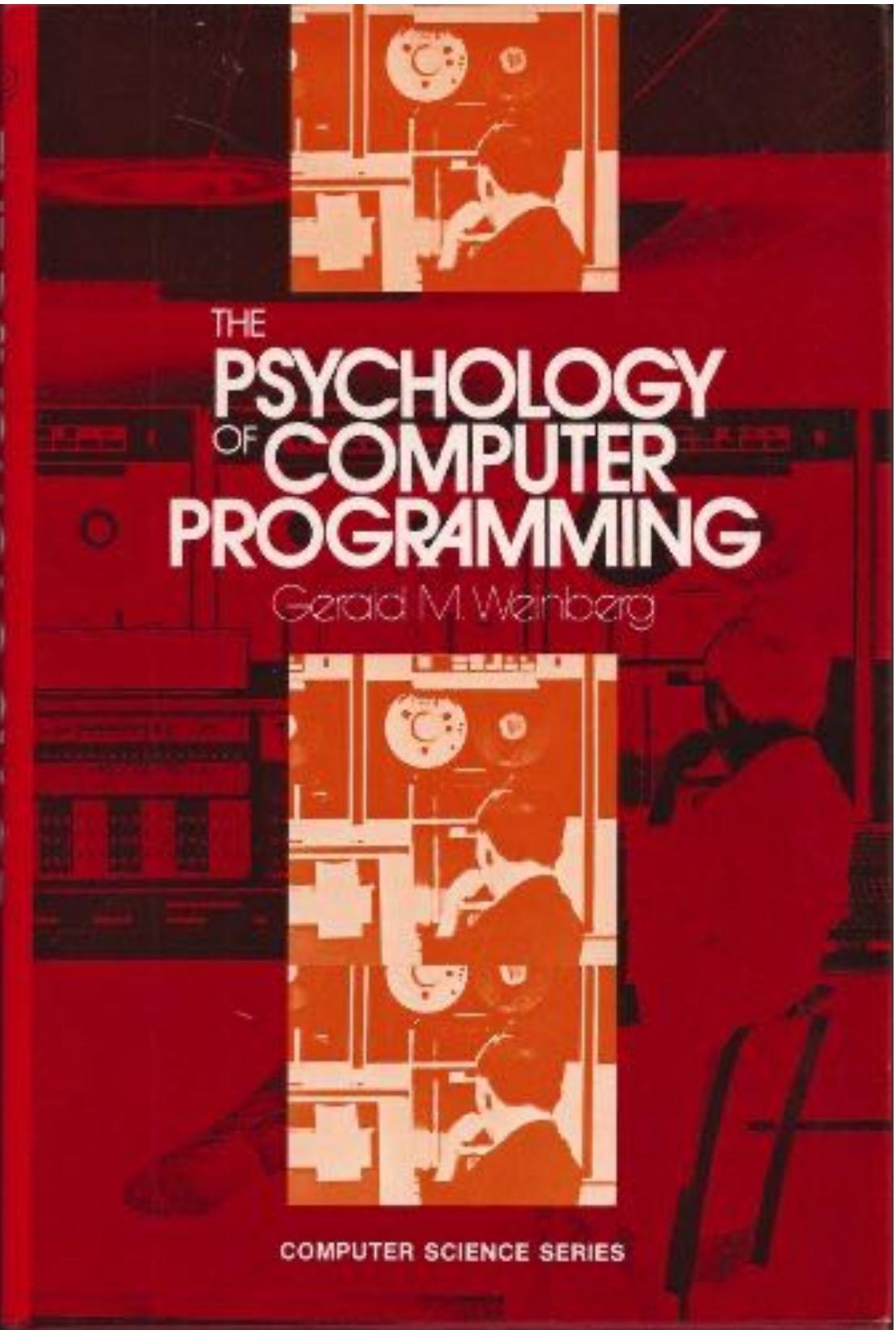
HOW DO COMMITTEES INVENT?

by MELVIN E. CONWAY

design organization criteria

“Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.” (Conway, 1968)

Aside: People have always been a topic in SE research



Technological issues are subordinate to human issues (Weinberg, 1971).

Aside: People have always been a topic in SE research

THE MYTHICAL MAN-MONTH

The above is an extract from the *Mythical Man-Month* by Frederick P. Brooks, Jr., copyright © 1975 by Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, pages 14-26, 88-94, 153-160, and 177. Reprinted with permission. The extract was prepared by the editors of *Datamation* and published in *Datamation*, December 1974, pages 44-52.

HOW DOES A PROJECT GET TO
BE A YEAR LATE?
ONE DAY AT A TIME.

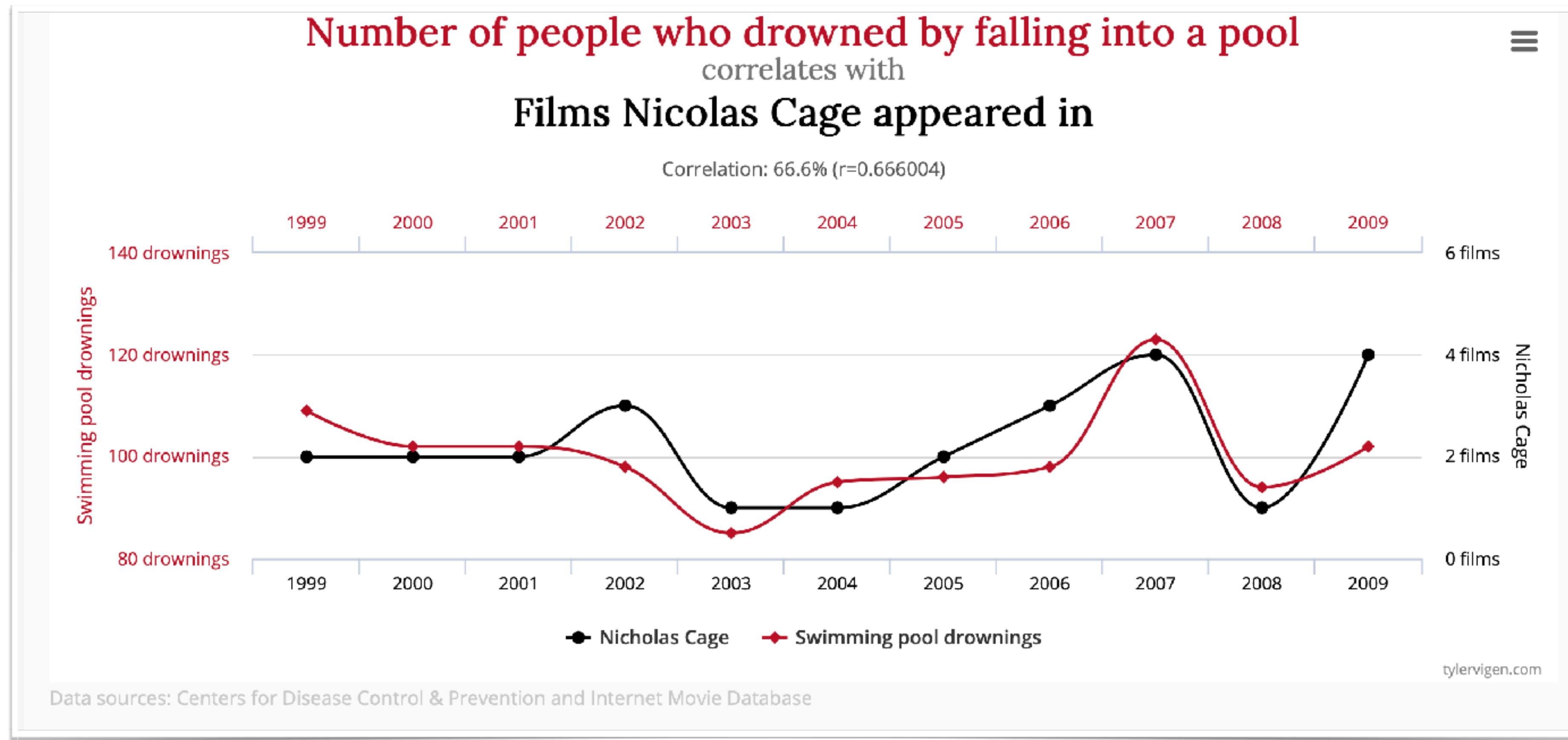
By Frederick P. Brooks, Jr.

Human aspects of software development have an important influence on software productivity (Brooks, 1975).

Where we are now in SE

3 examples

1. We're beginning to distinguish correlation from causation



<http://www.tylervigen.com/spurious-correlations>

Example: Do tweets *cause* GitHub stars?

The image shows a composite of two screenshots. The top part is a Twitter post from Max Woolf (@minimaxir). The post includes a profile picture, the author's name and handle, and the tweet text: "I just released my new Python package: simpleaichat, an open-source tool for working with ChatGPT/GPT-4 with minimal code yet max flexibility! I built simpleaichat out of sheer frustration with LangChain and aim to make it the easiest way to make AI apps." Below this is a GitHub project page for "minimaxir/simpleaichat". The page shows the repository name, a profile picture of the owner, a description ("Python package for easily interfacing with chat apps, with robust features and minimal code complexity."), and metrics: 3 contributors, 1 used by, 549 stars, and 22 forks. A GitHub icon is also present. At the bottom, there is a snippet of the GitHub page content: "github.com GitHub - minimaxir/simpleaichat: Python package for easily interfacing with chat apps, with robust features and minimal code complexity. - GitHub - ...". The bottom part of the image shows the timestamp "5:24 PM · Jun 8, 2023" and interaction buttons for likes, replies, and shares, along with a link to "Read 18 replies".

Fang, H., Lamba, H., Herbsleb, J., & Vasilescu, B. (2022). "This is damn slick!" estimating the impact of tweets on open source project popularity and new contributors. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 2116-2129).

Example: Do tweets *cause* GitHub stars?

Max Woolf (@minimaxir · Follow) posted a tweet on June 8, 2023, at 5:24 PM. The tweet reads: "I just released my new Python package: simpleaichat, an open-source tool for working with ChatGPT/GPT-4 with minimal code yet max flexibility! I built simpleaichat out of sheer frustration with LangChain and aim to make it the easiest way to make AI apps." The tweet has 737 likes, 1 reply, and 18 shares.

The tweet links to a GitHub repository page for "minimaxir/simpleaichat". The repository page shows the following details:

- Name:** minimaxir/simpleaichat
- Description:** Python package for easily interfacing with chat apps, with robust features and minimal code complexity.
- Contributors:** 3
- Used by:** 1
- Stars:** 549
- Forks:** 22

The GitHub page also displays the repository's history, showing recent commits from the user "minimaxir".

The GitHub repository page for "minimaxir/simpleaichat" shows the following details:

- Repository:** minimaxir/simpleaichat (Public)
- Code tab:** Selected
- Issues:** 7
- Pull requests:** 1
- Actions:**
- Projects:**
- Security:**
- Insights:**

The main branch is "main".

Commits:

- minimaxir notes on model param ... 2 days ago (75)
- .github GitHub sponsorship last week
- docs README images last week
- examples add exquisite corpse example 5 days ago
- simpleaichat add AIChat.session() as context manager 5 days ago
- .gitignore working packahe 2 weeks ago
- LICENSE initial last month
- PROMPTS.md last minute README tweaks last week

About:

Python package for easily interfacing with chat apps, with robust features and minimal code complexity.

#ai #chatgpt

Readme

MIT license

559 stars (circled in red)

11 watching

22 forks

Report repository

Fang, H., Lamba, H., Herbsleb, J., & Vasilescu, B. (2022). "This is damn slick!" estimating the impact of tweets on open source project popularity and new contributors. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 2116-2129).

Example: Do tweets *cause* GitHub stars?

Max Woolf (@minimaxir · Follow)
I just released my new Python package: simpleaichat, an open-source tool for working with ChatGPT/GPT-4 with minimal code yet max flexibility!
I built simpleaichat out of sheer frustration with LangChain and aim to make it the easiest way to make AI apps.

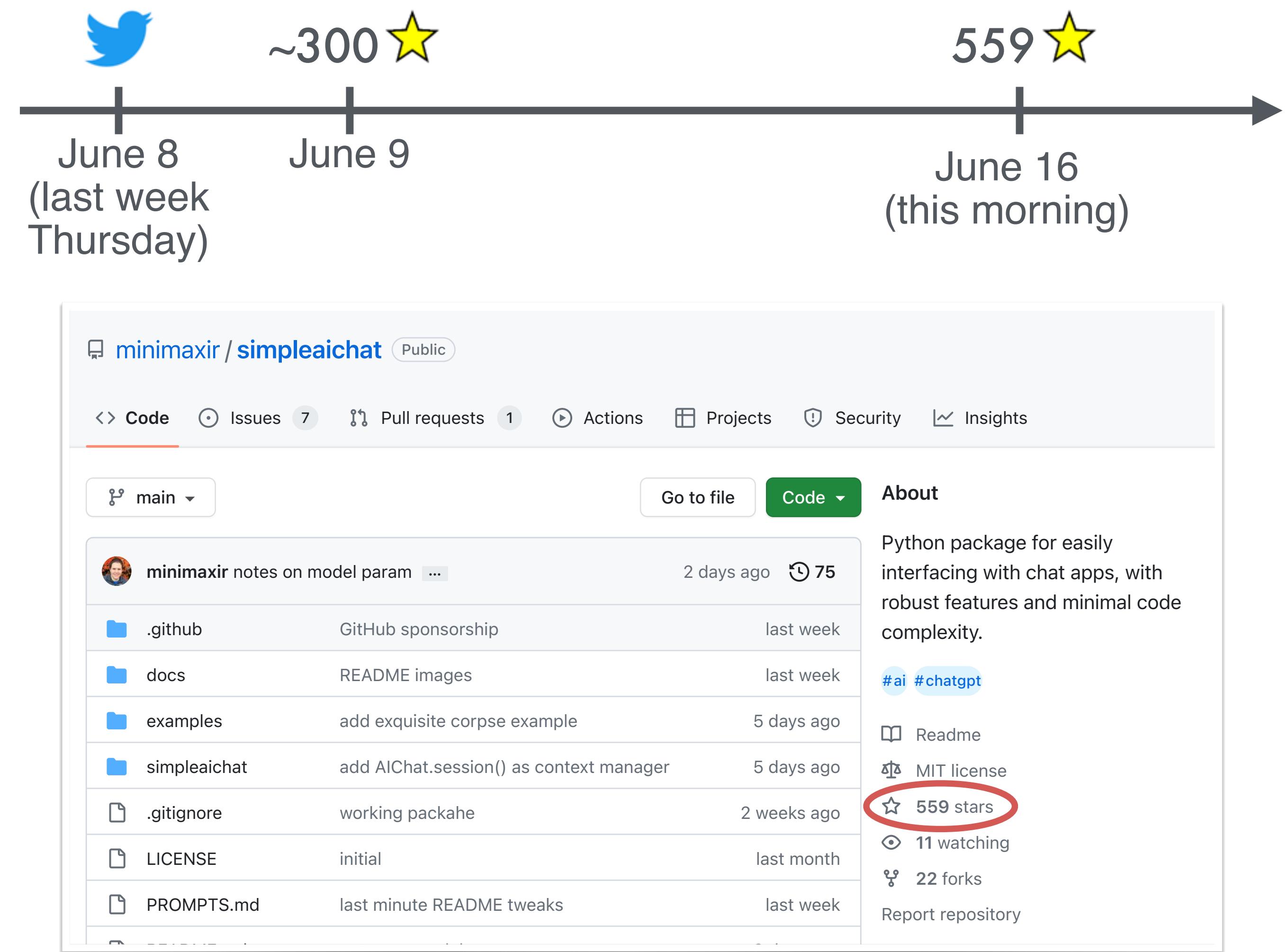
minimaxir/simpleaichat

Python package for easily interfacing with chat apps, with robust features and minimal code complexity.
Contributors: 3 | Used by: 1 | Stars: 549 | Forks: 22

github.com
GitHub - minimaxir/simpleaichat: Python package for easily interfacing with chat apps, with robust features and minimal code complexity. - GitHub - ...

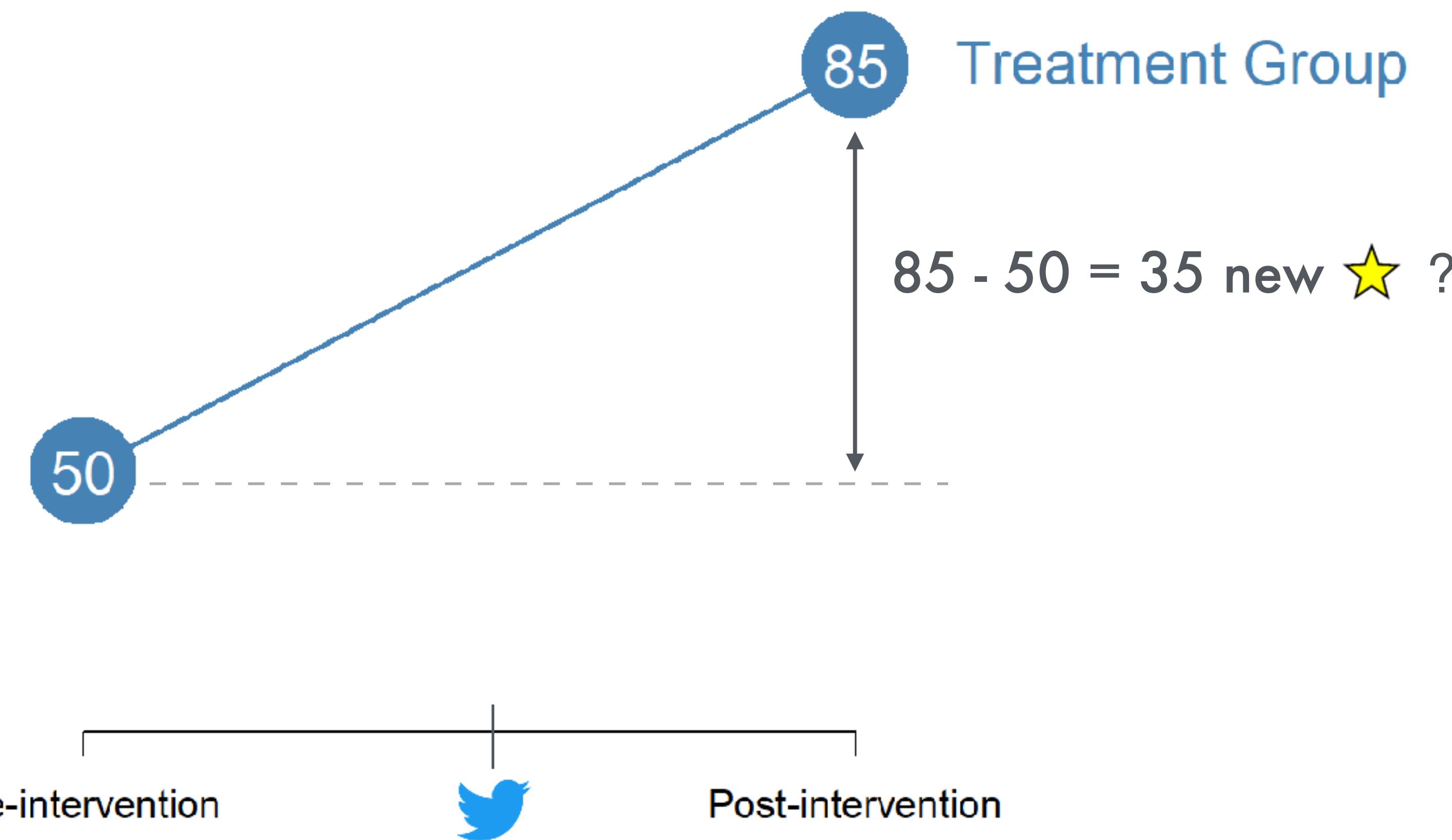
5:24 PM · Jun 8, 2023

737 replies | Share | Read 18 replies



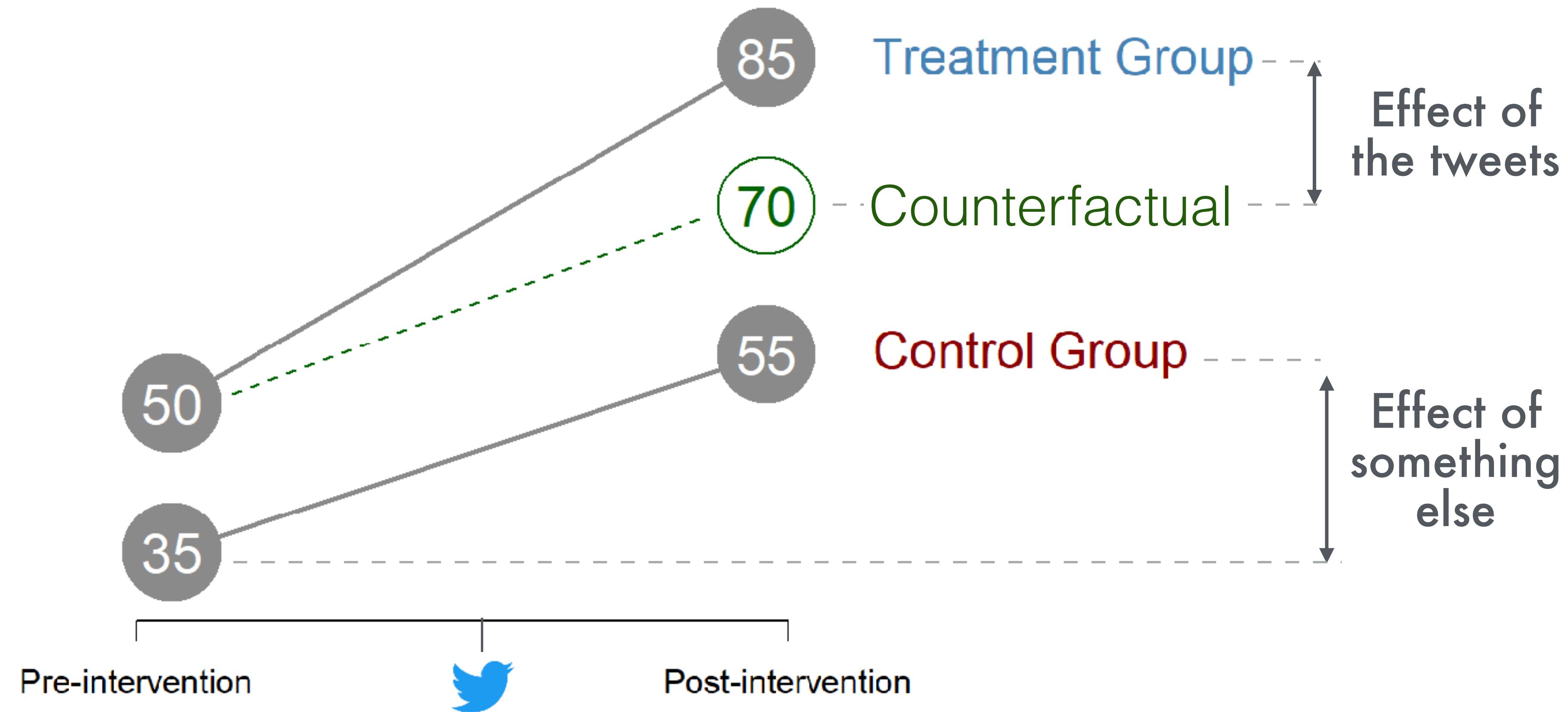
Fang, H., Lamba, H., Herbsleb, J., & Vasilescu, B. (2022). "This is damn slick!" estimating the impact of tweets on open source project popularity and new contributors. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 2116-2129).

Idea: Measure how much a group mean changes before and after an intervention



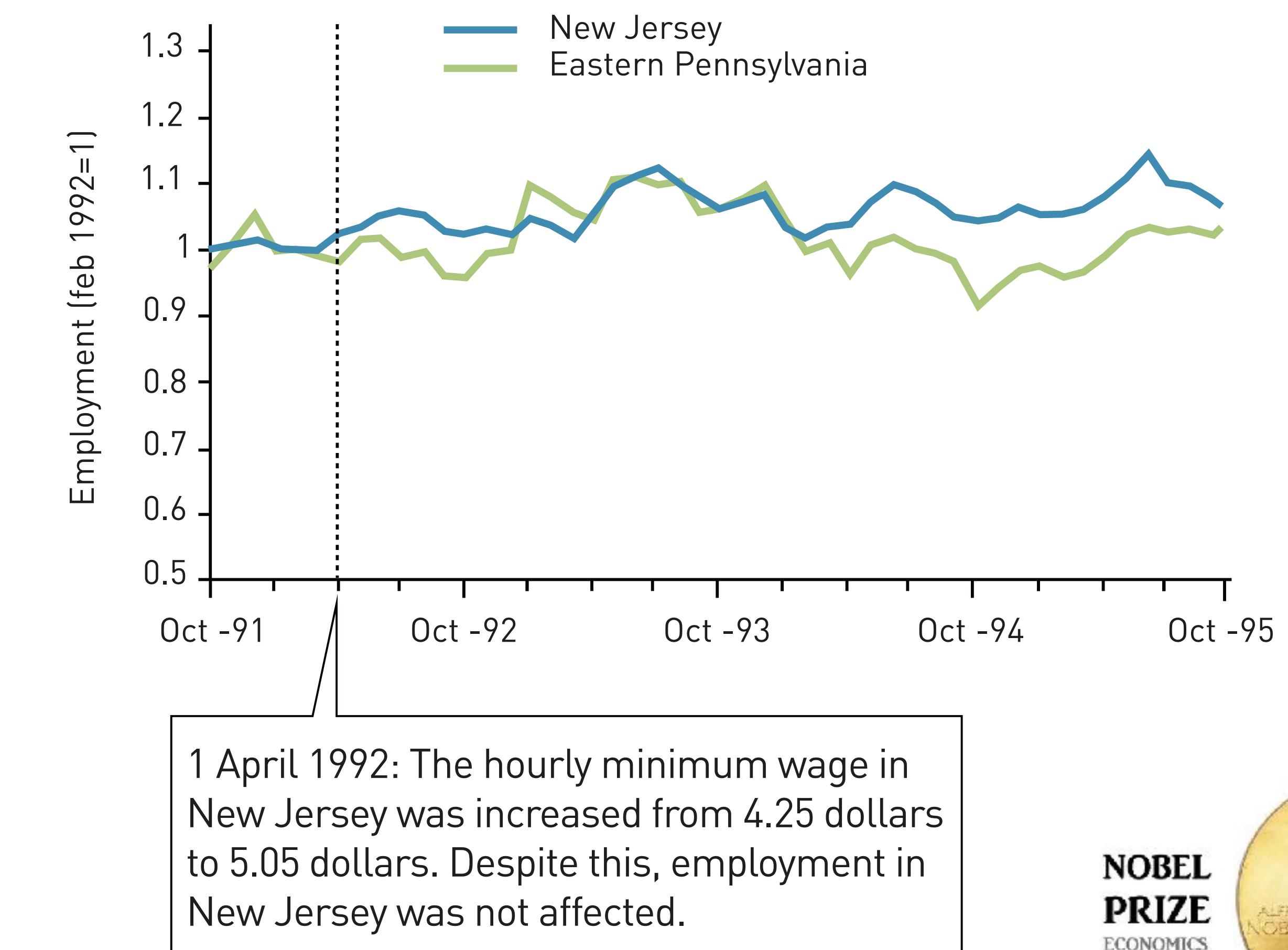
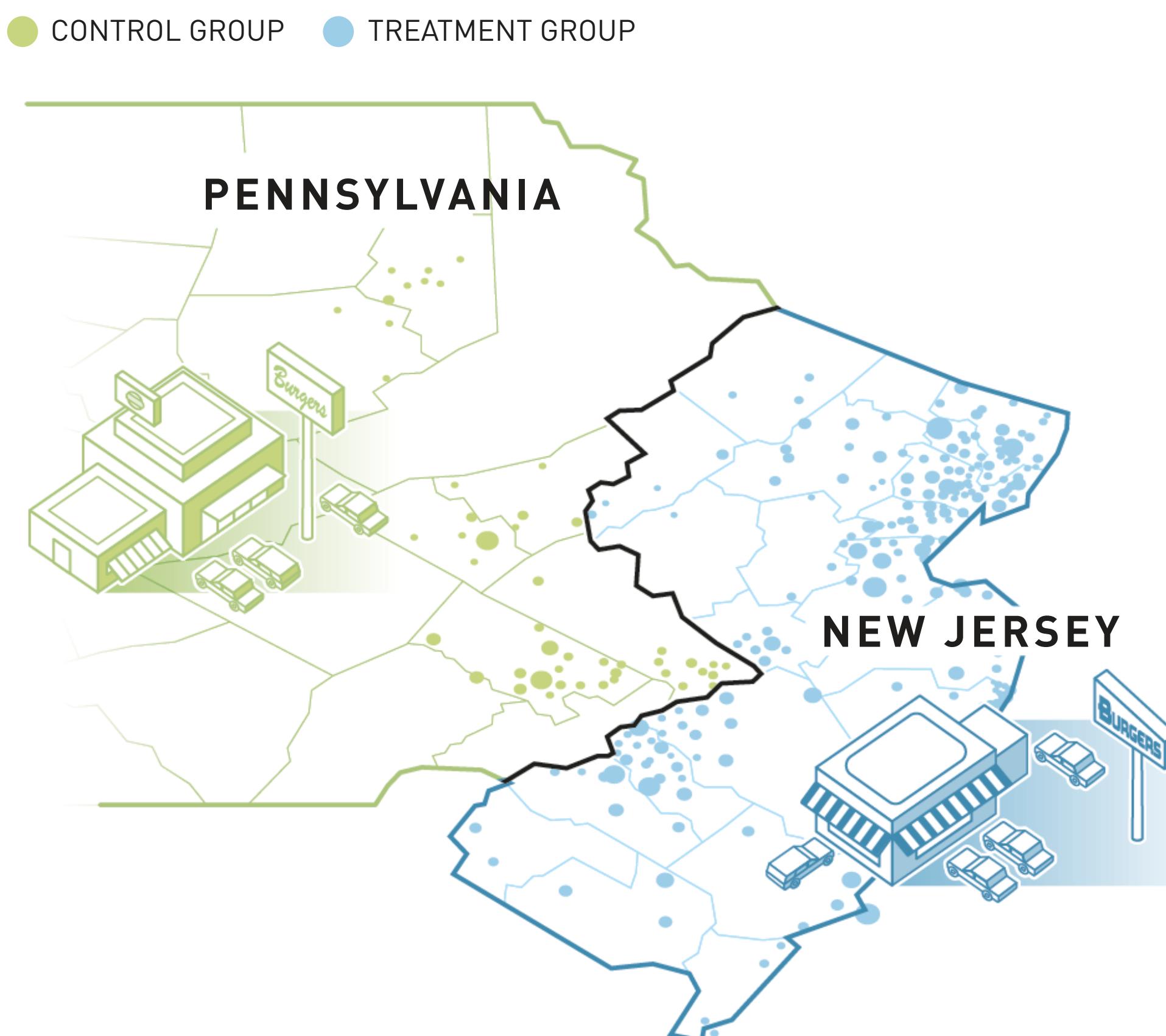
Fang, H., Lamba, H., Herbsleb, J., & Vasilescu, B. (2022). "This is damn slick!" estimating the impact of tweets on open source project popularity and new contributors. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 2116-2129).

Better idea: Compare that change to the change in an appropriate control group



Fang, H., Lamba, H., Herbsleb, J., & Vasilescu, B. (2022). "This is damn slick!" estimating the impact of tweets on open source project popularity and new contributors. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 2116-2129).

Card and Krueger (1993) natural experiment to study how increasing the minimum wage affects employment.



NOBEL
PRIZE
ECONOMICS



<https://www.nobelprize.org/uploads/2021/10/popular-economicsciencesprize2021-2.pdf>

Are we 20 years behind on quantitative analysis methods in empirical SE?

“This Is Damn Slick!” Estimating the Impact of Tweets on Open Source Project Popularity and New Contributors

Hongbo Fang, Hemank Lamba, James Herbsleb, Bogdan Vasilescu
Carnegie Mellon University, USA
{hongbofa,hlambda,jdh,bogdanv}@cs.cmu.edu

ABSTRACT

Twitter is widely used by software developers. But how effective are tweets at promoting open source projects? How could one use Twitter to increase a project’s popularity or attract new contributors? In this paper we report on a mixed-methods empirical study of 44,544 tweets containing links to 2,370 open-source GitHub repositories, looking for evidence of causal effects of these tweets on the projects attracting new GitHub stars and contributors, as well as characterizing the high-impact tweets, the people likely being attracted by them, and how they differ from contributors attracted otherwise. Among others, we find that tweets have a statistically significant and practically sizable effect on obtaining new stars and a small average effect on attracting new contributors. The popularity, content of the tweet, as well as the identity of tweet authors all affect the scale of the attraction effect. In addition, our qualitative analysis suggests that forming an active Twitter community for an open source project plays an important role in attracting

Several mechanisms through which OSS projects can gain attention [11, 40, 68] and attract new contributors [9, 40, 53] have been studied in the past. The literature is especially rich in recent years, in the context of social coding platforms like GitHub, because of the high level of transparency and many opportunities for project maintainers to *signal*, explicitly and implicitly, about their work [20]. For example, prior studies of OSS projects hosted on GitHub have found that how projects organize their repository homepages and README files [53], whether projects get *featured* by the hosting platform [40], whether projects have public releases [10], and how maintainers use prominent repository badges to indicate less observable project qualities [68], all have an impact on how the project is perceived by its audience and even the actions that some audience members take, e.g., joining the project.

However, prior work has, by and large, focused only on *endogenous* or “in-network” attention eliciting mechanisms, i.e., taking actions or displaying signals afforded by the code hosting platform

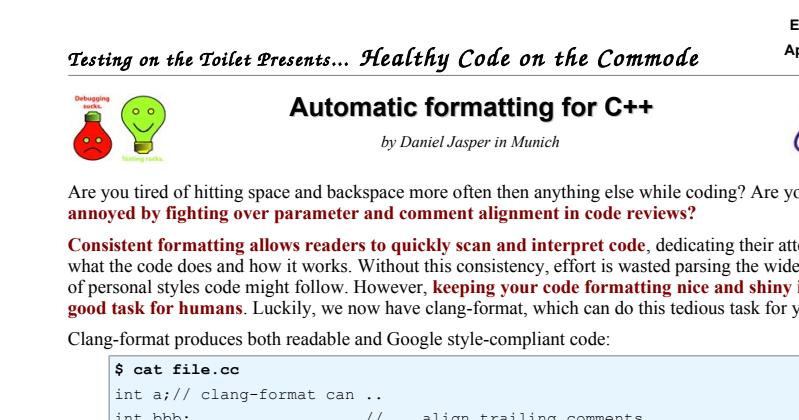
2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)

Do Developers Discover New Tools On The Toilet?

Emerson Murphy-Hill *
Google, LLC
emersonm@google.com
Edward K. Smith *
Bloomberg
tedks@riseup.net
Caitlin Sadowski
Google, LLC
supertri@google.com
Ciera Jaspan
Google, LLC
ciera@google.com
Collin Winter *
Waymo
collinwinter@waymo.com

Matthew Jorde
Google, LLC
majorde@google.com
Andrea Knight
Google, LLC
aknight@google.com
Andrew Trenk
Google, LLC
atrenk@google.com
Steve Gross
Google, LLC
stevegross@google.com

Abstract—Maintaining awareness of useful tools is a substantial challenge for developers. Physical newsletters are a simple technique to inform developers about tools. In this paper, we evaluate such a technique, called Testing on the Toilet, by performing a mixed-methods case study. We first quantitatively evaluate how effective this technique is by applying statistical causal inference over six years of data about tools used by thousands of developers. We then qualitatively contextualize these results by interviewing and surveying 382 developers, from authors to editors to readers. We found that the technique was generally effective at increasing



Difference in differences (ICSE 2022)

CausalImpact (ICSE 2019)

2. We're beginning to see macro-level phenomena and connections

Study of the (recent) past:

“We mined 10,000 GitHub projects and found a correlation between .. X .. and .. Y ..”

But: projects are assumed independent of each other.

2. We're beginning to see macro-level phenomena and connections

Study of the (recent) past:

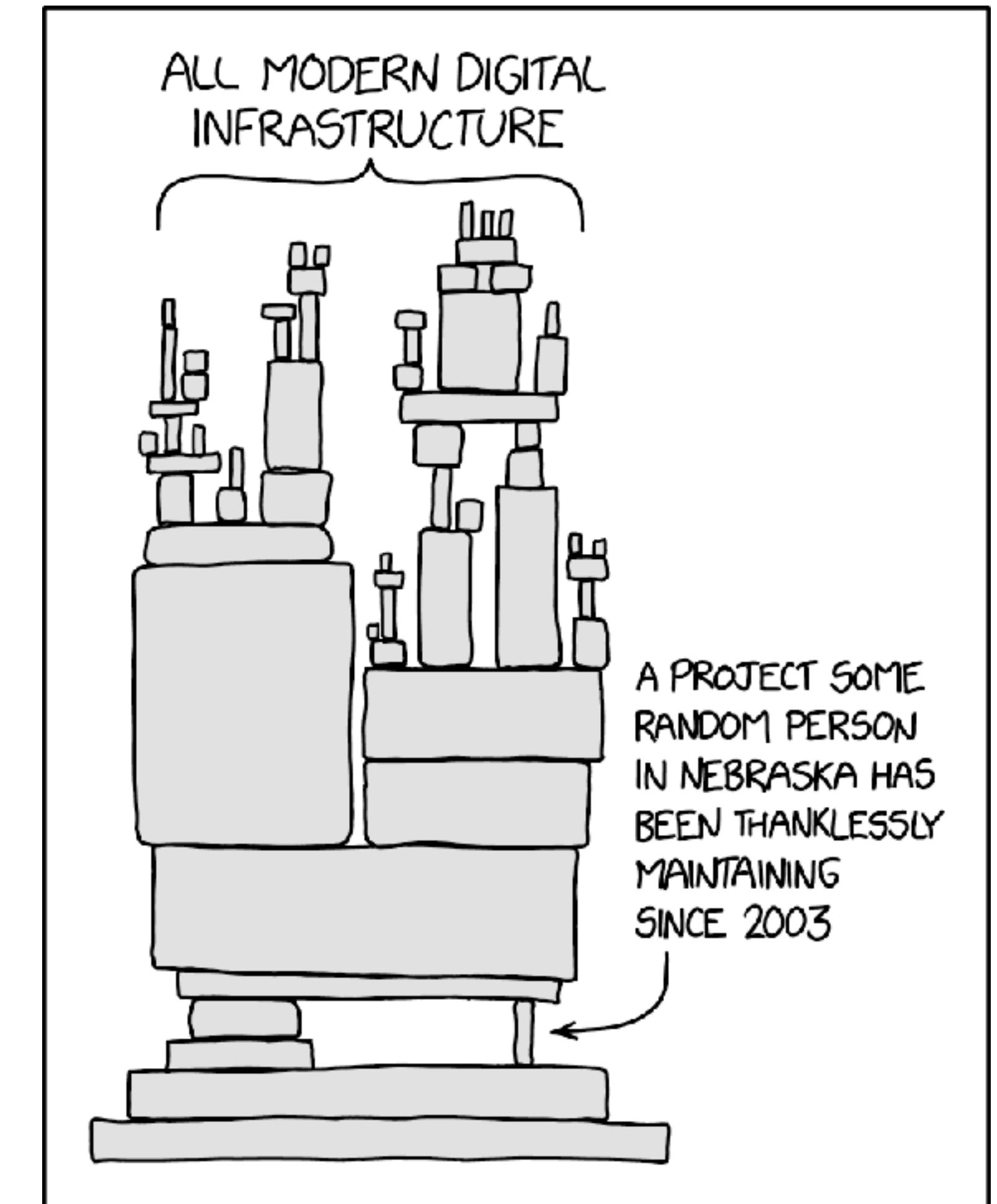
“We mined 10,000 GitHub projects and found a correlation between .. X .. and .. Y ..”

But: projects are assumed independent of each other.

Study of the present:

- Open-source supply chains
- Developer collaboration / social networks

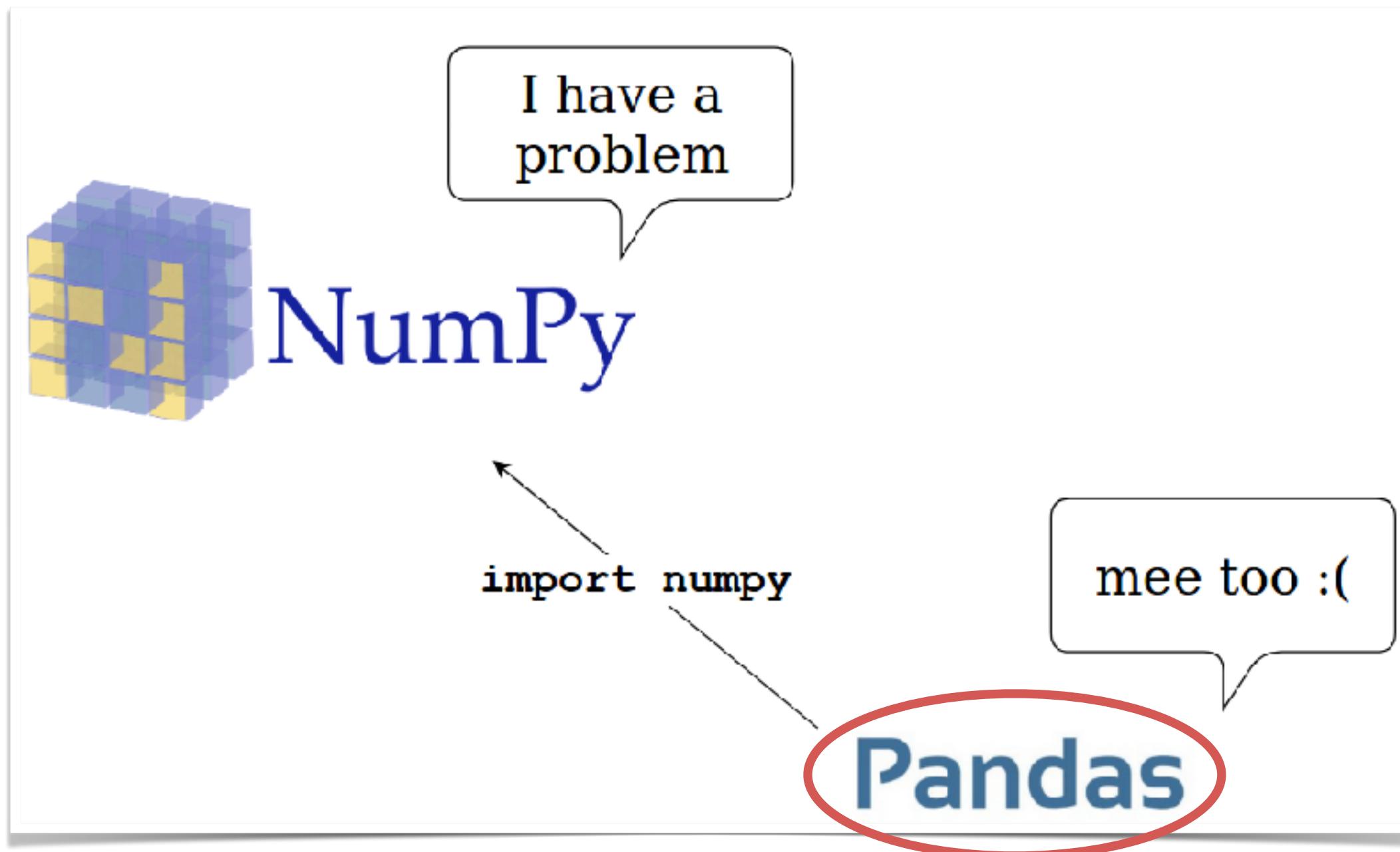
=> Ability to study network effects



<https://xkcd.com/2347/>

In the package dependency network, influence flows both up- and downstream

Survival analysis across 70k Python packages

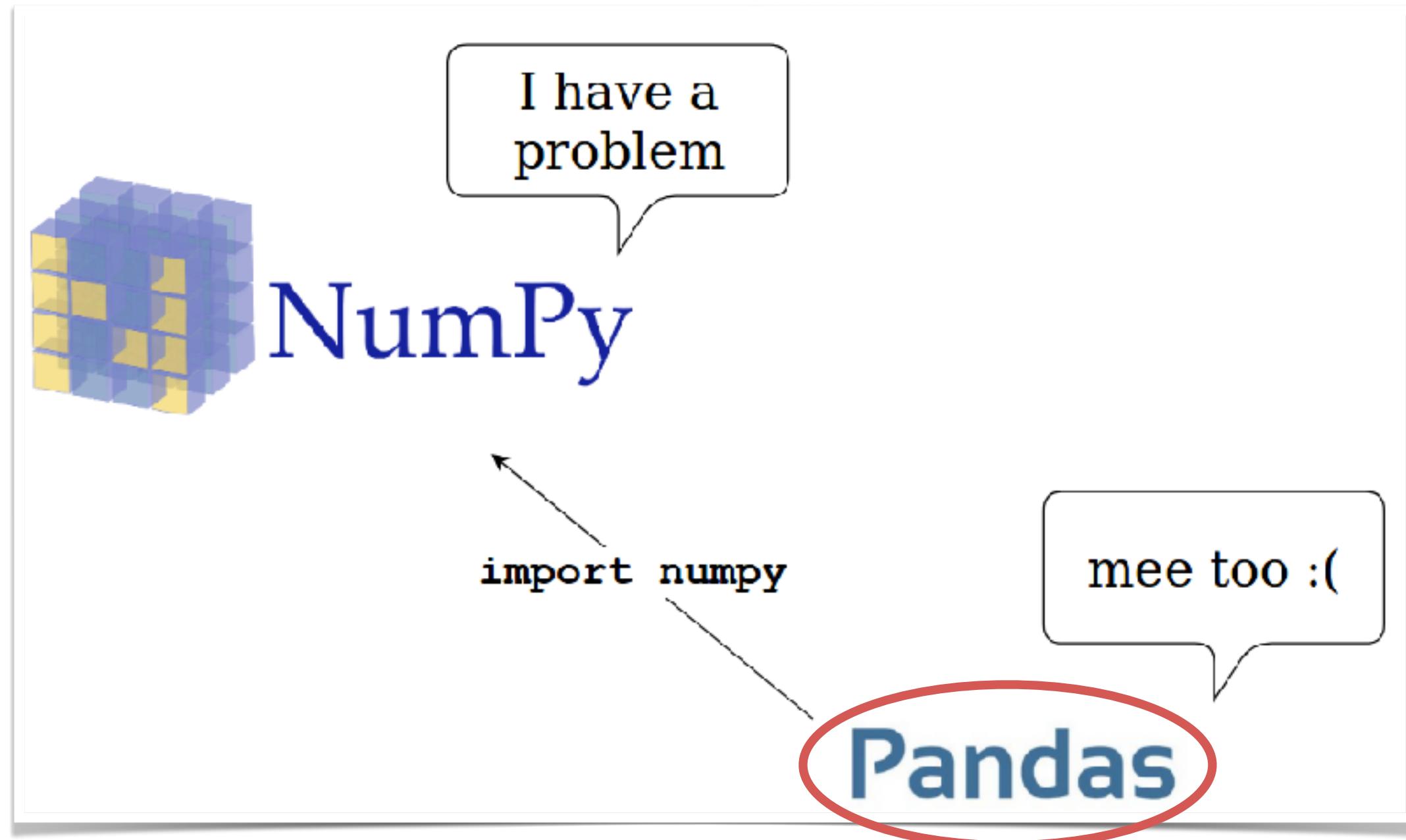


~25% **more** likely to become unmaintained,
for every extra upstream dependency

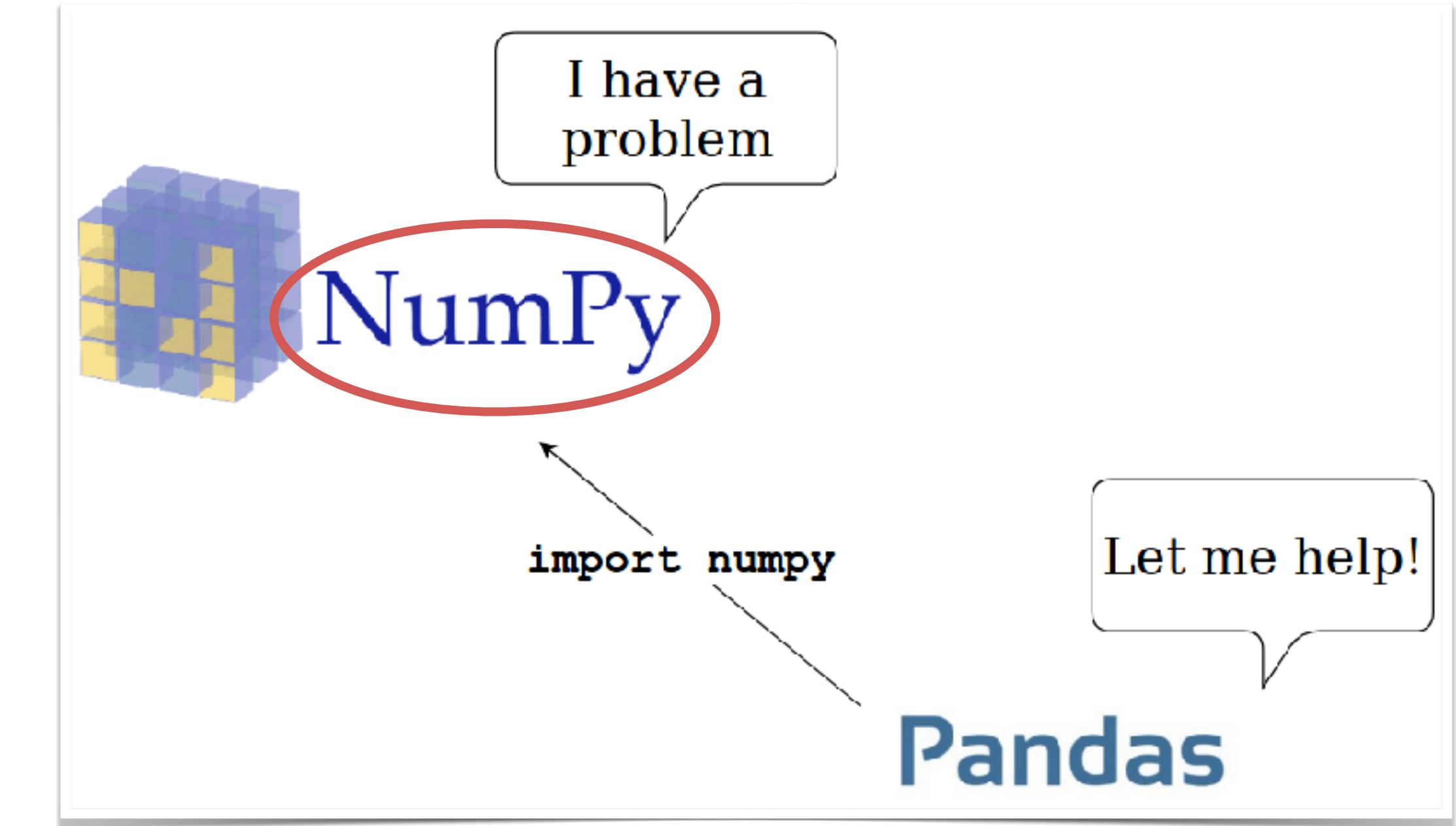
Valiev, M., Vasilescu, B., & Herbsleb, J. (2018). Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem. In Proceedings of the Joint Meeting on the Foundations of Software Engineering (pp. 644-655).

In the package dependency network, influence flows both up- and downstream

Survival analysis across 70k Python packages



~25% **more** likely to become unmaintained,
for every extra upstream dependency

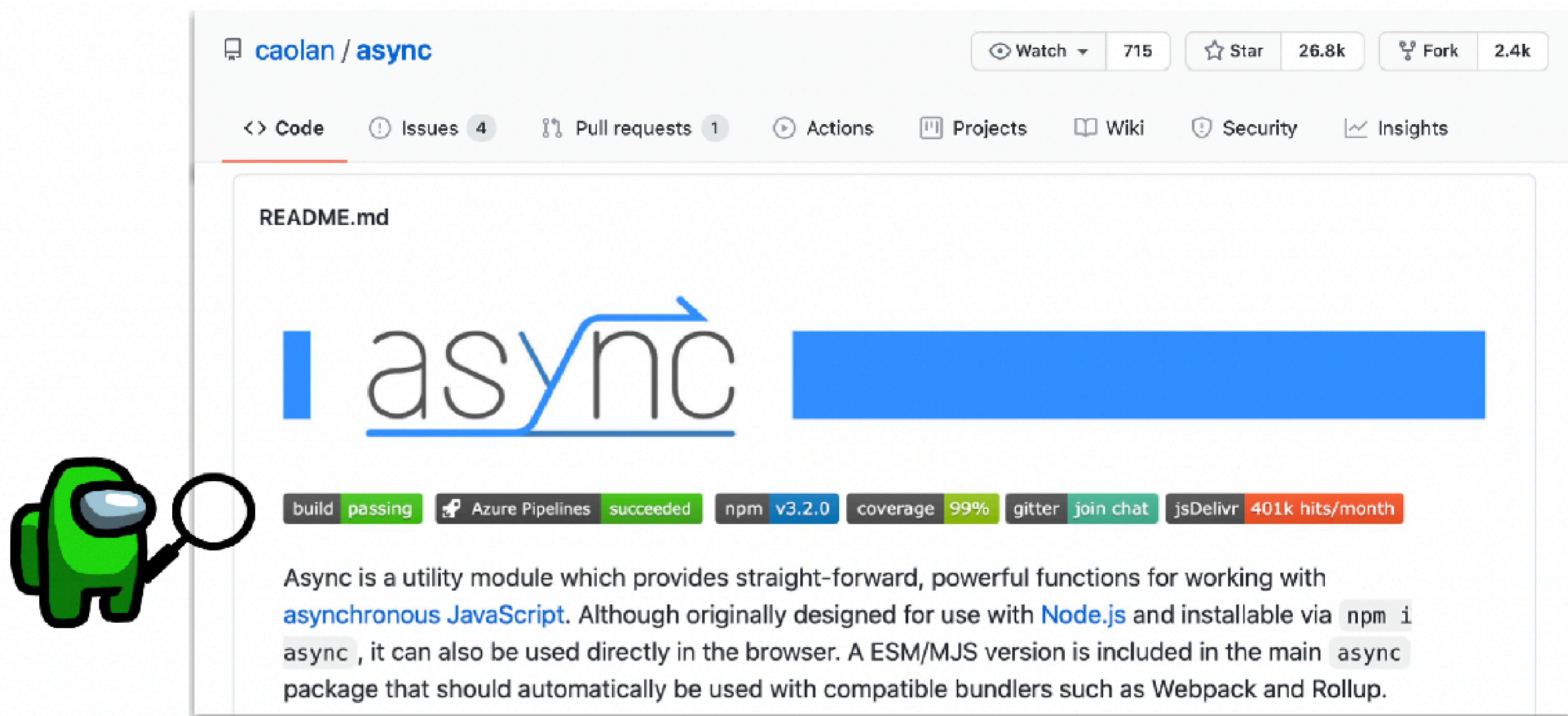


~60% **less** likely to become unmaintained,
for every extra downstream dependency

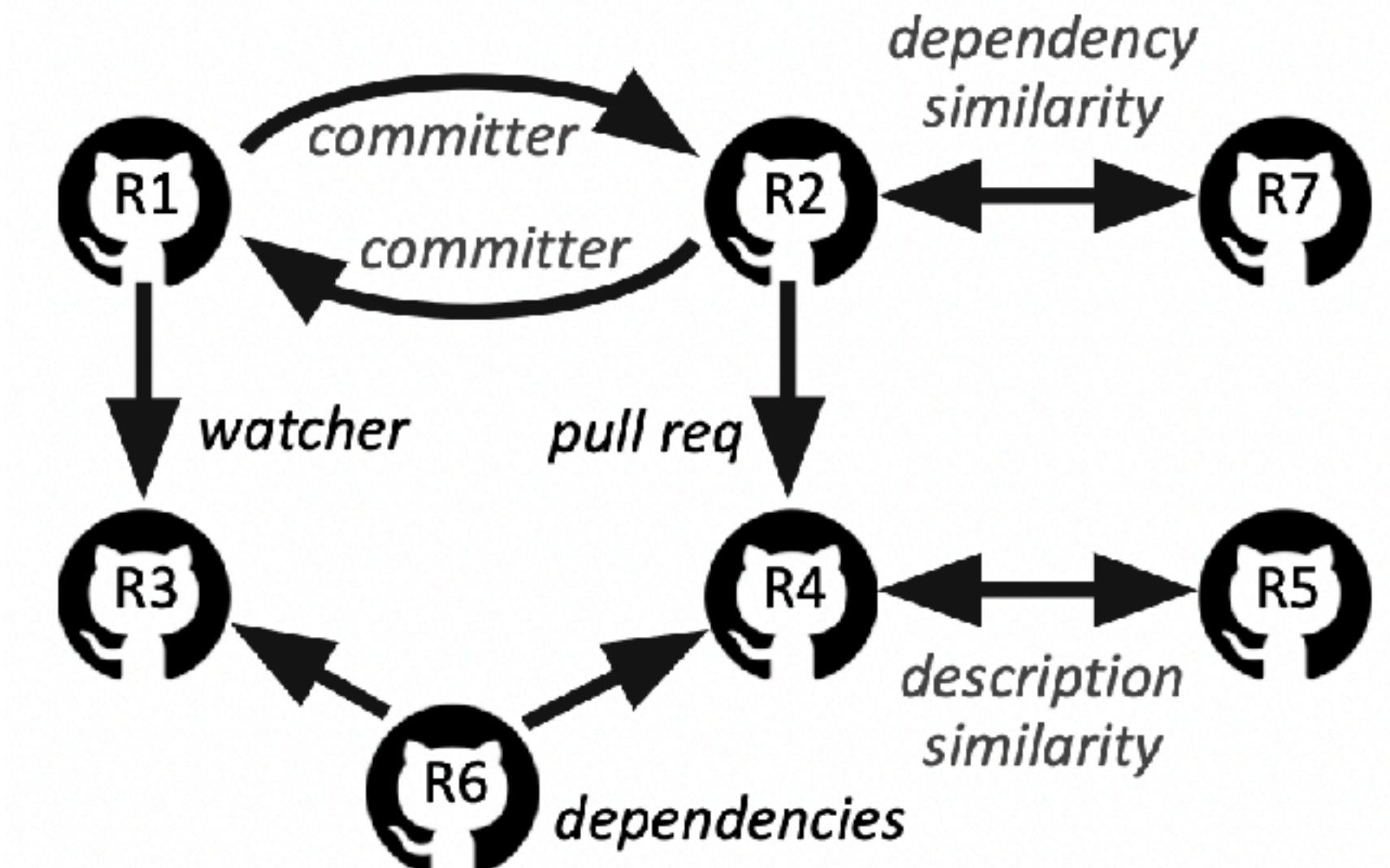
Valiev, M., Vasilescu, B., & Herbsleb, J. (2018). Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem. In Proceedings of the Joint Meeting on the Foundations of Software Engineering (pp. 644-655).

In the socio-technical network, tool adoption is proportional to the amount of exposure through network ties

Survival analysis across 86,000 npm packages

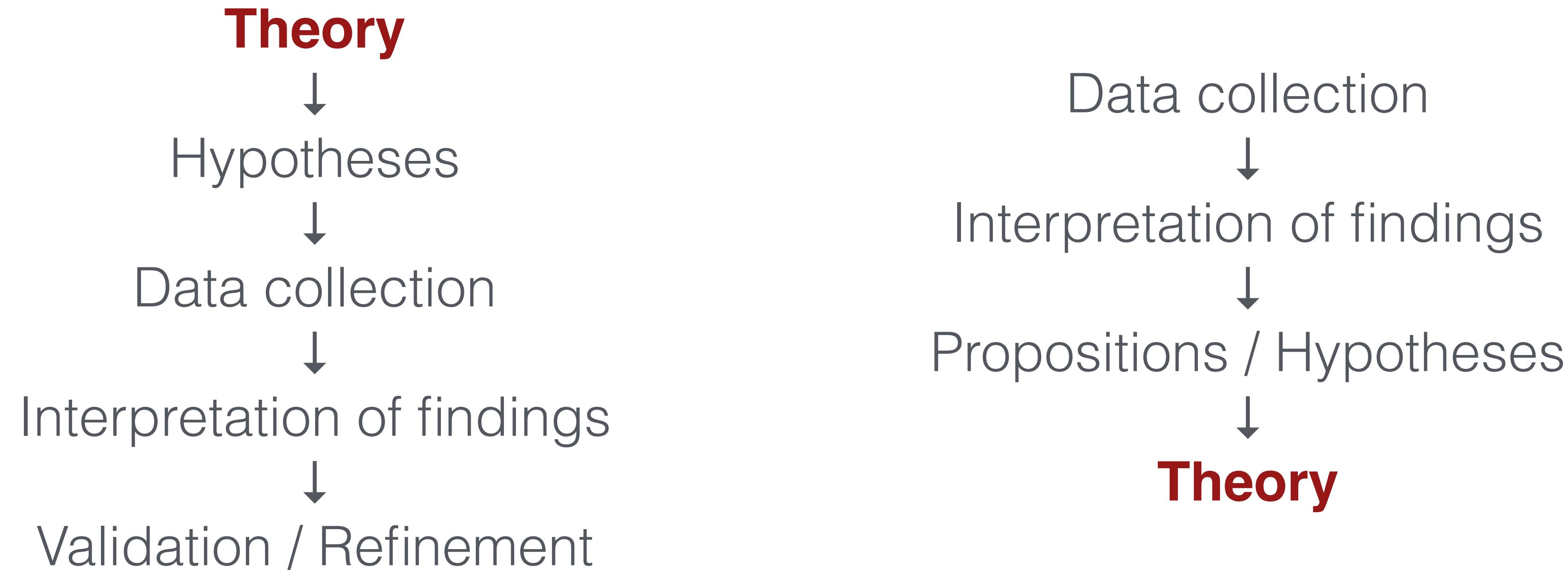


Heterogeneous network



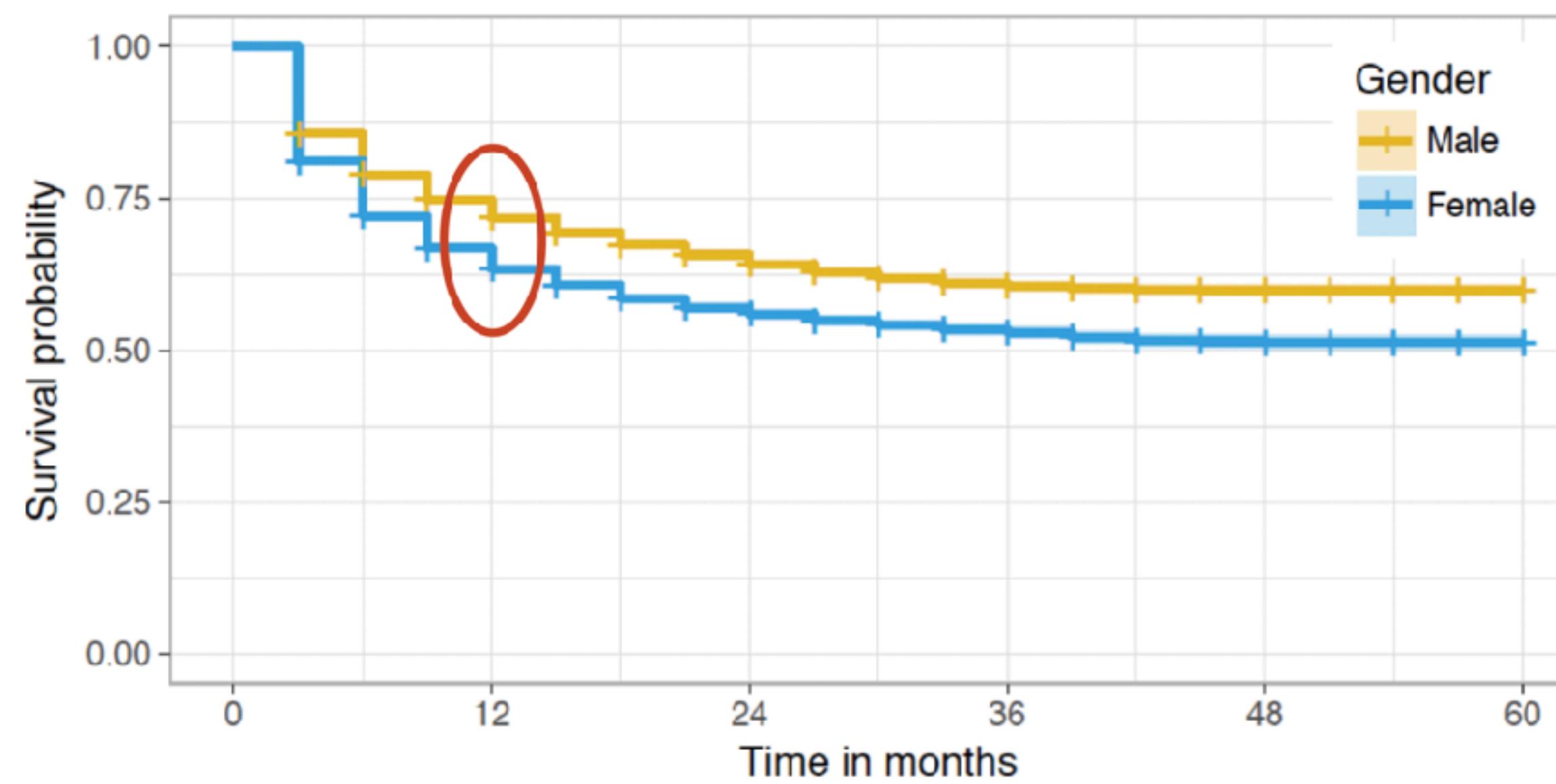
Lamba, H., Trockman, A., Armanios, D., Kästner, C., Miller, H., & Vasilescu, B. (2020). Heard it through the Gitvine: An empirical study of tool diffusion across the npm ecosystem. In Proceedings of the Joint Meeting on the Foundations of Software Engineering (pp. 505-517).

3. We're still not appreciating theory enough

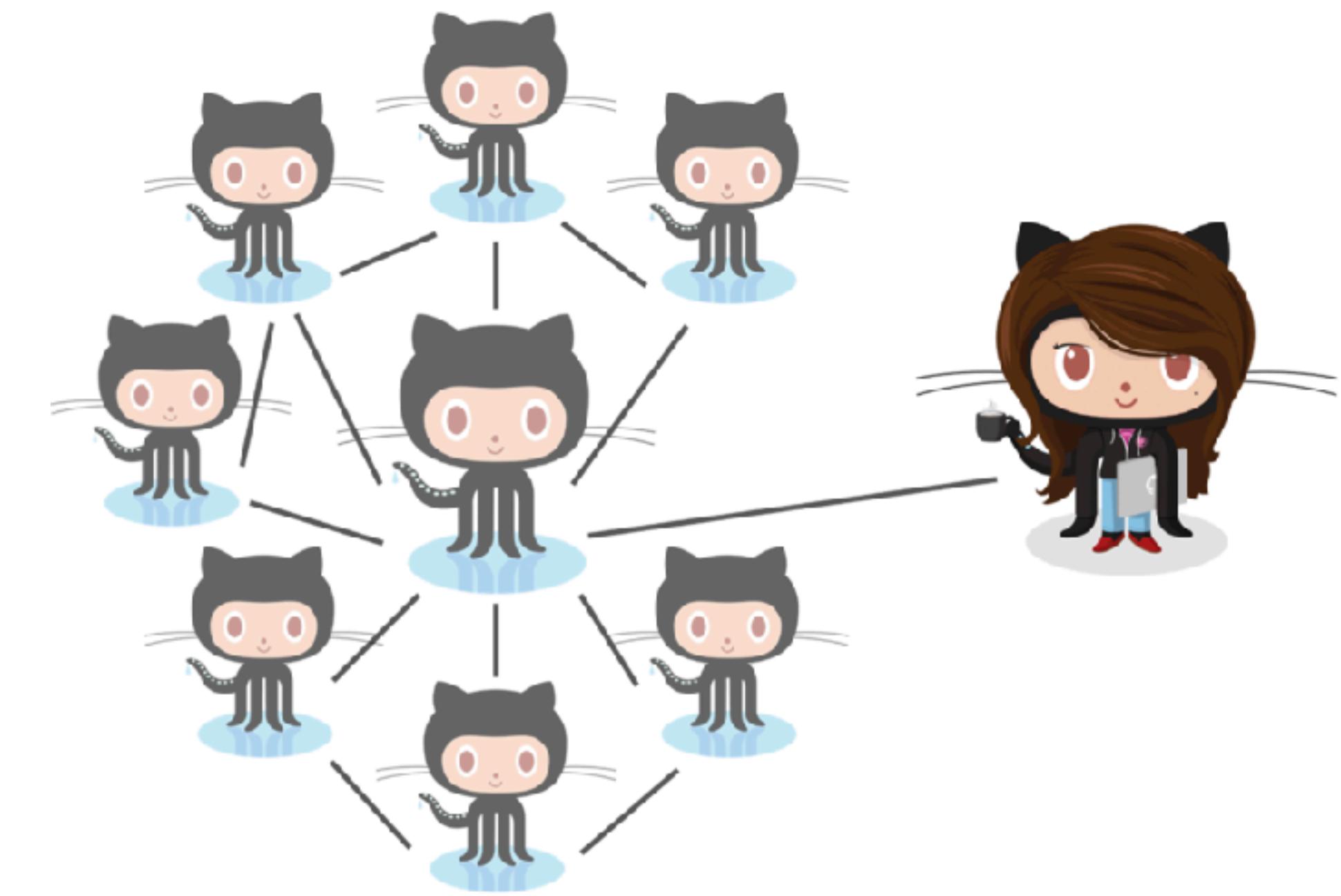


Theories are the building blocks of scientific knowledge.
They explain how and why certain phenomena occur, and allow predictions to be made.

Social capital theory explains why in open source women disengage earlier than men



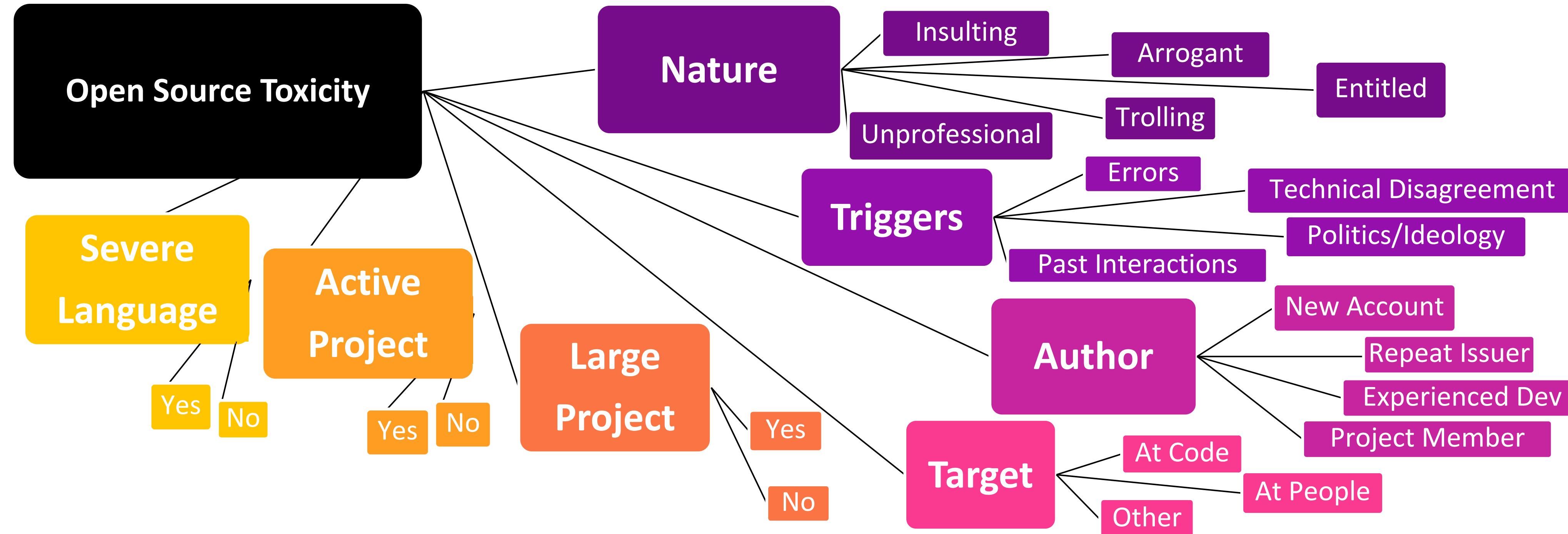
After one year ~70% of men are still active
but only ~60% of women.



Higher risk of discrimination / exclusion in cohesive networks, when being part of an underrepresented group.

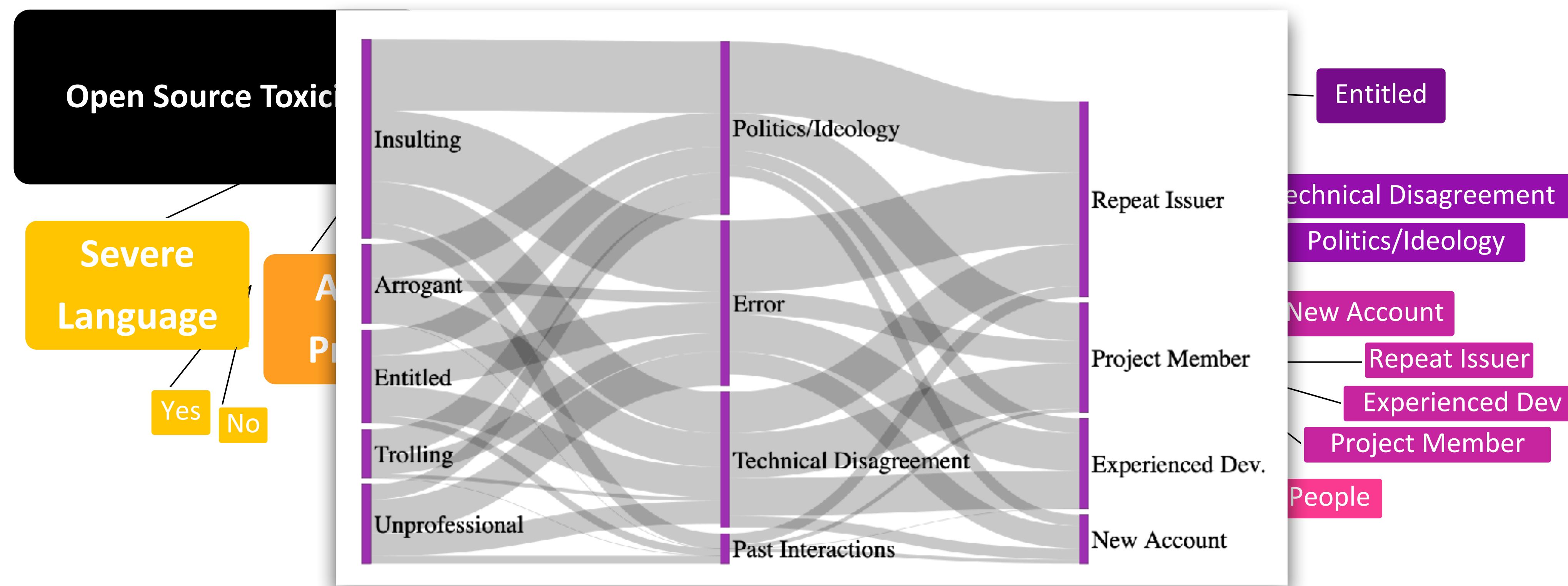
Qiu, H. S., Nolte, A., Brown, A., Serebrenik, A., & Vasilescu, B. (2019). Going farther together: The impact of social capital on sustained participation in open source. In Proceedings International Conference on Software Engineering (ICSE) (pp. 688-699). IEEE.

Toxicity in open source (GitHub) is different than on other platforms: e.g., less hate speech, more entitlement



Miller, C., Cohen, S., Klug, D., Vasilescu, B., & KaUstner, C. (2022). "Did you miss my comment or what?" Understanding toxicity in open source discussions. In Proceedings of the 44th International Conference on Software Engineering (pp. 710-722).

Toxicity in open source (GitHub) is different than on other platforms: e.g., less hate speech, more entitlement



Miller, C., Cohen, S., Klug, D., Vasilescu, B., & KaUstner, C. (2022). "Did you miss my comment or what?" Understanding toxicity in open source discussions. In Proceedings of the 44th International Conference on Software Engineering (pp. 710-722).

History of appreciation of theory in SE research

K. Stol, B. Fitzgerald (2015) Theory-Oriented Software Engineering. *Science of Computer Programming*, vol. 101, pp. 79-98. DOI:10.1016/j.scico.2014.11.010

Theory-Oriented Software Engineering[☆]

Klaas-Jan Stol*, Brian Fitzgerald

*Lero—The Irish Software Engineering Research Centre
University of Limerick, Ireland*

Abstract

There has been a growing interest in the role of theory within Software Engineering (SE) research. For several decades, researchers within the SE research community have argued that, to become a ‘real’ engineering science, SE needs to develop stronger theoretical foundations. However, so far, the role of theory is neither fully appreciated nor well understood in SE research. Without a good common understanding of what theory *is*, what it constitutes in SE research, and the various roles it can play in SE research, it is difficult to appreciate how theory building can help to strengthen SE research. In this paper we discuss the importance of theory

“The purpose of this paper is to increase awareness of the importance, purpose and role of theory in software engineering research.” (2015)

History of appreciation of theory in SE research

K. Stol, B. Fitzgerald (2015) Theory-Oriented Software Engineering. *Science of Computer Programming*, vol. 101, pp. 79-98. DOI:10.1016/j.scico.2014.11.010

Theory-Oriented Software Engineering[☆]

Klaas-Jan Stol*, Brian Fitzgerald

*Lero—The Irish Software Engineering Research Centre
University of Limerick, Ireland*

Abstract

There has been a growing interest in the role of theory within Software Engineering (SE) research. For several decades, researchers within the SE research community have argued that, to become a ‘real’ engineering science, SE needs to develop stronger theoretical foundations. However, so far, the role of theory is neither fully appreciated nor well understood in SE research. Without a good common understanding of what theory *is*, what it constitutes in SE research, and the various roles it can play in SE research, it is difficult to appreciate how theory building can help to strengthen SE research. In this paper we discuss the importance of theory

“The purpose of this paper is to increase awareness of the importance, purpose and role of theory in software engineering research.” (2015)

Building a Socio-Technical Theory of Coordination: Why and How (Outstanding Research Award)

James Herbsleb
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
+1 412 268 8933
jdh@cs.cmu.edu

ABSTRACT

Research aimed at understanding and addressing coordination breakdowns experienced in global software development (GSD) projects at Lucent Technologies took a path from open-ended qualitative exploratory studies to quantitative studies with a tight focus on a key problem – delay – and its causes. Rather than being directly associated with delay, multi-site work items involved more people than comparable same-site work items, and the number of people was a powerful predictor of delay. To counteract this, we developed and deployed tools and practices to support more effective communication and expertise location. After conducting two case studies of open source development, an extreme form of GSD, we realized that many tools and practices could be effective for multi-site work, but none seemed to work

or innovate. When people organize in a habitual, consistent way, for example, in collocated teams, it is easy to overlook day-to-day coordination mechanisms that are simply taken for granted. It is easy to see the importance of things such as meetings of various flavors, processes, methods, and architectural separation, which have long been studied. Other, subtler mechanisms such as informal communication, practices, habits, and shared mental models are often only made visible by their absence.

Very interesting – and often disturbing – things happen when an organization is geographically split apart. Much can be learned by observing the mayhem that often ensues when organizations are distributed, and much is revealed about what must have been happening in the collocated case that keeps such chaos more or less at bay. Adding new tools and practices in these novel

@FSE 2016

History of appreciation of theory in SE research

K. Stol, B. Fitzgerald (2015) Theory-Oriented Software Engineering. *Science of Computer Programming*, vol. 101, pp. 79-98. DOI:10.1016/j.scico.2014.11.010

Theory-Oriented Software Engineering

Klaas-Jan Stol*, Brian Fitzgerald
Lero—The Irish Software Engineering Research Institute
University of Limerick, Ireland

Abstract

There has been a growing interest in the role of theory within Software Engineering (SE). Some within the SE research community have argued that, to become a ‘real’ discipline, SE must have solid foundations. However, so far, the role of theory is neither fully appreciated nor well understood. This lack of common understanding of what theory *is*, what it constitutes in SE, and how it can be used, makes it difficult to appreciate how theory building can help to strengthen SE.

“The purpose of this paper is to argue that theory is important, and to explain the importance, purpose and value of theory in engineering research.” (2015)

University of Toronto

Department of Computer Science

You Gotta Have A Theory

Steve Easterbrook
sme@cs.toronto.edu
www.cs.toronto.edu/~sme

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Technical Theory of Coordination: Outstanding Research Award)

James Herbsleb
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
+1 412 268 8933
jdh@cs.cmu.edu

or innovate. When people organize in a habitual, consistent way, for example, in collocated teams, it is easy to overlook day-to-day coordination mechanisms that are simply taken for granted. It is easy to see the importance of things such as meetings of various flavors, processes, methods, and architectural separation, which have long been studied. Other, subtler mechanisms such as informal communication, practices, habits, and shared mental models are often only made visible by their absence.

Very interesting – and often disturbing – things happen when an organization is geographically split apart. Much can be learned by observing the mayhem that often ensues when organizations are distributed, and much is revealed about what must have been happening in the collocated case that keeps such chaos more or less at bay. Adding new tools and practices in these novel

@FSE 2016

@FSE Doctoral Symposium 2006

History of appreciation of theory in SE research ... contradicts recent experience on the ICSE 2023 PC :(

K. Stol, B. Fitzgerald (2015) Theory-Oriented Software Engineering. *Science of Computer Programming*, vol. 101, pp. 79-98. DOI:10.1016/j.scico.2014.11.010

Theory-Oriented Software Engineering

Klaas-Jan Stol*, Br

Lero—The Irish Software Engin
University of Lime

Abstract

There has been a growing interest in the role of theory within Software Engineering (SE). Some within the SE research community have argued that, to become a ‘real’ discipline, SE must have solid foundations. However, so far, the role of theory is neither fully appreciated nor well understood. This lack of common understanding of what theory *is*, what it constitutes in SE, and how it can be used, makes it difficult to appreciate how theory building can help to strengthen SE.

“The purpose of this paper is to argue that theory is important, the importance, purpose and value of theory in software engineering research.” (2015)

University of Toronto

Department of Computer Science

You Gotta Have A Theory

Steve Easterbrook
sme@cs.toronto.edu
www.cs.toronto.edu/~sme

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

1

Technical Theory of Coordination: Outstanding Research Award)

James Herbsleb
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
+1 412 268 8933
jdh@cs.cmu.edu

or innovate. When people organize in a habitual, consistent way, for example, in collocated teams, it is easy to overlook day-to-day coordination mechanisms that are simply taken for granted. It is easy to see the importance of things such as meetings of various flavors, processes, methods, and architectural separation, which have long been studied. Other, subtler mechanisms such as informal communication, practices, habits, and shared mental models are often only made visible by their absence.

Very interesting – and often disturbing – things happen when an organization is geographically split apart. Much can be learned by observing the mayhem that often ensues when organizations are distributed, and much is revealed about what must have been happening in the collocated case that keeps such chaos more or less at bay. Adding new tools and practices in these novel

@FSE 2016

@FSE Doctoral Symposium 2006

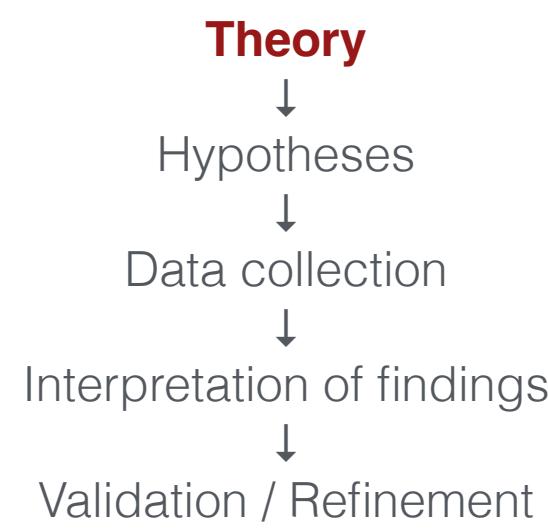
OUR UNIVERSITY[About the University](#)[Departments](#)[Calendar and events](#)[TU/e Campus](#)[Library](#)[Community](#)[Student teams](#)[HOME](#) / [OUR UNIVERSITY](#) / [CALENDAR AND EVENTS](#) / [INAUGURAL LECTURE PR...](#)

Inaugural lecture prof.dr. George Fletcher

“... In closing, I will outline where we are going in Eindhoven in putting people in the center and present a call for participation in the emerging social turn in the study of data systems.”

Let's make sure we value prior work and methods from disciplines where people are more of a tradition.

Theory both as part of the object of research and the study of research

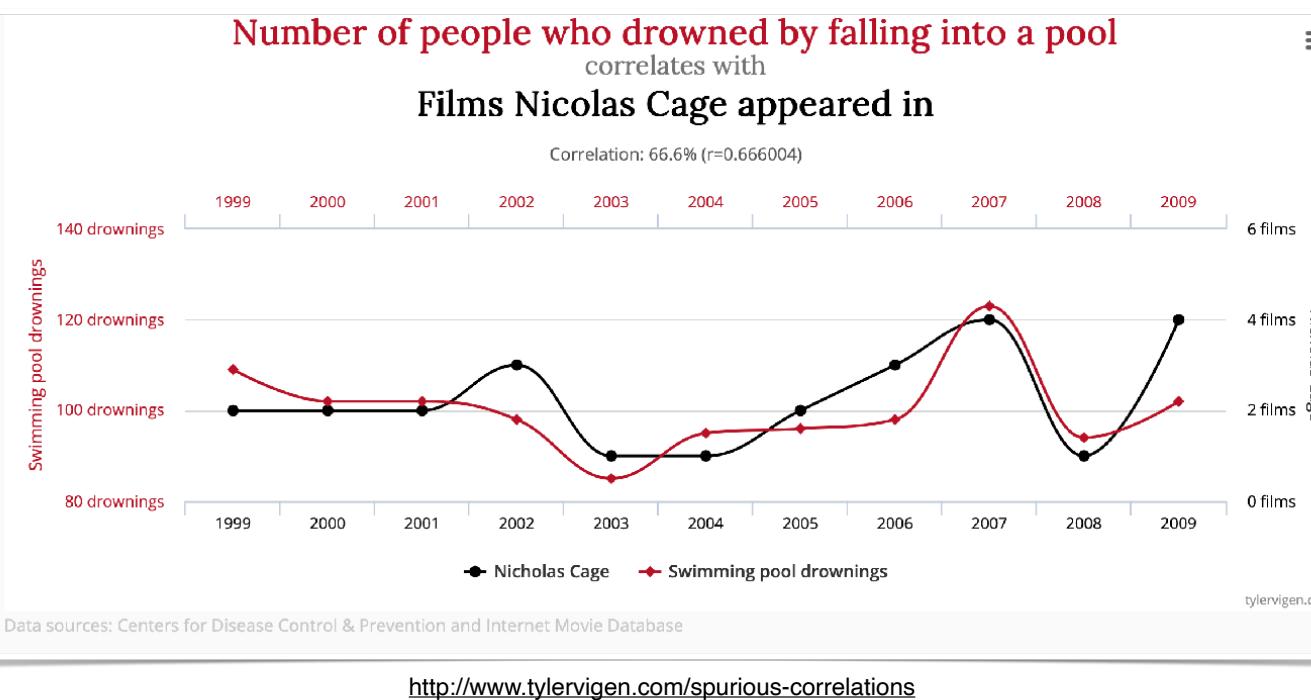


Theories are the building blocks of science
They explain how and why certain phenomena occur,

Carnegie Mellon University School of Computer Science STRUDEL

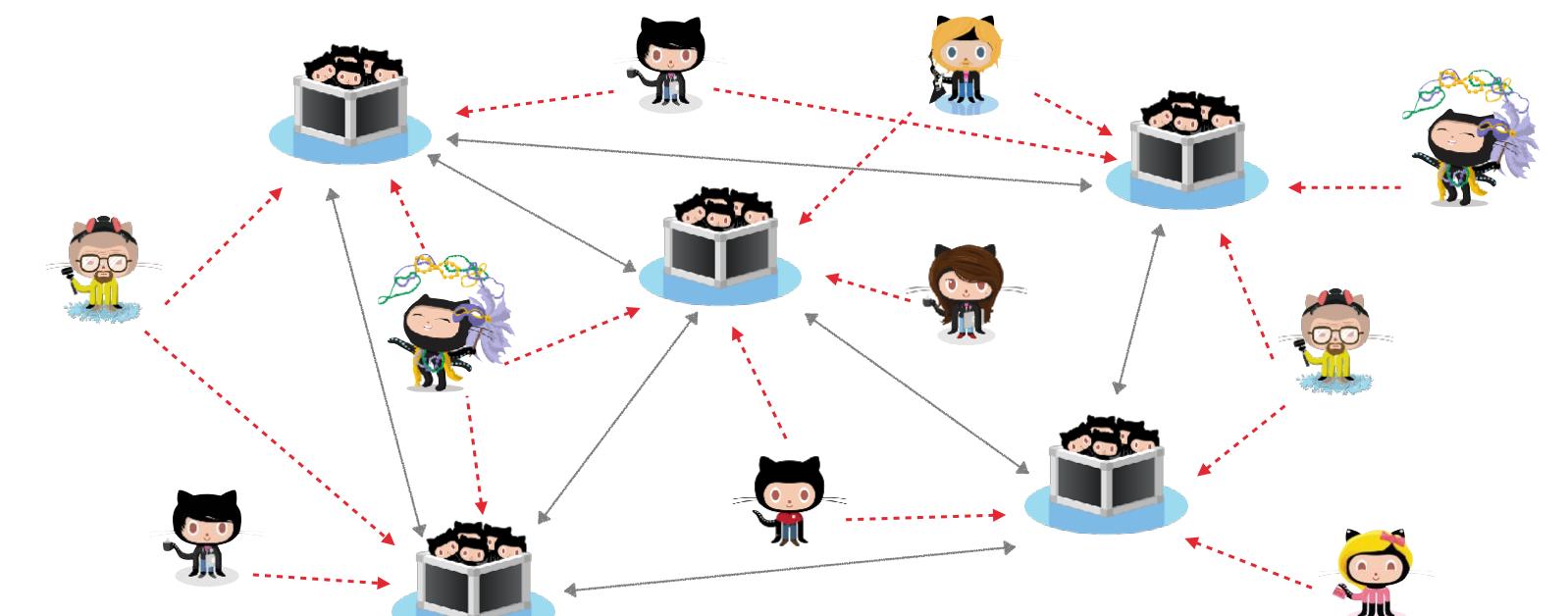
Data collection
Int
Pro

Correlation vs causation



Carnegie Mellon University School of Computer Science STRUDEL

Network effects



Carnegie Mellon University School of Computer Science STRUDEL

“... In closing, I will outline where we are going in Eindhoven in putting people in the center and present a call for participation in the emerging social turn in the study of data systems.”