

# Network Analysis:

The Hidden Structures behind the Webs We Weave

17-213 / 17-668

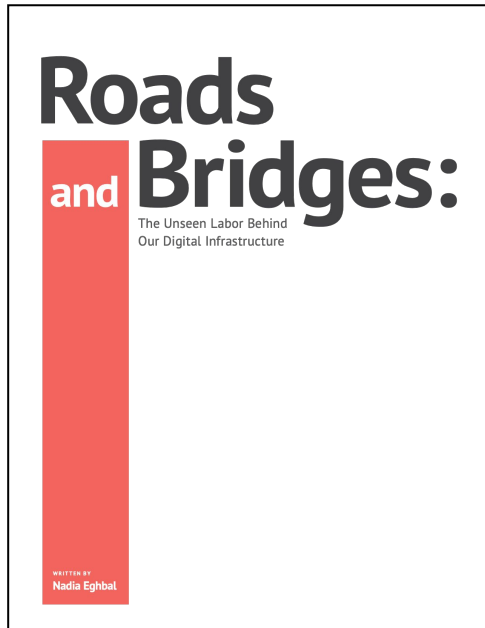
## Network Analysis of Open Source Software

Tuesday, November 14, 2023

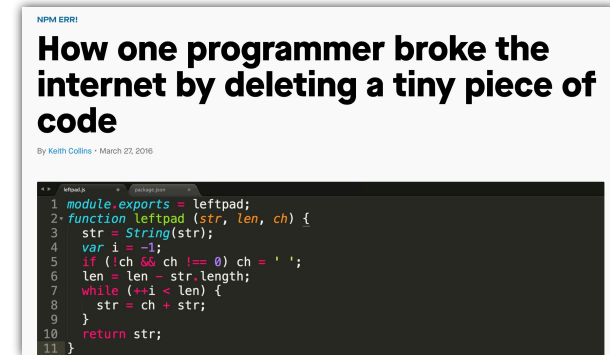
Patrick Park & Bogdan Vasilescu

# 2-min Quiz, on Canvas

# Open Source as digital infrastructure: Needs regular upkeep and maintenance



- Everybody uses open source code:
  - Fortune 500 companies
  - major software companies
  - startups
  - government
  - ...
- If undermaintained:
  - Risks for downstream users
  - Slows down innovation
  - ...



Creating **sustainable open source** communities is hard

In some ways **harder today than ever before**  
... because of how **open source** has  
**changed**

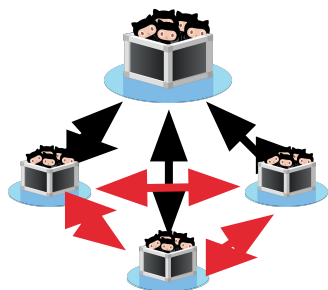


Today: more problems than  
solutions

How has open source  
changed?

# Change #1: GitHub standardized the practices

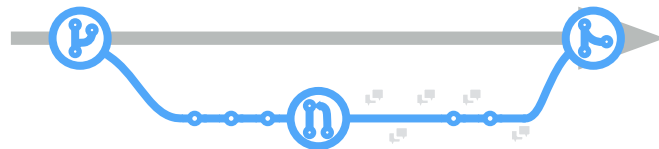
- Git version control



- GitHub UI



- The Pull Request model



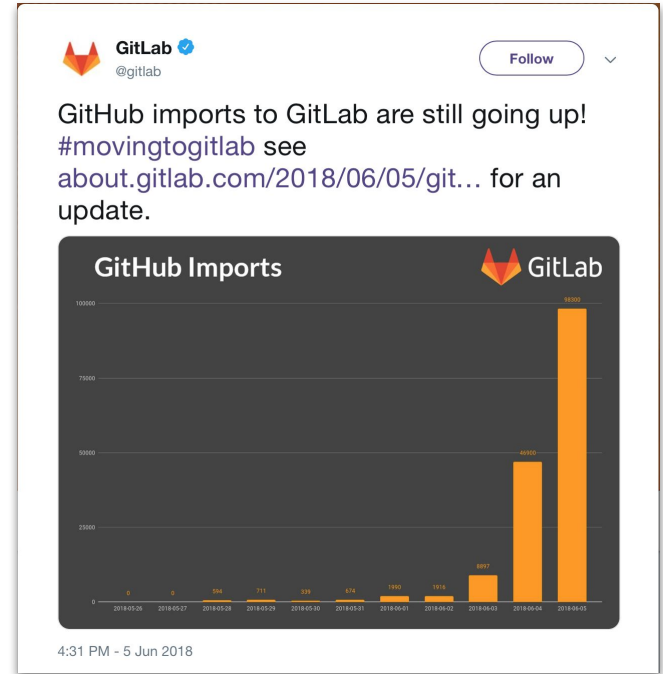
- Lower barrier to entry
- Easier to contribute



More production

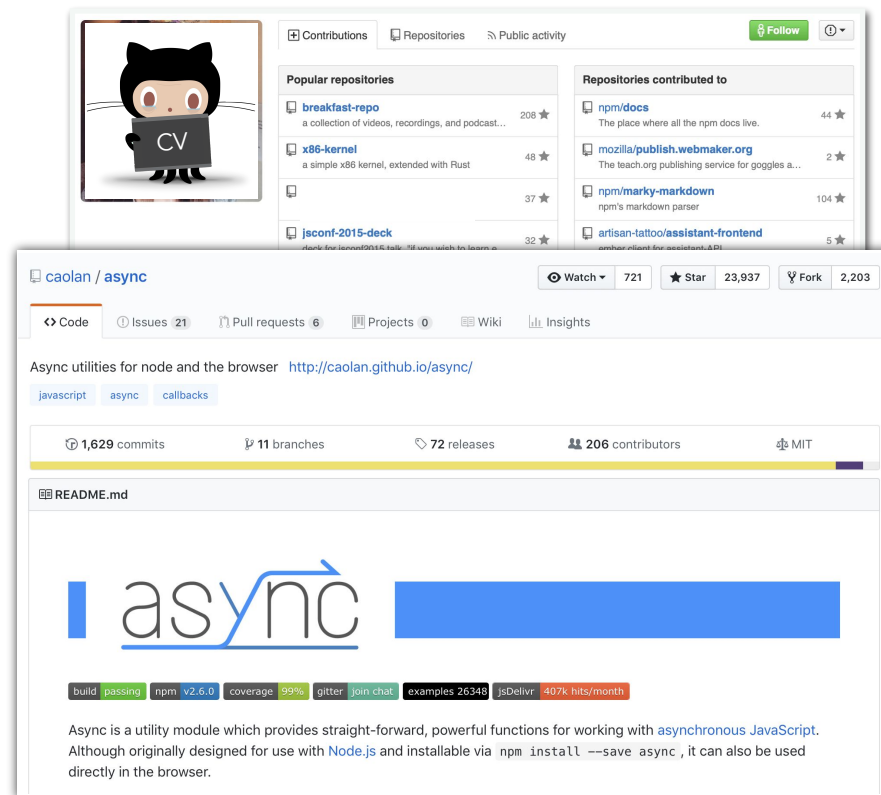
# Change #2: More open source now than ever before

- Explosion of production in the past seven years



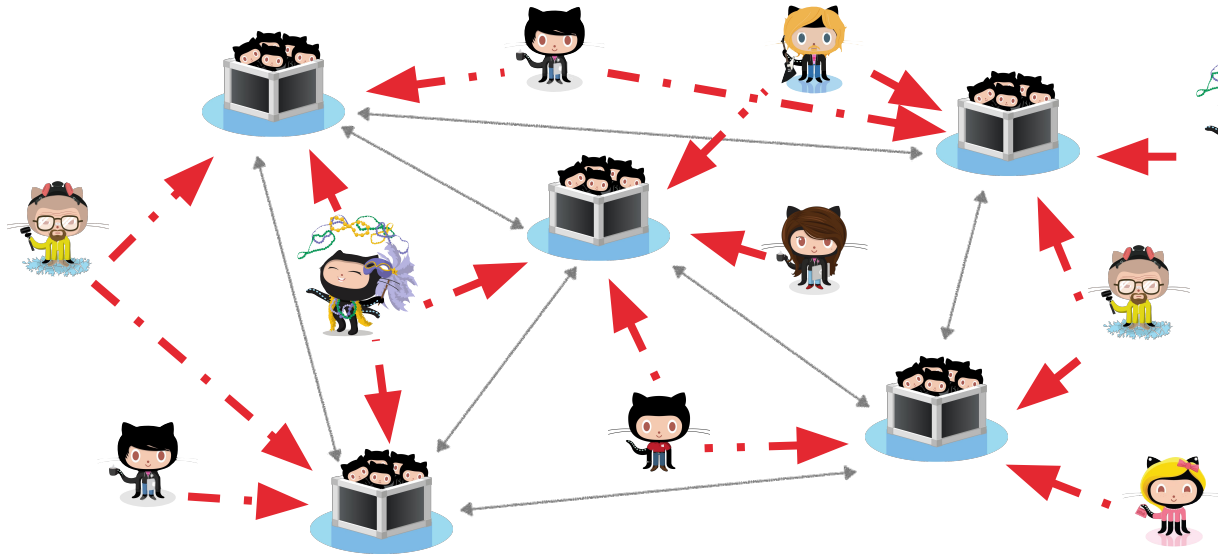
# Change #3: High level of transparency

- Profile pages for users and projects
- Rich inferences about people's expertise and level of commitment
- Impacts collaboration, but also recruiting and hiring
  - (Dabbish et al. 2012), (Marlow et al. 2013), (Marlow and Dabbish 2013)





# Change #4: Complex socio-technical ecosystems



Interconnections between people and projects

Can be brittle

NPM ERR!

### How one programmer broke the internet by deleting a tiny piece of code

By Keith Collins · March 27, 2016

```
1 module.exports = leftpad;
2 function leftpad(str, len, ch) {
3   str = String(str);
4   var i = -1;
5   if (!ch || ch !== 0) ch = ' ';
6   len = len - str.length;
7   while (++i < len) {
8     str = ch + str;
9   }
10  return str;
11 }
```

# Change #5: Increasing commercialization and professionalization

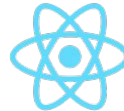
- Historically

- Mostly community-based projects (Python, RubyGems, Twisted)



- Currently

- Lots of commercial involvement
  - Companies (Go - Google, React - Facebook, Swift - Apple)
  - Startups (Docker, npm, Meteor)



- 23% of respondents to 2017 GitHub survey: job duties include contributing to open source

# Change #6: High expectations toward the quality, reliability, and security of open source infrastructure

- Equifax (market cap \$14 billion) built products on top of open-source infrastructure, including Apache Struts
- Equifax did not make any contributions to open source projects
- A flaw in Apache Struts contributed to the breach (CVE-2017-5638)
- Equifax publicly blamed (with national news coverage) Apache Struts for the breach

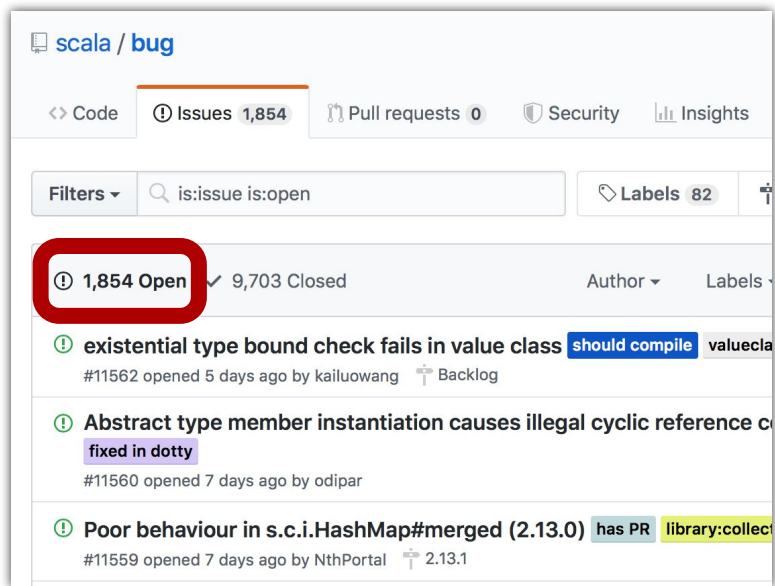


<https://www.zdnet.com/article/equifax-confirms-apache-struts-flaw-it-failed-to-patch-was-to-blame-for-data-breach/>

# Change #7: High level of demands & stress

- Easy to report issues / submit PRs
  - Growing volume of requests
- Social pressure to respond quickly
  - Otherwise, off-putting to newcomers (Steinmacher et al. 2015)
- Entitlement, unreasonable requests from users:
  - *"I have been waiting 2 years for Angular to track the 'progress' event and it still can't get it right?!?"*
  - *"Thank you for your ever useless explanations."*
  - ...

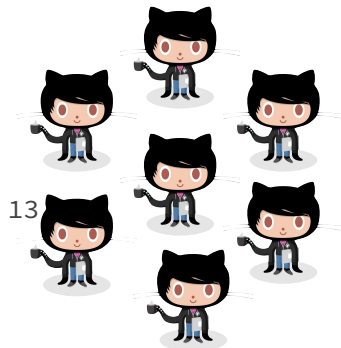
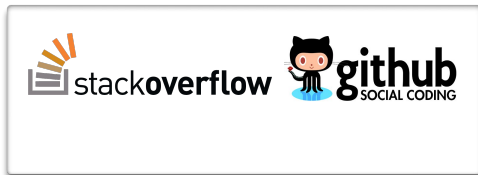
12



The screenshot shows the GitHub interface for the 'scala / bug' repository. At the top, there are navigation tabs for 'Code', 'Issues 1,854', 'Pull requests 0', 'Security', and 'Insights'. Below the tabs, there is a search bar with the query 'is:issue is:open' and a 'Filters' dropdown. To the right, there is a 'Labels 82' button. The main content area displays a summary of issues: '1,854 Open' (highlighted with a red circle) and '9,703 Closed'. Below this, there are three issue cards. The first card is titled 'existential type bound check fails in value class' with a 'should compile' label. The second card is titled 'Abstract type member instantiation causes illegal cyclic reference c' with a 'fixed in dotty' label. The third card is titled 'Poor behaviour in s.c.i.HashMap#merged (2.13.0)' with 'has PR' and 'library:collect' labels.

# Change #8: Low demographic diversity

- Gender representation reality



- Expectation




“More about the contributions to the *code* than the ‘characteristics’ of the person”

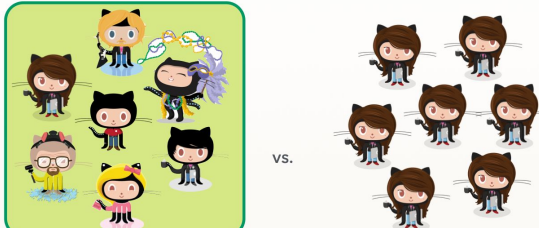
“Any *demographic identity* is irrelevant”

“Code sees *no color or gender*”

# Aside: Why should you care about gender diversity?

Productivity  
boosts

 **DIVERSE TEAMS ARE MORE PRODUCTIVE!**



vs.

Other confounds held fixed, **higher team diversity (gender & tenure)** is associated with **increased code production** (commits per quarter).

But small effects!

Inclusivity helps  
everyone

Why care? Inclusive design is not just for people with disabilities.

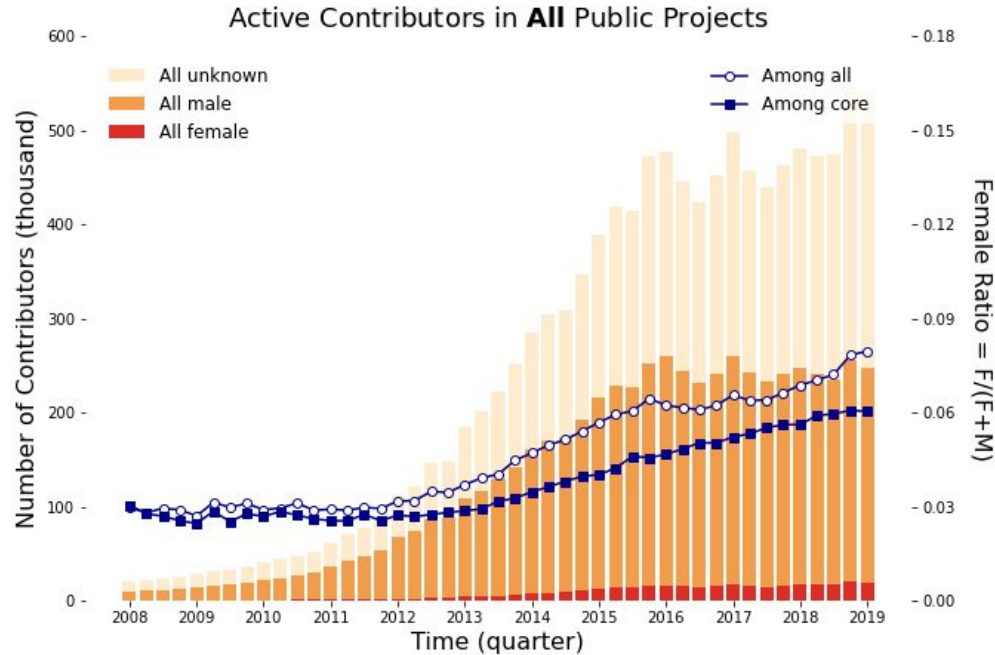


- For example, a ramp is useful for a person in a wheelchair, a person pushing a stroller, and a person with a suitcase.
- For example, a hand truck is useful for a person with a back injury, a person with a heavy load, and a person with a heavy suitcase.

• Gender and tenure diversity in GitHub teams. Vasilescu, B., Posnett, D., Ray, B., Brand, M.G.J. van den, Serebrenik, A., Devanbu, P., and Filkov, V. *CHI 2015*

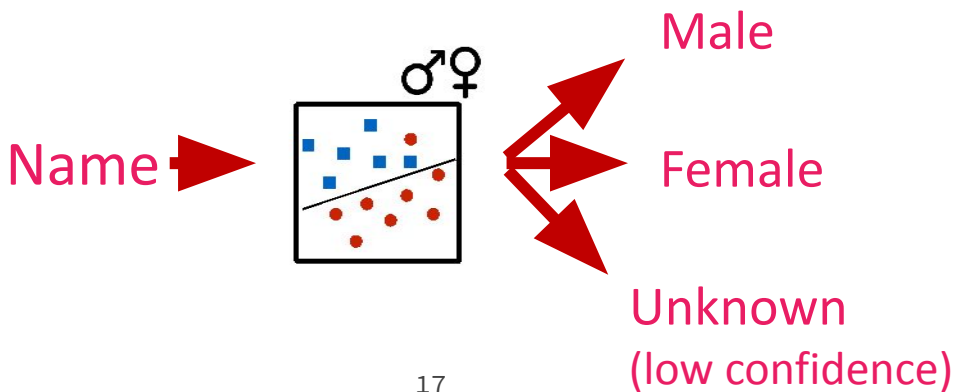
**“Going farther together: The impact of social capital on  
sustained participation in open source”  
Qiu\* et al, ICSE 2019**

# Skewed gender ratio: more than 90% of the OSS population is male





# Research scope - binary gender, GitHub

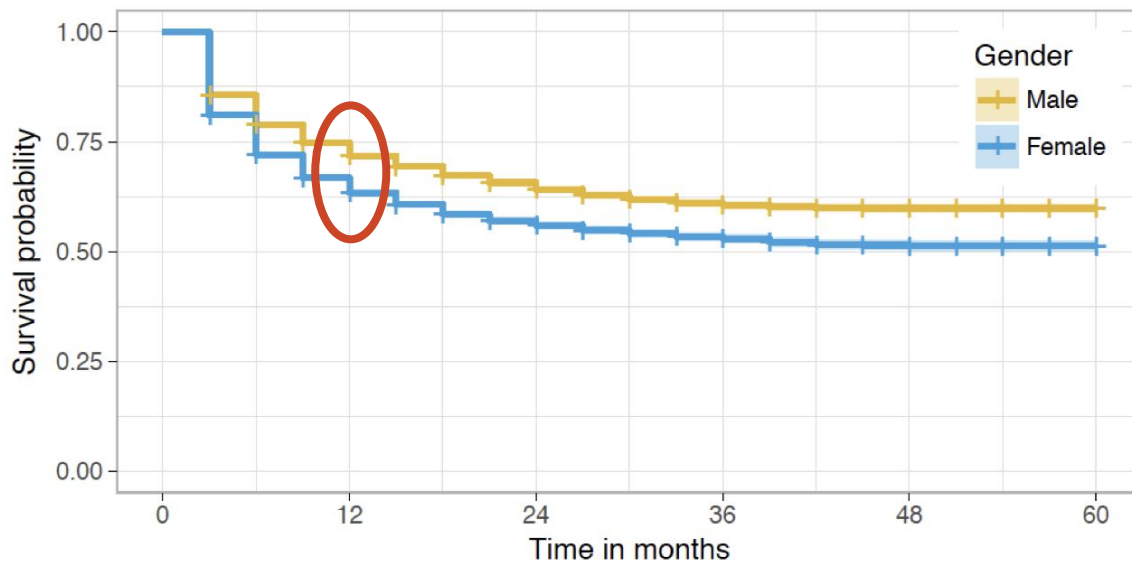


Gender diversity = Women + Men

A simplifying assumption: gender is binary

# On GitHub, women disengage earlier than men

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



# Low gender diversity as a challenge to OSS sustainability: limits contributor pool



19

[https://w3techs.com/technologies/history\\_overview/web\\_server](https://w3techs.com/technologies/history_overview/web_server)

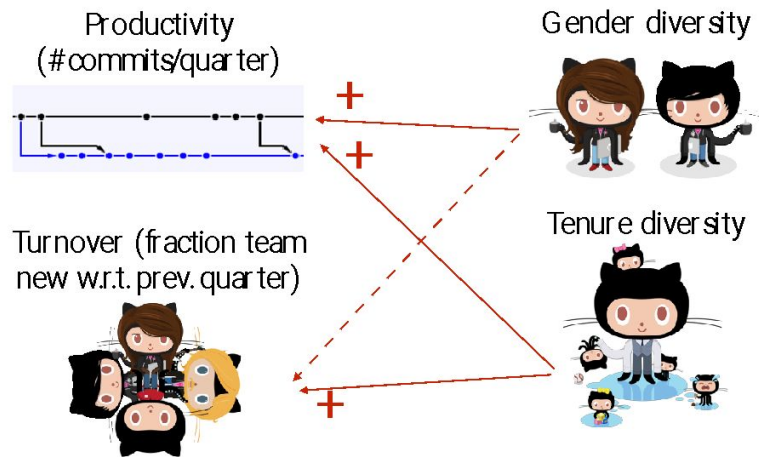
(Greenstein and Nagel, 2016)

# Low gender diversity as a challenge to OSS sustainability: harms project success

CHI'15, Seoul, South Korea

April 23, 2015

## Results



@b\_vasilescu

@baishakhr

@MarkvandenBrand

@aserebrenik

@devanbu

@vfilkov

[Vasilescu et al., 2015]

# Low gender diversity as a challenge to OSS sustainability: limits opportunities

Employers (and job seekers) use open-source experience to make inferences (or form impressions) about a candidate's technical skills.

(Marlow et al., 2013)

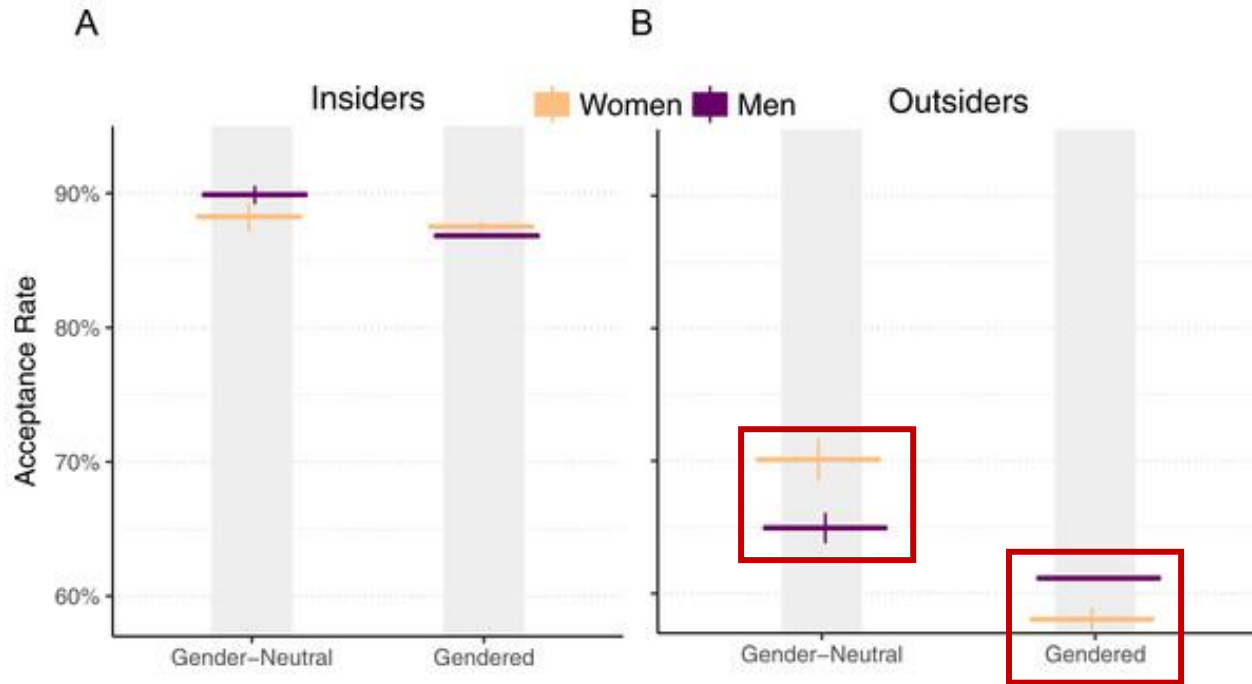
<CODE  
/\*for  
.MORE

Career advice for developers

**How to write up open-source experience when you don't have any**

<https://codeformore.com/how-to-write-up-open-source-experience-when-you-dont-have-any/>

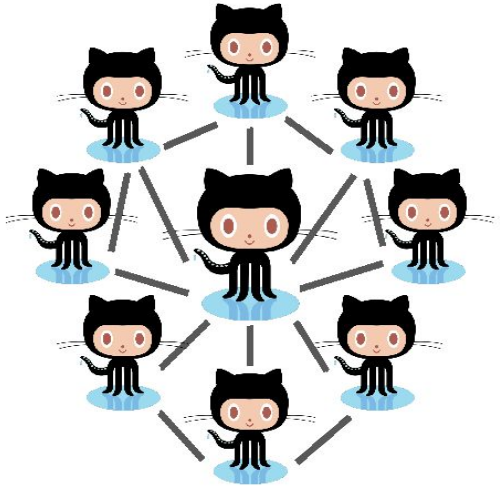
# Minorities face bias and discrimination.



[Terrell et al., 2017]

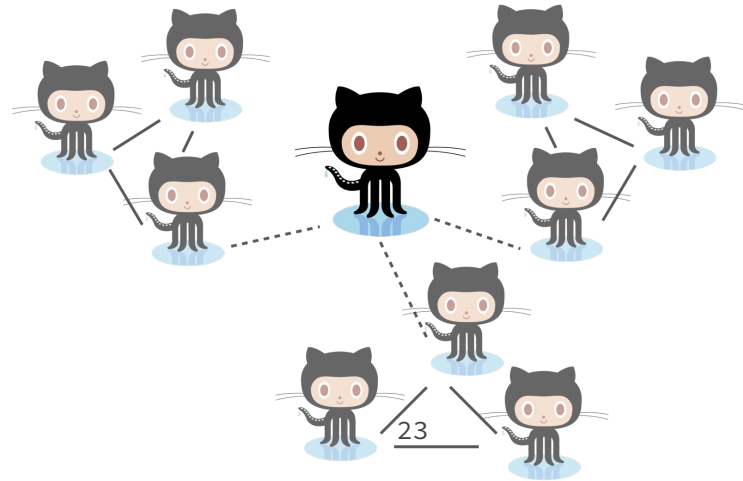
# Social capital theory for sustained participation

Bonding social capital:  
benefiting from strongly connected network



Willingness to continue  
(Coleman, 1990)

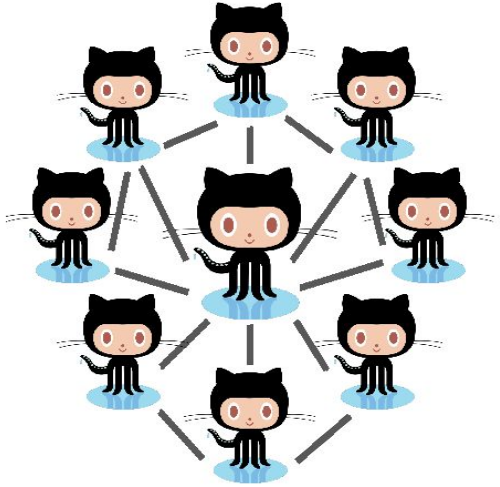
Bridging social capital:  
benefiting from network with diverse info



Opportunity to continue  
(Burt, 1998, 2001)

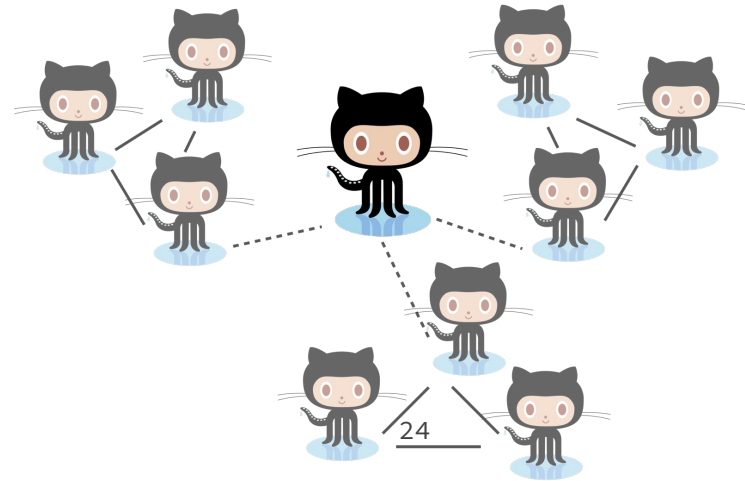
# H1: more social capital ~ more prolonged engagement

Bonding social capital:  
benefiting from strongly connected network



Willingness to continue  
(Coleman, 1990)

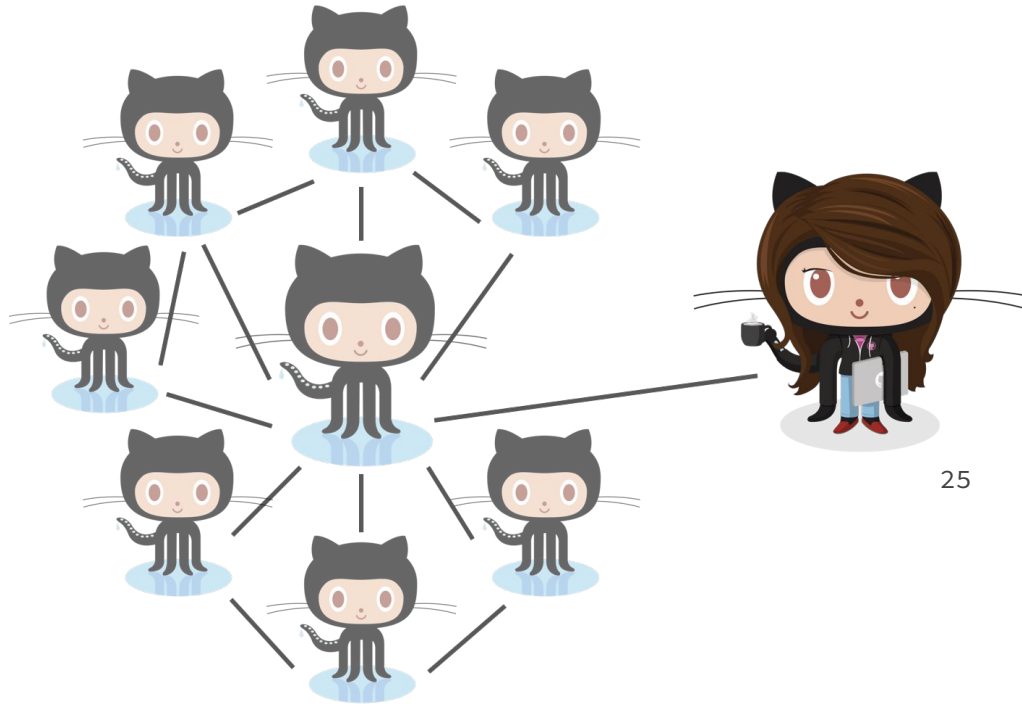
Bridging social capital:  
benefiting from network with diverse info



Opportunity to continue  
(Burt, 1998, 2001)

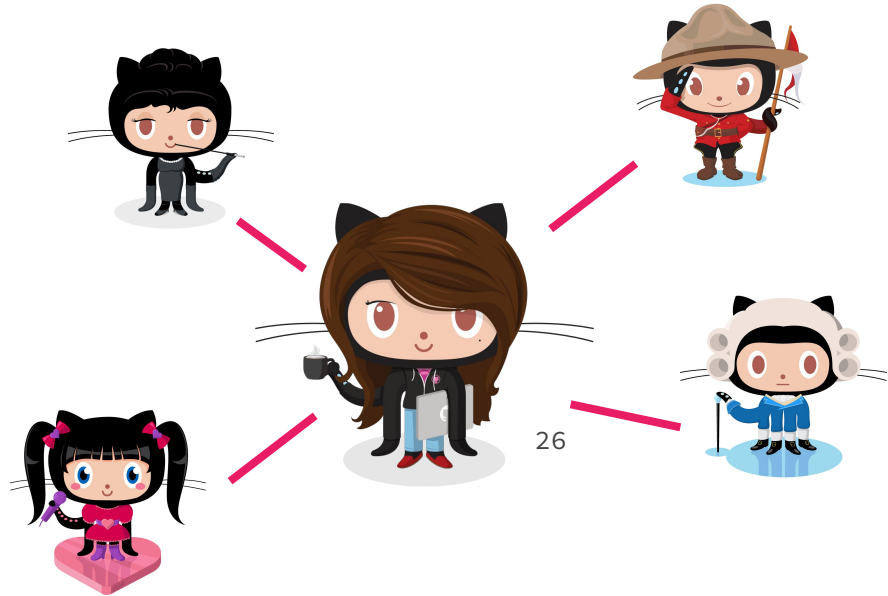


# Cohesive network might foster discrimination and exclusion

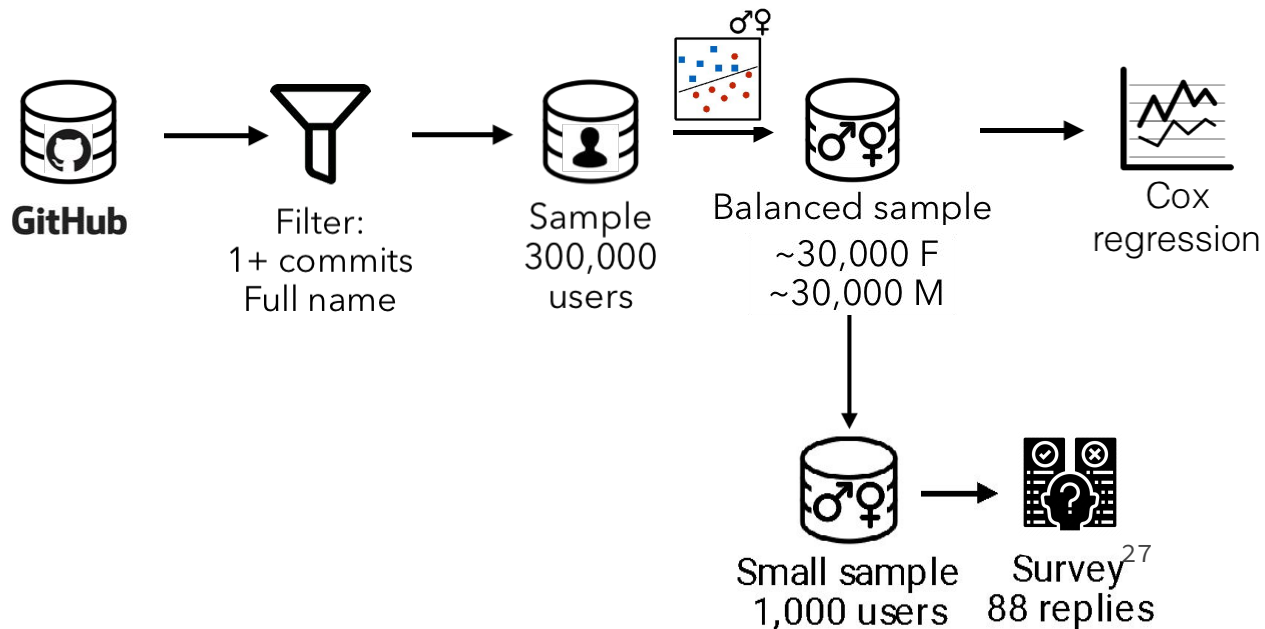


## H2: Teams with more diverse information ~ more prolonged engagement, esp. for women

Information diversity should reduce the risk of demographic-based echo chambers.

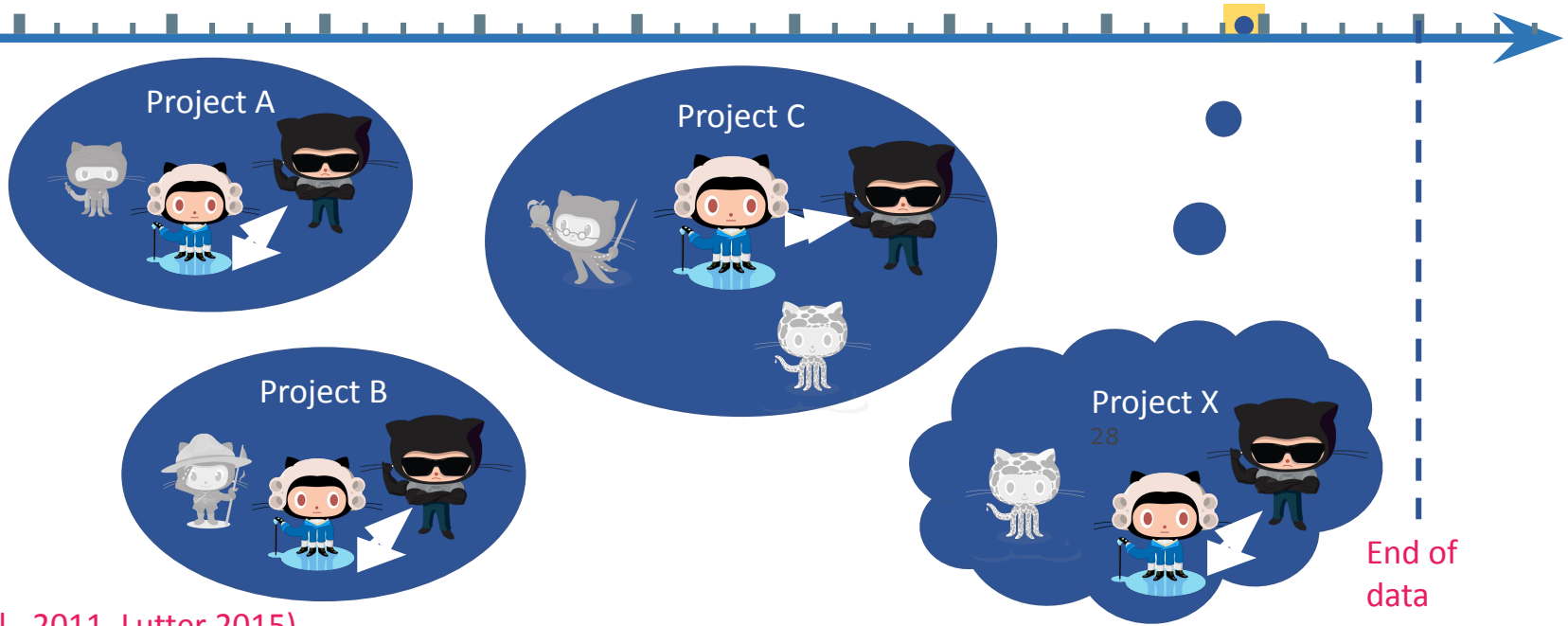


# Large-scale mixed-methods study



# Bonding social capital – Team Familiarity

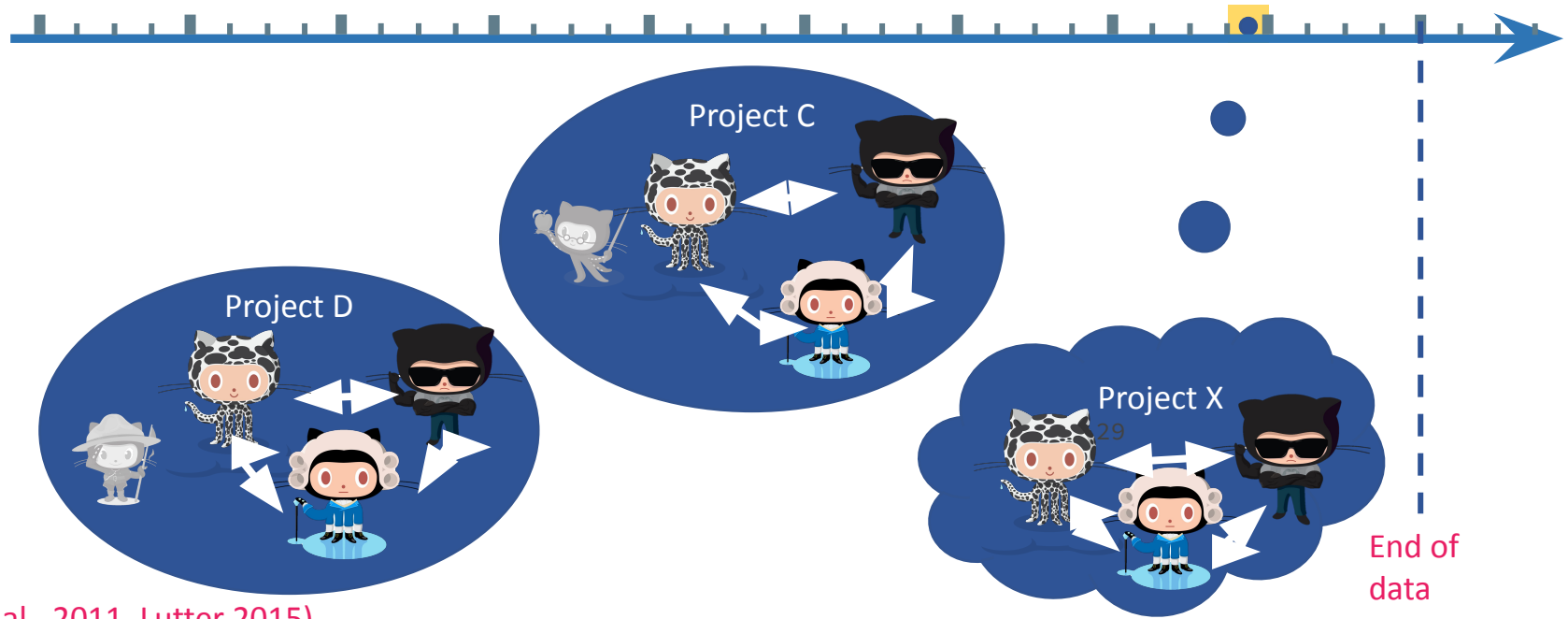
TIME



(de Vaan et al., 2011, Lutter 2015)

# Bonding social capital – Recurring Cohesion

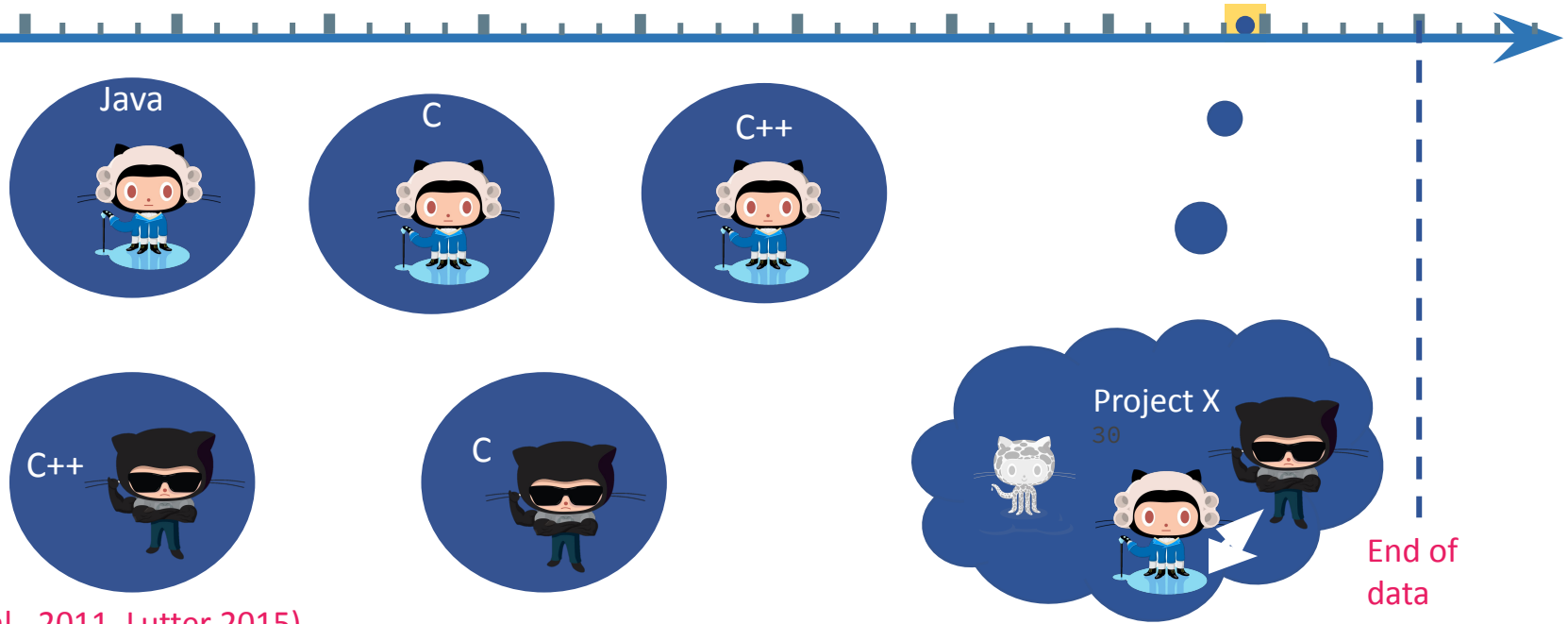
TIME



(de Vaan et al., 2011, Lutter 2015)

# Bridging social capital – Language Diversity

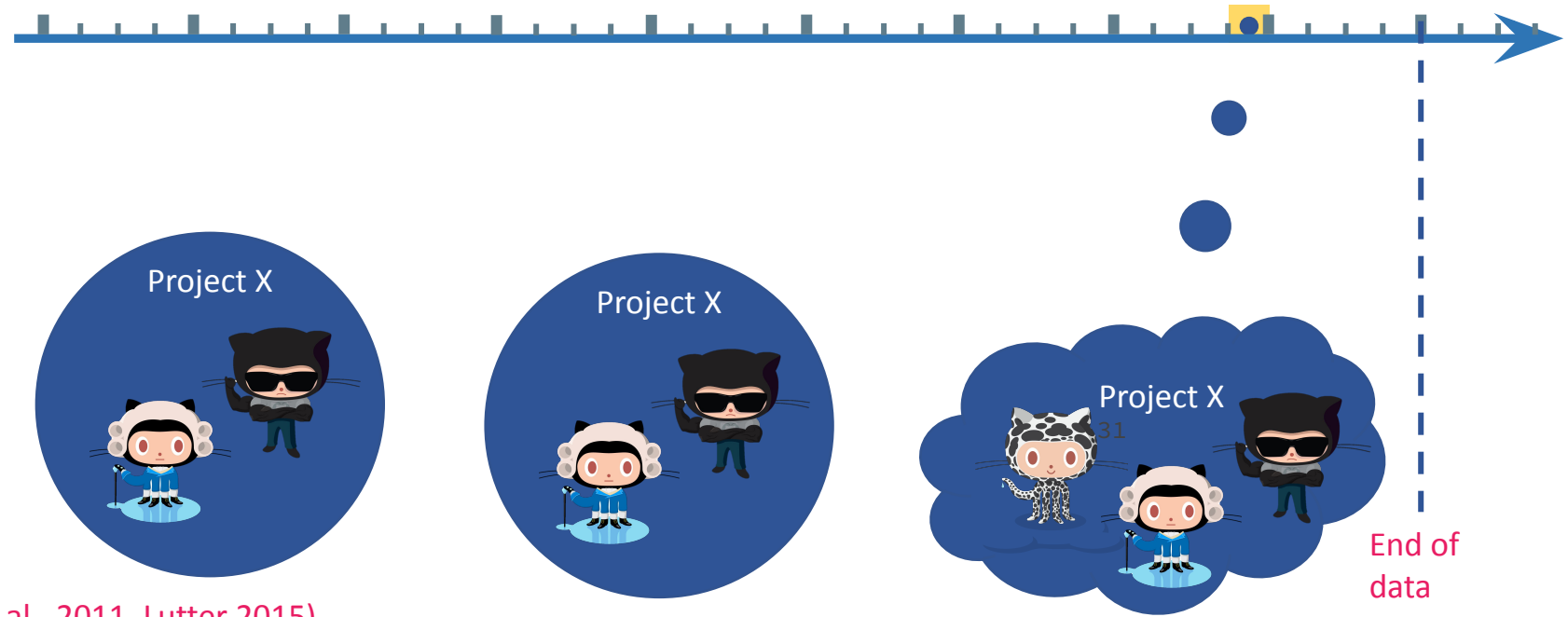
TIME



(de Vaan et al., 2011, Lutter 2015)




# Bridging social capital – Share of Newcomers

TIME



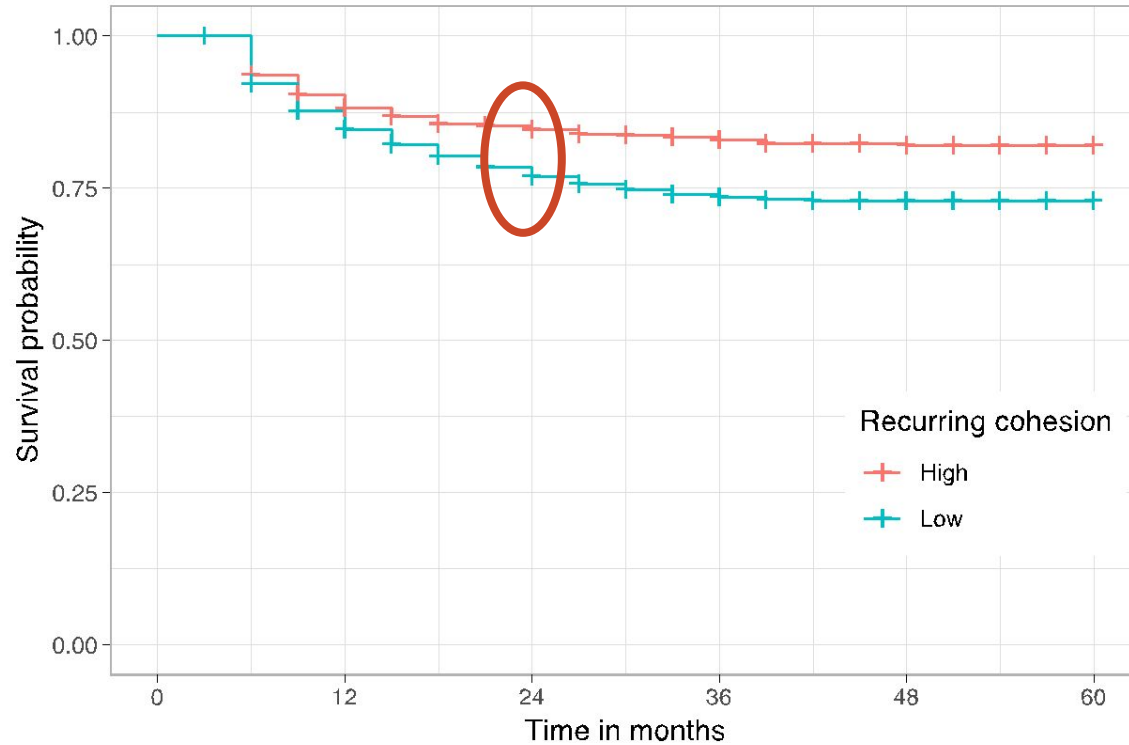
(de Vaan et al., 2011, Lutter 2015)

# COX regression model

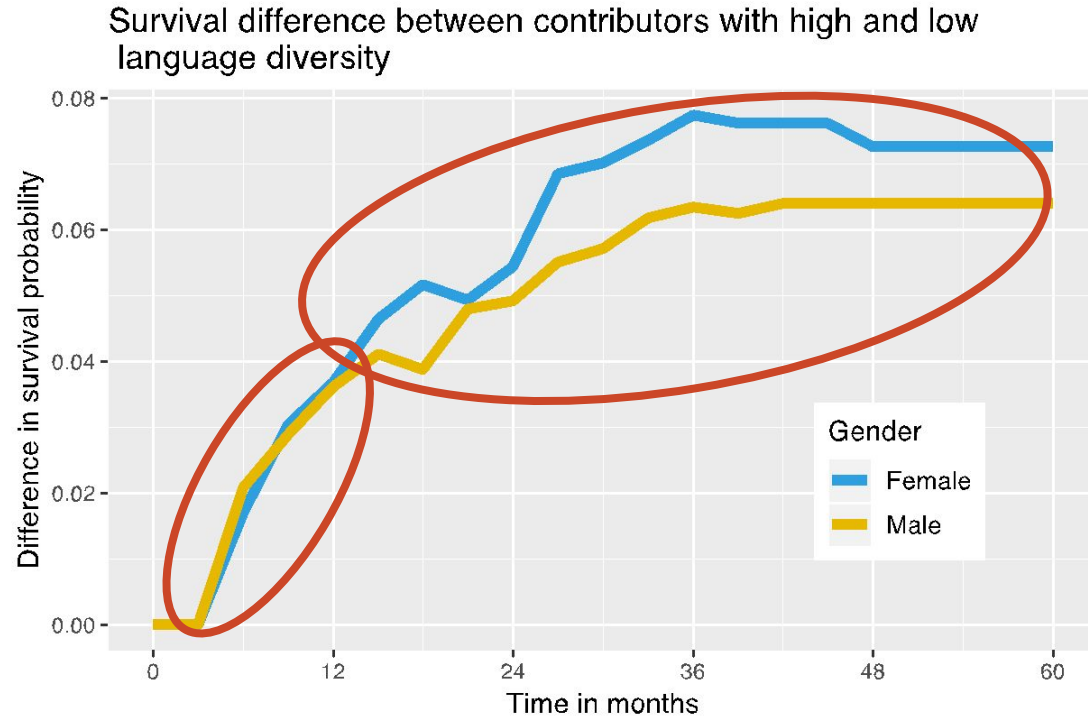
Contributor	Time	Active	Social capital	Control variables
	2008 Jan – Mar	True	Team familiarity Recurring cohesion Language diversity Share of newcomers	Project size Project owner .....
	2008 Jan – Mar	True	Team familiarity Recurring cohesion Language diversity Share of newcomers	Project size Project owner .....
	2009 Apr – Jun	False	Team familiarity Recurring cohesion Language diversity Share of newcomers	Project Size Not project owner .....



# H1: more social capital ~ more prolonged engagement

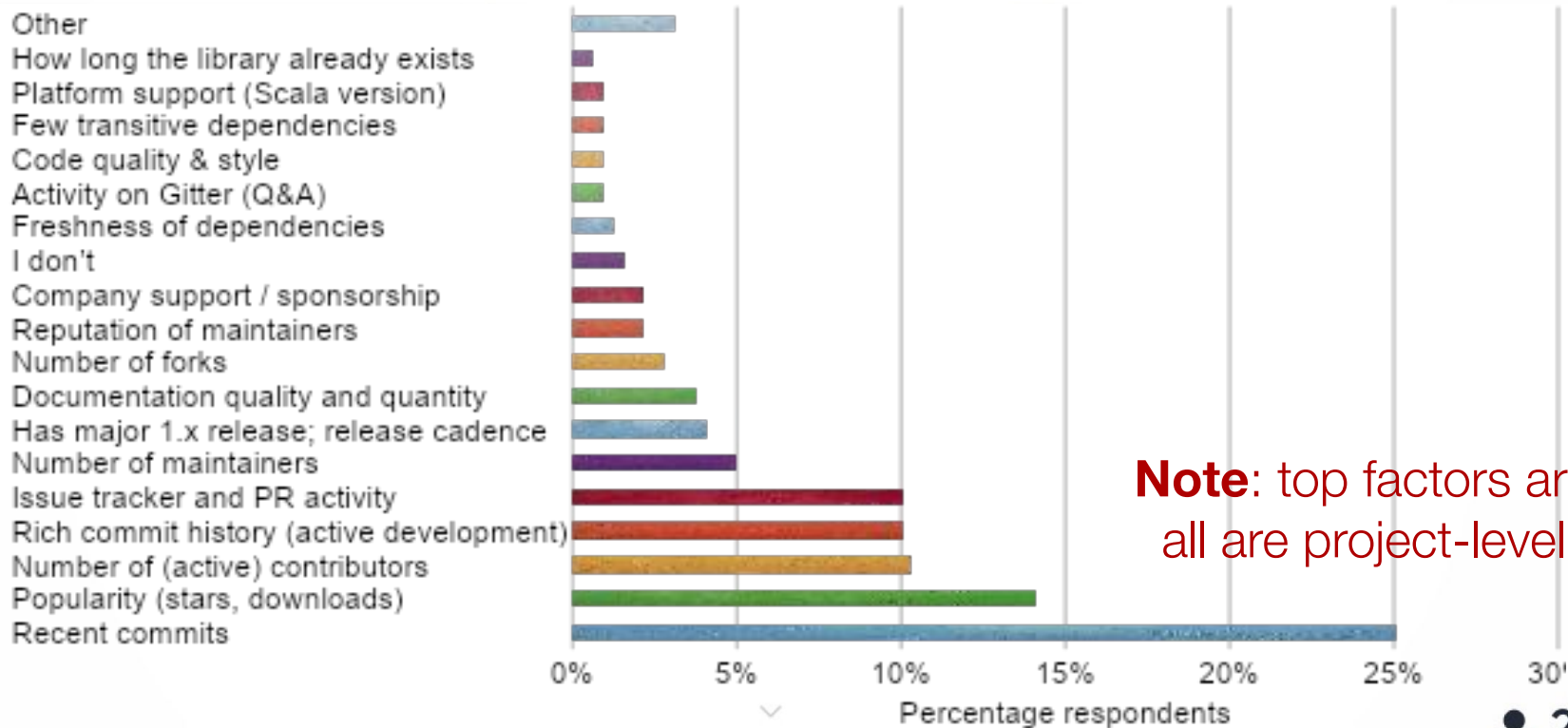


## H2: Language diversity interacts with gender



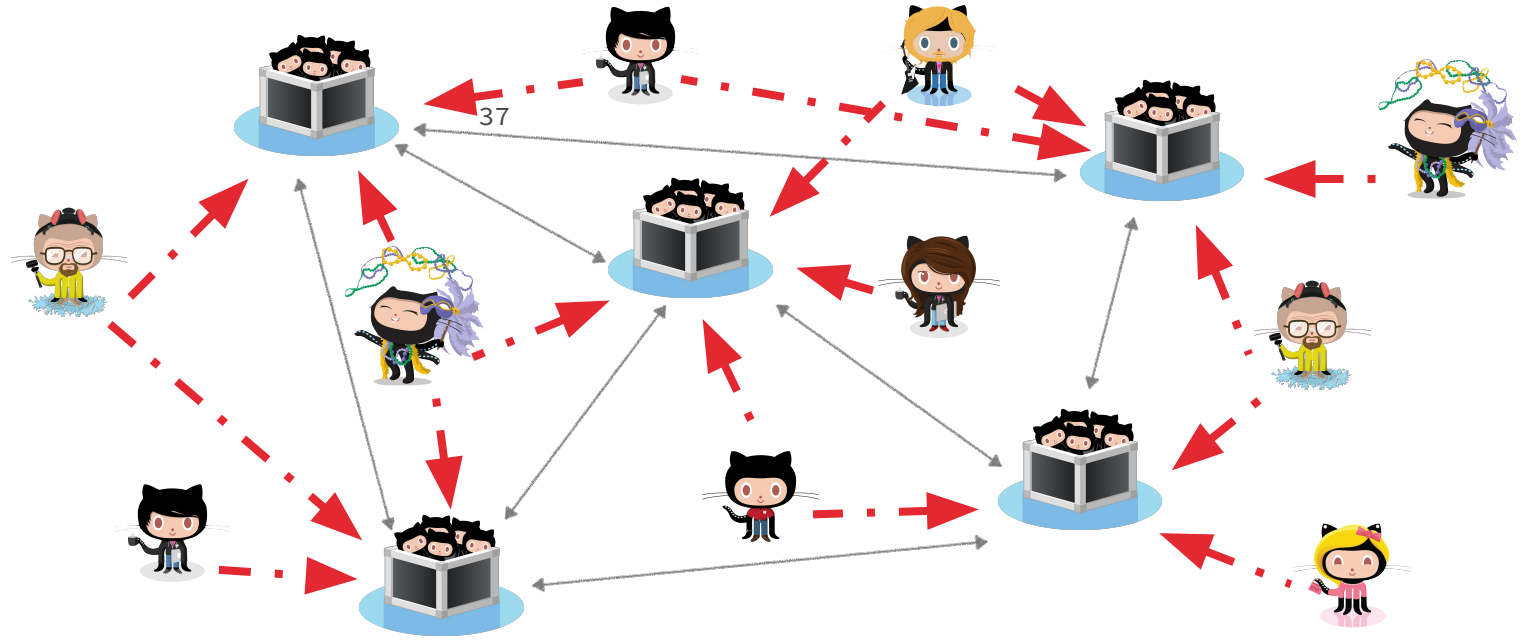
**“Ecosystem-level determinants of sustained activity in  
open-source projects: A case study of the PyPI ecosystem”  
Valiev et al, FSE 2018**

# How do you screen open source libraries to make sure they would still be maintained in the future?

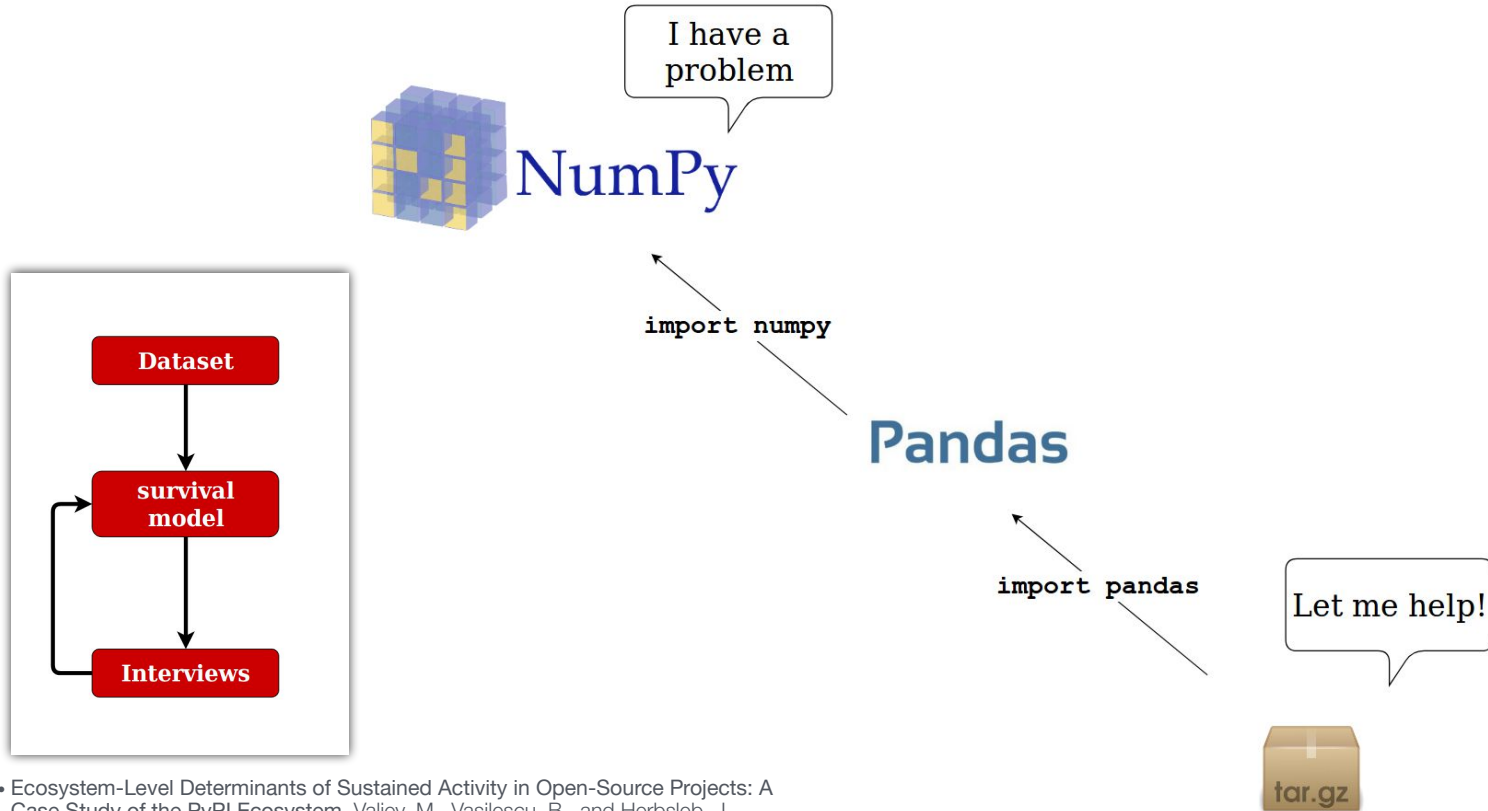


**Note:** top factors are all are project-level

But projects are often part of larger ecosystems

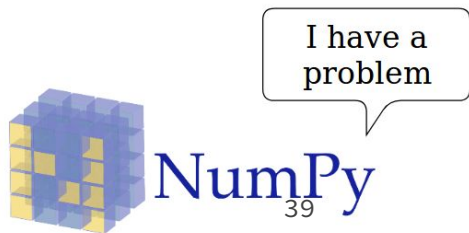


# Transitive downstream dependencies are .....



- Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. Valiev, M., Vasilescu, B., and Herbsleb, J. *ESEC/FSE 2018*

# Transitive downstream dependencies are harmful



## Survival models

Early stage: **-12%** survival

Long term: **-27%** survival

`import numpy`

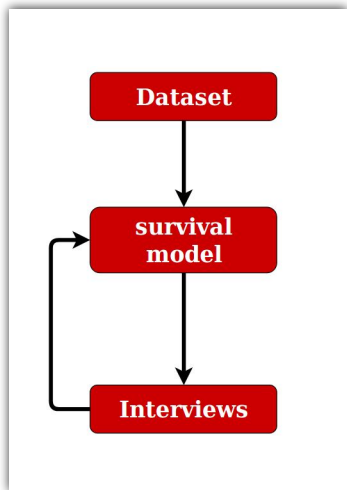
Pandas

`import pandas`

## Interviews:

- less likely to fix
- just as likely to complain

Let me help!

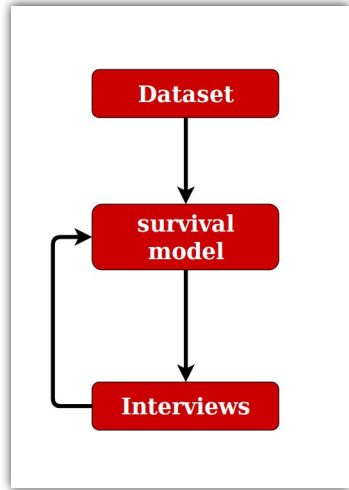


# Commercial involvement is .....

I have a  
problem

OR

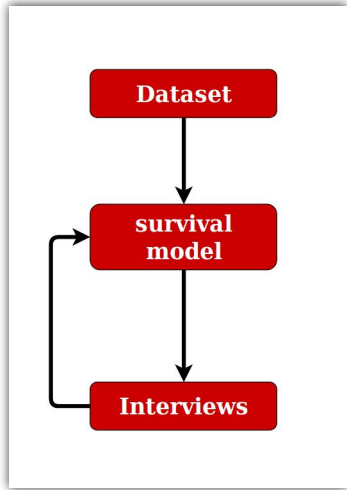
Let me help!



- Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. Valiev, M., Vasilescu, B., and Herbsleb, J. *ESEC/FSE 2018*



# Commercial involvement is harmful



I have a problem

OR

Let me help!



## Survival models

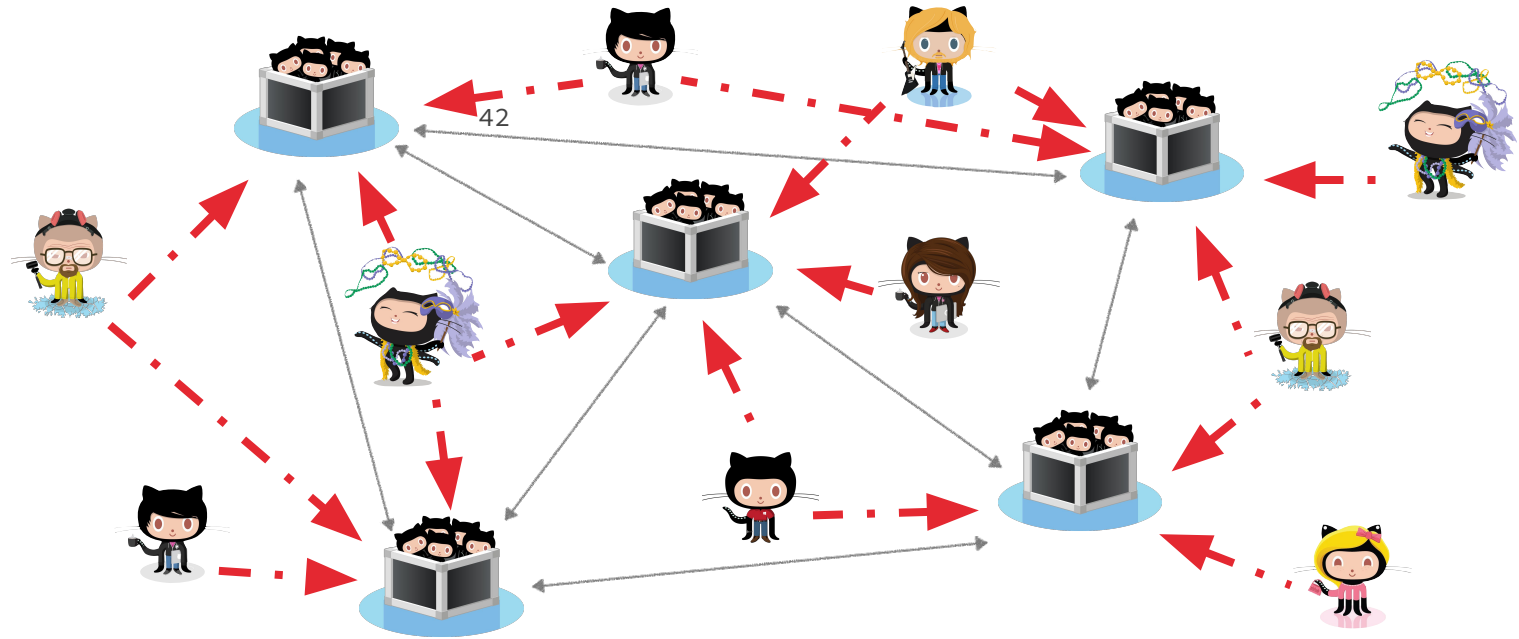
Early stage: **-51%** survival

Long term: **-15%** survival

## Interviews:

- more resources
- but can withdraw anytime

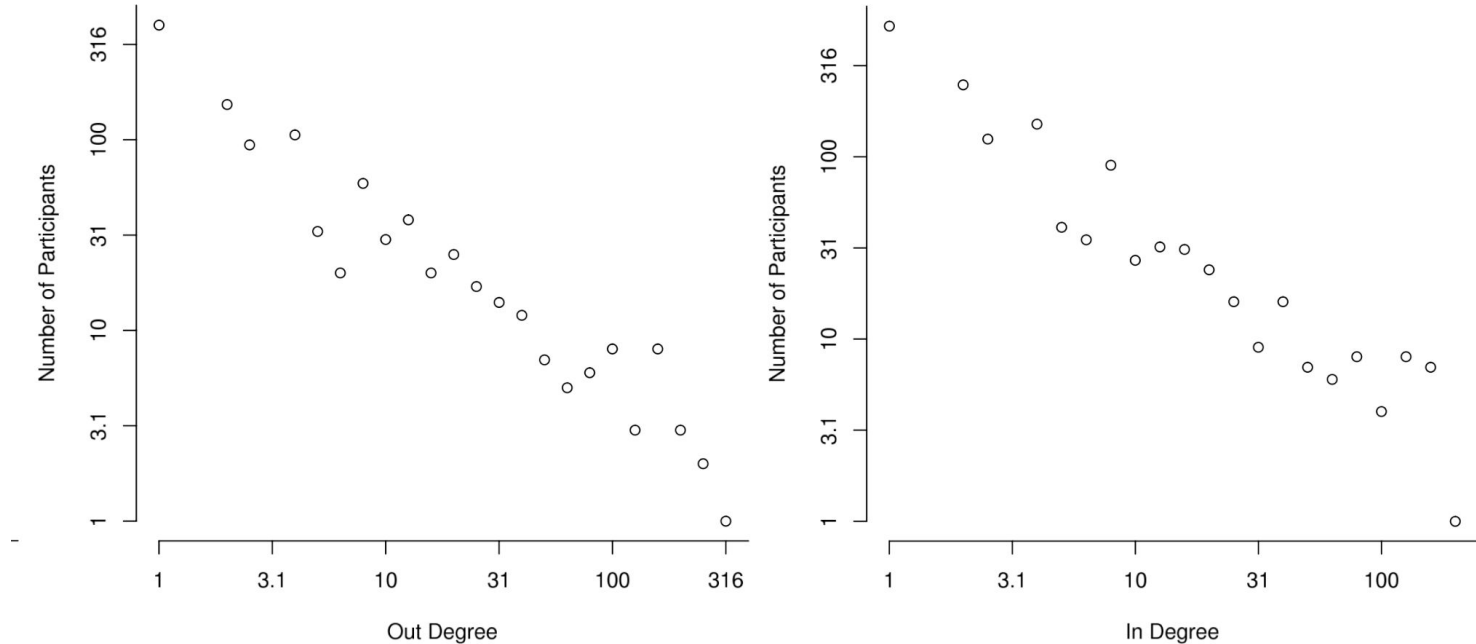
# Take away: Network effects!



# **“Mining Email Social Networks”**

**Bird et al, MSR 2006**

# Email social networks are scale free



Out degree is an indication of status, as it indicates the number of different people who replied to the ego's messages.

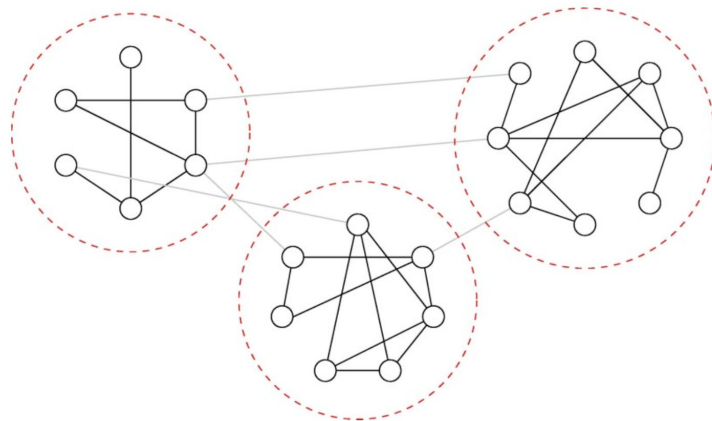
# **“Latent Social Structure in Open Source Projects”**

**Bird et al, FSE 2008**

# Do OSS projects have some latent structure?

Are there dynamic, self-organizing subgroups that spontaneously form and evolve?

Hypothesis 1 – Subcommunities of participants will form in the email social networks of large open source projects and the levels of modularity will be statistically significant.



**Figure 1:** A network with strong community structure. *Modularity*, the measure of strength of community structure, which ranges from 0 to 1, has a value of 0.493 for the given division of nodes in this graph.

# Two types of discussions on the development mailing lists

“Product” – development activity, function interfaces, APIs, bug fixes, feature implementation, etc.

“Process” – policy decisions, high-level architectural changes, release plans, licensing issues, and admission of newcomers.

Hypothesis 2 – Social networks constructed from product-related discussions will be more modular than those relating to non-product related discussions or all discussions.

# The subcommunities should be related to the software engineering activities in a meaningful way.

Hypothesis 3 – Pairs of developers within the same subcommunity will have more files in common than pairs of developers from different subcommunities.

Hypothesis 4 – The average directory distance between files committed to by developers in the same subcommunity will be less than similar sized groups of developers drawn different subcommunities.



# Mining the developer mailing list archives and source code repositories for a set of popular OSS projects.

Name	Apache	Ant	Python	Perl	PostgreSQL
Begin Date	1995-02-27	2000-01-12	1999-04-21	1999-03-01	1998-01-03
End Date	2005-07-13	2006-08-31	2006-07-27	2007-06-20	2007-03-01
Messages	101250	73157	66541	112514	132698
List Participants	2017	1960	1329	3621	3607
Files	1092	7682	4290	13308	6083
Developers	57	40	92	25	29
Commits	28517	58254	48318	92502	111847

**Table 1:** Information on the data gathered for the projects studied.

# Finding community structure

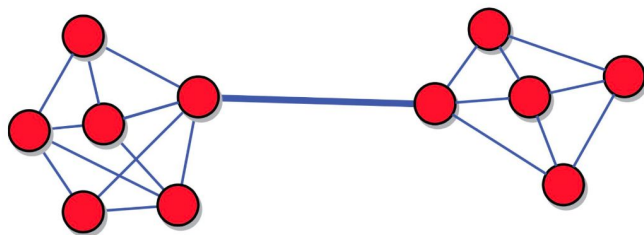
“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

# Finding community structure

“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

## 3.1. Bridge removal

Key idea: Find links with high betweenness and remove them.



Link betweenness defined similarly to node betweenness centrality in previous lecture – fraction of shortest paths that run through that link.

Link betweenness should be higher for bridges than for links inside a cluster.

# Finding community structure

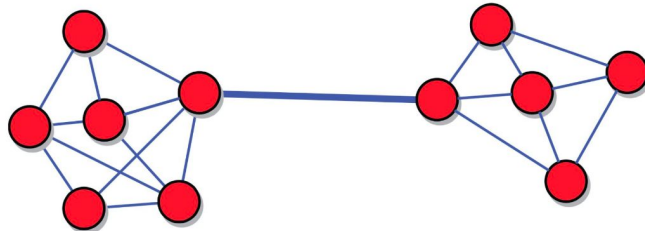
“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

## Girvan-Newman algorithm (similar to hierarchical clustering)

We start by calculating the betweenness for all links. Then, each iteration of the algorithm consists of two steps:

1. Remove the link with largest betweenness; in case of ties, one of them is picked at random.
2. Recalculate the betweenness of the remaining links.

The procedure ends when all links are removed and the nodes are isolated.



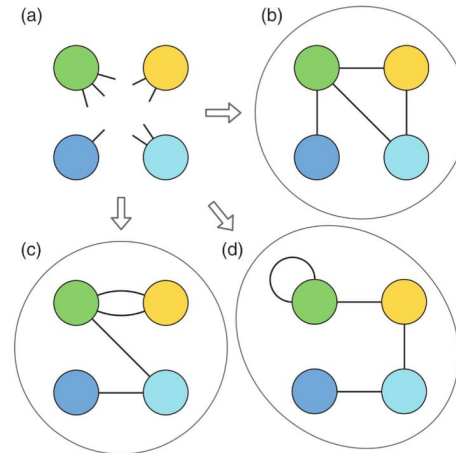
# Finding community structure

“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

## Modularity

The difference between the number of links internal to all clusters and the expected equivalent number in a randomized network.

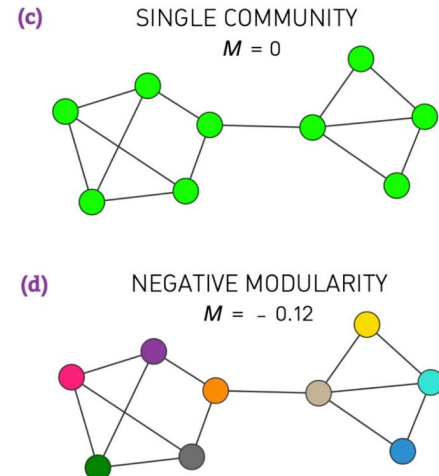
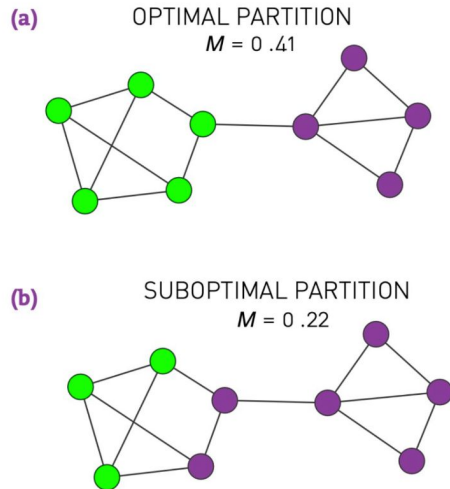
Randomization strategy: maintain number of nodes and degree sequence, shuffle links.



# Finding community structure

“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

The higher the modularity for a partition, the better the corresponding community structure



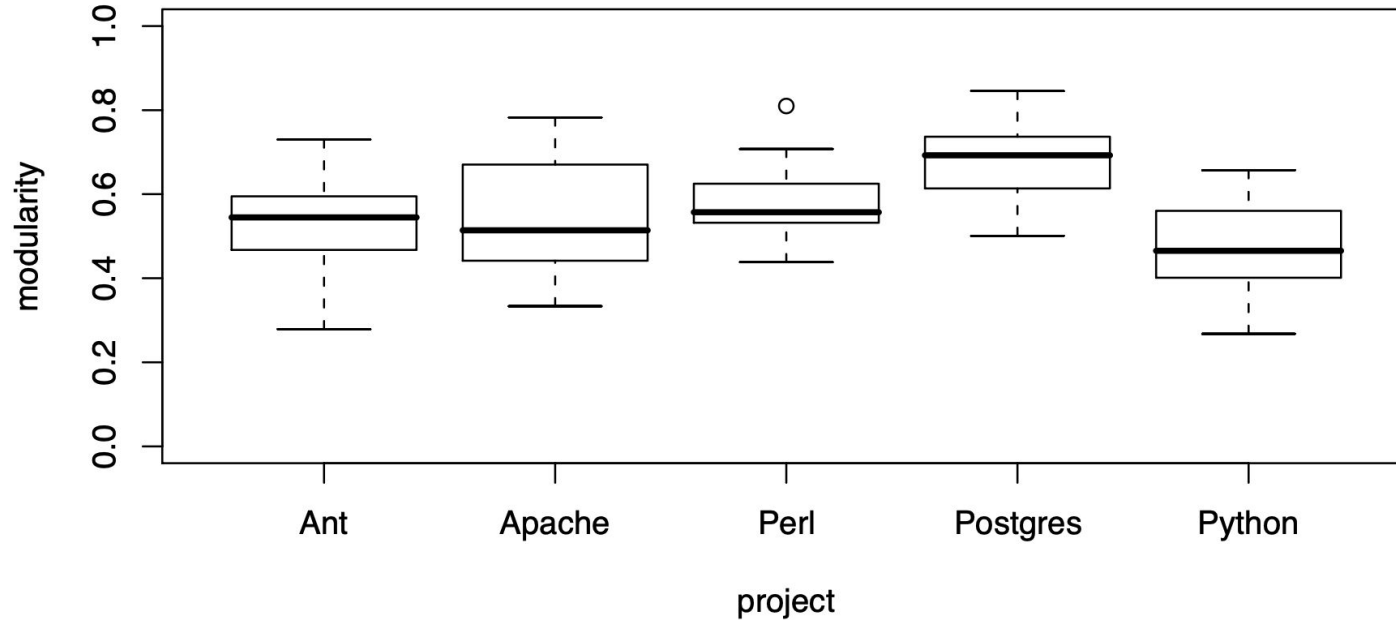
# Finding community structure

“Girvan and Newman’s original algorithm [...] doesn’t handle networks with weighted edges. Our social networks contain weighted edges, representing the number of emails exchanged between two participants in each time period. A high number of messages between a pair of participants should increase their likelihood of being in the same group.

[...] we modified our social networks by introducing one edge between each pair of nodes per email sent between them (i.e. creating a multi-edge network) and modified Newman’s algorithm above to handle multi-edge networks.”

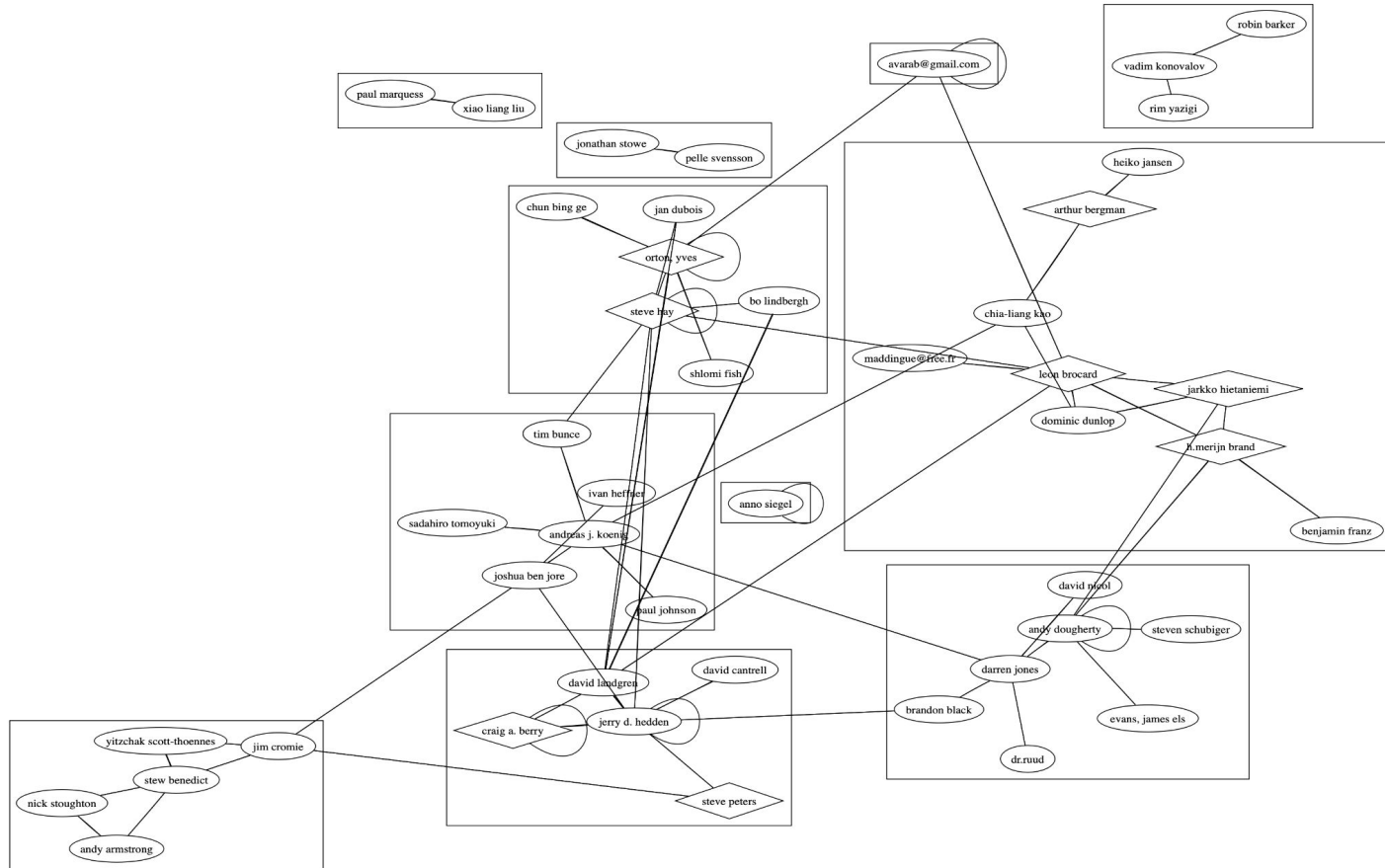
# Community structure exists

**Boxplots of Modularity in Projects**



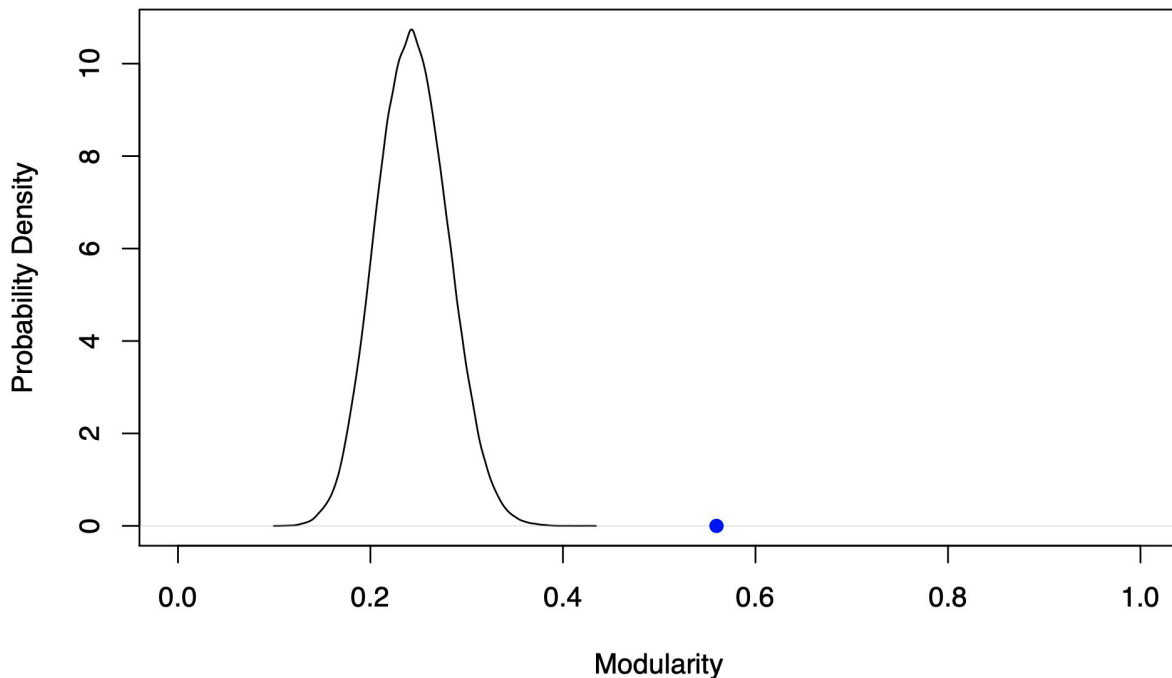


# The community structure of Perl from April to June 2007

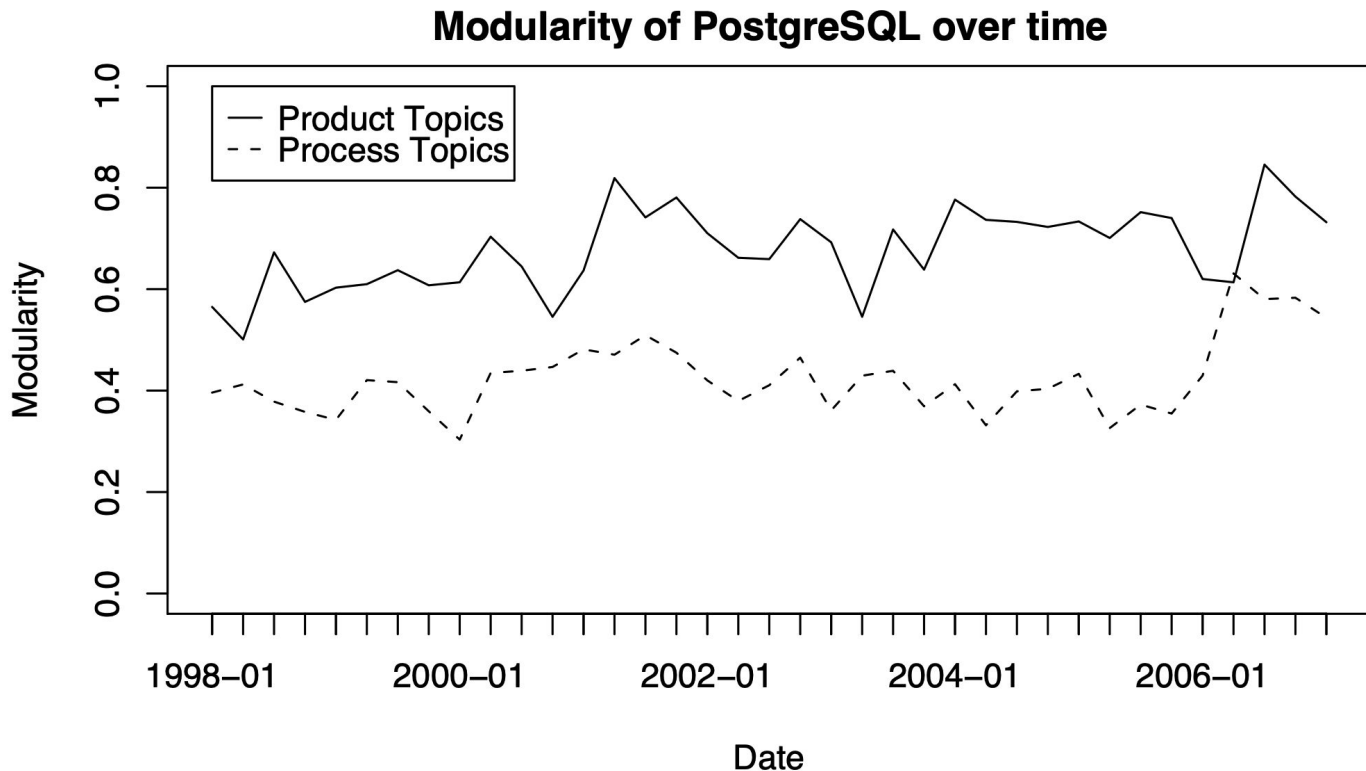


# The distribution of modularity values for 100,000 random graphs with the same degree distribution as the observed network.

**Ant, April to June of 2006**

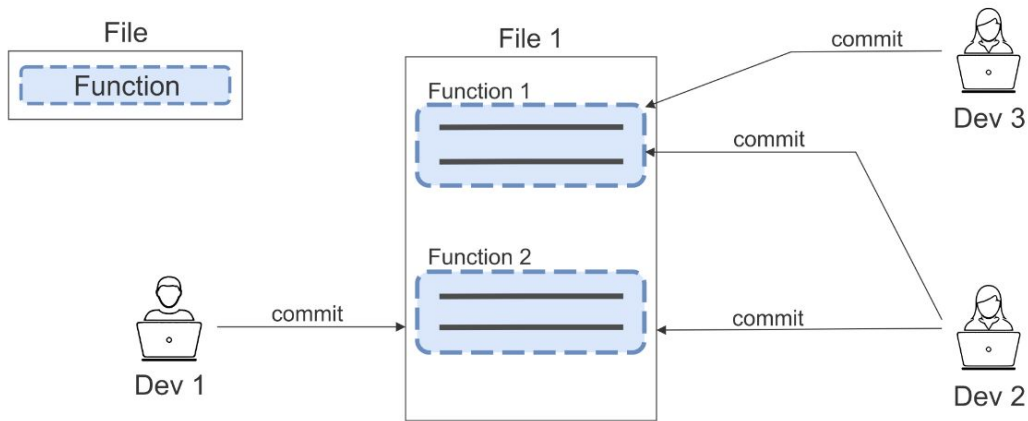


# Grouping into subcommunities is much stronger for discussions directly related to the source code.

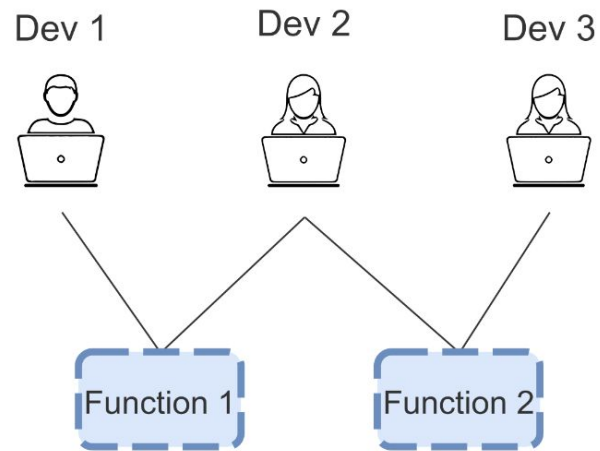


**“From developer networks to verified communities: A fine-grained approach” – Joblin et al, ICSE 2015**

# Developer activity (a) recorded in a version control system at the granularity of functions is abstracted as a two-mode network (b)



(a) Developer Activity



(b) Two-mode Network

# File-based vs function-based community detection

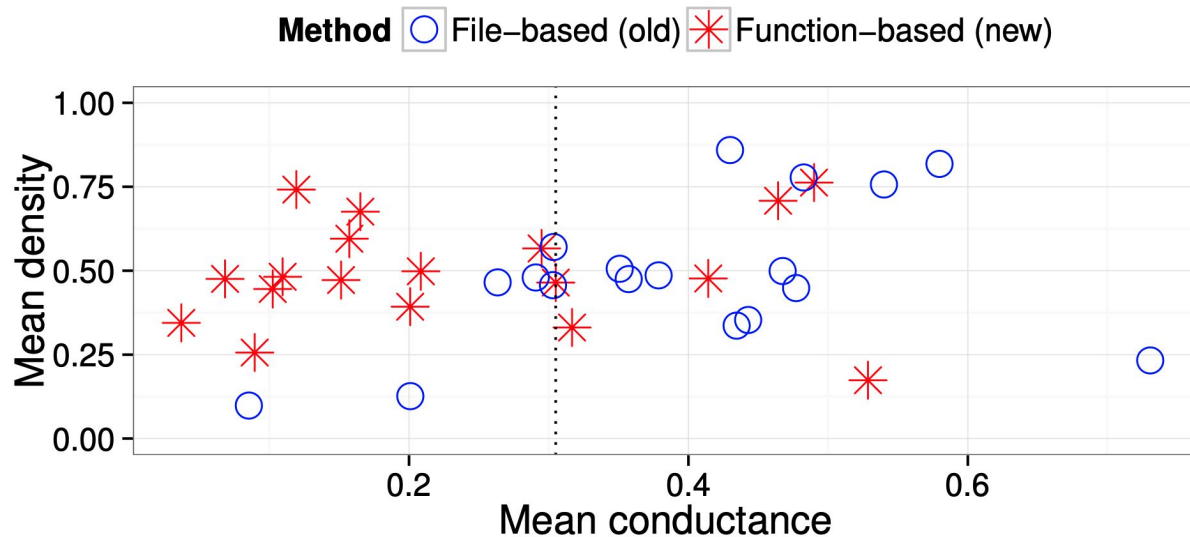


Fig. 2: Scatter plot of projects analyzed using both file-based and function-based methods for two different revisions. A clustering by crosses (left) and circles (right) is visible; the function-based approach is able to resolve more significant communities without compromising density.

# “Validity of Network Analyses in Open Source Projects” – Nia et al, MSR 2010

# OSS communication and coordination networks

“One can derive social networks from the online mailing list archives.

The nodes are the people sending messages on the list.

If a person A replies to a message from another person B, then there is an edge connecting the node representing A to that representing B.”



# Incorrect information flow due to temporal aggregation

How much temporal data aggregation can be tolerated before SNA results become unreliable?

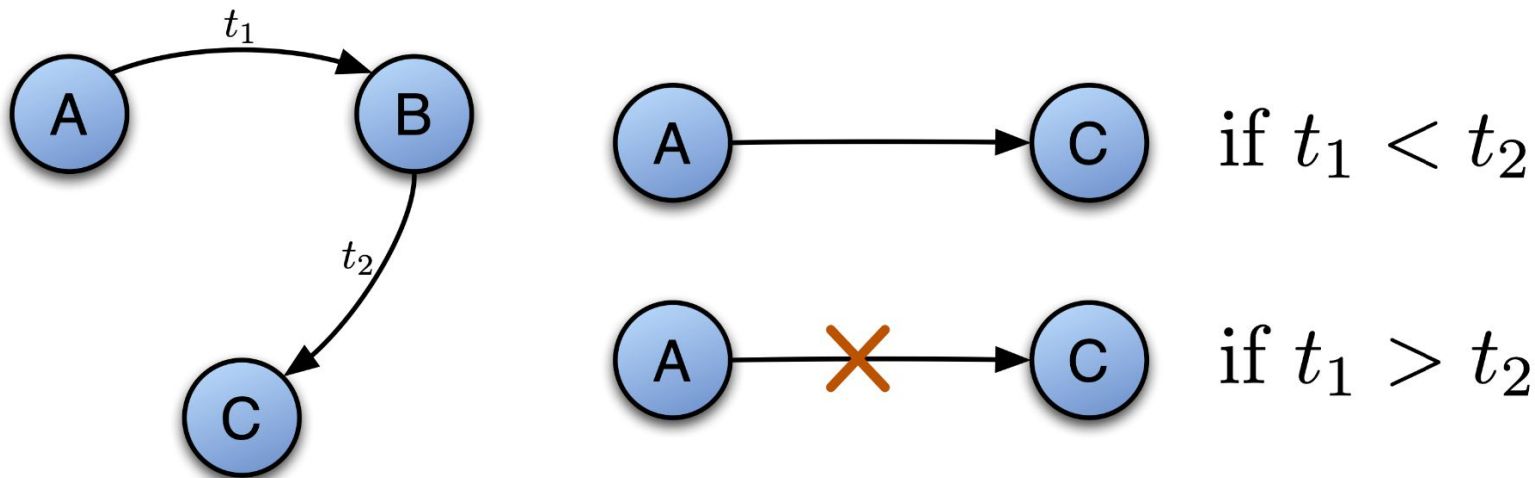


Fig. 1: The same topology, left, may apply to two different cases based on the order in which the messages were posted. If  $t_1 < t_2$ , then information can flow from A to C. But if  $t_1 > t_2$  no information can flow from A to C

# Information flow in the presence of inadequate or missing data

“Typically, social networks are derived from mailing list archives, using the ‘reply-to’ field in messages.

[...] If B read’s a message posted by A, but does not reply, then there is information flowing from A to B, but there is no way for us to know that.”

To what extent does missing data influence SNA metrics?

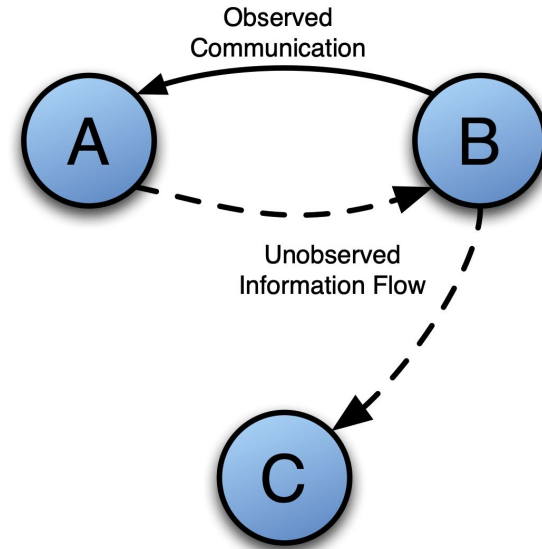


Fig. 2: Observed communication (solid edges) is evidence of information flow from B to A. However, C may read B’s message and B may have read A’s response, which indicates unobserved information flow (dashed edges).

# It doesn't matter?

“We find that while transitive faults can be as frequent as 50%, their frequency is highly dependent on the time interval of aggregation, and that even when very frequent, they do not change results from SNA analysis critically.”

## *B. Network Measures*

In this paper we use the following SNA measures.

- *Number of 2-paths (2P)* — The number of 2-paths through a node is a measure of local social status as defined previously [27].
- *Betweenness Centrality (BW)* — The betweenness centrality of a node is a function of the how many communication paths a node lies on and is often used a measure of global social status [28].
- *Clustering Coefficient (CC)* – The clustering coefficient measures the local connectivity density, or local structure in the graphs [29].

# Summary

... to be continued

Tons of data and research opportunities in OSS, join us!