Google Open Source

Who does the dev?

or

How we failed to make a taxonomy of open source contribution

Katie McLaughlin (they/them)
@glasnt on all the things

Ranting about contributions since 2015

opensource.com

A tool for tracking non-code GitHub contributions

By Katie McLaughlin

October 15, 2015

In today's open source world, t
incentives for contributing to p
attract tips, and once you becc
of perks, including free entry to

For smaller projects based on G
contributors for a project. Ther

But what about everything else

What about the hours spent in
design, testing, maintenance, a
learning? What about the meet
making those things happen? W
conversations?

MODEL VIEW CULTURE

A magazine about technology, culture and diversity.

HOME RESOURCES ALL ISSUES AUTHORS NEWS SUPPORT

ISSUE 32

Acknowledging Non-Coding Contributions

Giving people the recognition and respect they deserve is the start of helping evolve open source software into a more sustainable ecosystem.

—*by* Katie McLaughlin *on* February 1st, 2016

Open Source software development has a firm foothold in the technical world. The plethora of social networks around code, the thousands of conferences and community groups all attest to the many people working towards the common goals of free and open source software.

However, most of the time it's only the contributions of coders that are acknowledged. Software platforms will issue a press release about their latest major version, and append a generated list of registered users whose code was accepted in the last release. The contributors page on a software project's online profile only includes the usernames of programmers whose code has made it into the master branch.

https://opensource.com/life/15/10/octohat-github-non-code-contribution-tracker
https://modelviewculture.com/pieces/acknowledging-non-coding-contributions

ACROSS

Attributing Contributor Roles in Open Source Software

a subproject of OCEAN
UVM-Google Open Source Complex Ecosystems and Networks

Google Open Source

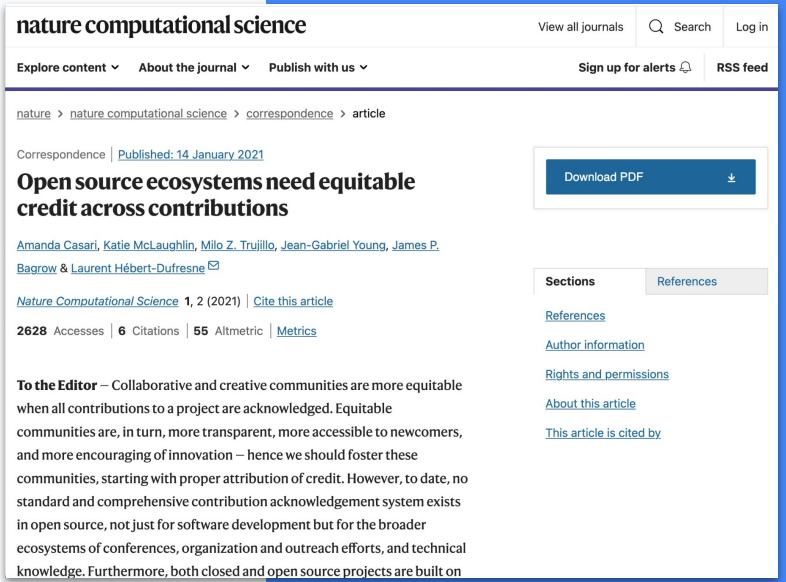
https://vermontcomplexsystems.org/partner/OCEAN/

Google Open Source

That time we corresponded

In Nature Computational Science

A **standard taxonomy** of recognized contributions will benefit all levels of open source.



nature computational science

View all journals | Search | Log in

Explore content | About the journal | Publish with us | Sign up for alerts | RSS feed

nature > nature computational science > correspondence > article

Correspondence | Published: 14 January 2021

Open source ecosystems need equitable credit across contributions

Amanda Casari, Katie McLaughlin, Milo Z. Trujillo, Jean-Gabriel Young, James P. Bagrow & Laurent Hébert-Dufresne

Nature Computational Science **1**, 2 (2021) | Cite this article

2628 Accesses | **6** Citations | **55** Altmetric | Metrics

Download PDF

Sections | References

References

Author information

Rights and permissions

About this article

This article is cited by

To the Editor – Collaborative and creative communities are more equitable when all contributions to a project are acknowledged. Equitable communities are, in turn, more transparent, more accessible to newcomers, and more encouraging of innovation – hence we should foster these communities, starting with proper attribution of credit. However, to date, no standard and comprehensive contribution acknowledgement system exists in open source, not just for software development but for the broader ecosystems of conferences, organization and outreach efforts, and technical knowledge. Furthermore, both closed and open source projects are built on

https://doi.org/10.1038/s43588-020-00011-w

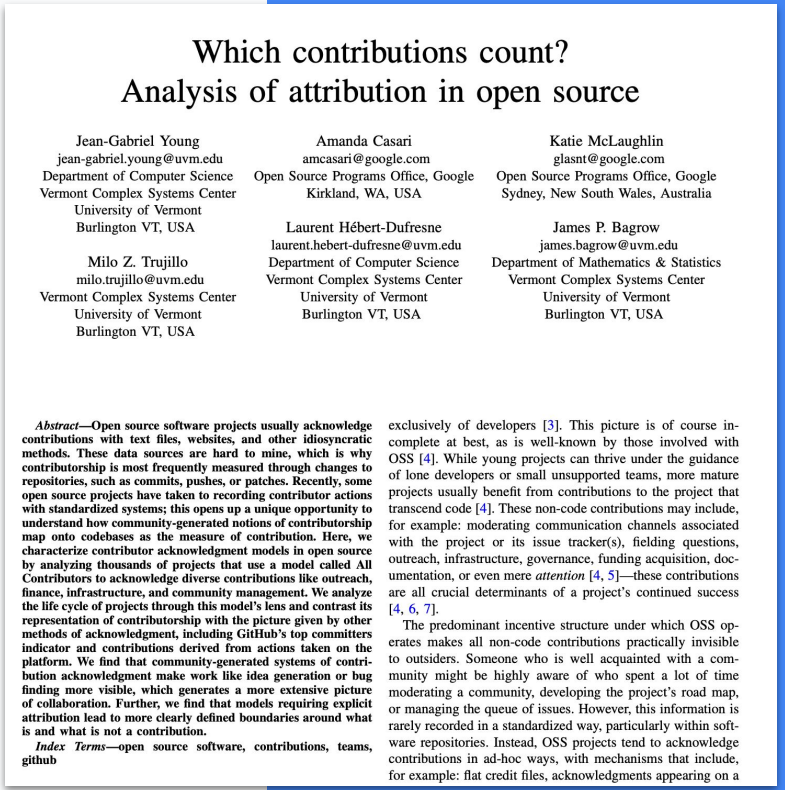
We'll get back to this.

Google Open Source

That time we analysed

many of the code bases

Community-generated systems of **contribution acknowledgment** make work like idea generation or bug finding more visible.



Which contributions count? Analysis of attribution in open source

Jean-Gabriel Young
jean-gabriel.young@uvm.edu
Department of Computer Science
Vermont Complex Systems Center
University of Vermont
Burlington VT, USA

Milo Z. Trujillo
milo.trujillo@uvm.edu
Vermont Complex Systems Center
University of Vermont
Burlington VT, USA

Amanda Casari
amcasari@google.com
Open Source Programs Office, Google
Kirkland, WA, USA

Laurent Hébert-Dufresne
laurent.hebert-dufresne@uvm.edu
Department of Computer Science
Vermont Complex Systems Center
University of Vermont
Burlington VT, USA

Katie McLaughlin
glasnt@google.com
Open Source Programs Office, Google
Sydney, New South Wales, Australia

James P. Bagrow
james.bagrow@uvm.edu
Department of Mathematics & Statistics
Vermont Complex Systems Center
University of Vermont
Burlington VT, USA

***Abstract*—Open source software projects usually acknowledge contributions with text files, websites, and other idiosyncratic methods. These data sources are hard to mine, which is why contributorship is most frequently measured through changes to repositories, such as commits, pushes, or patches. Recently, some open source projects have taken to recording contributor actions with standardized systems; this opens up a unique opportunity to understand how community-generated notions of contributorship map onto codebases as the measure of contribution. Here, we characterize contributor acknowledgment models in open source by analyzing thousands of projects that use a model called All Contributors to acknowledge diverse contributions like outreach, finance, infrastructure, and community management. We analyze the life cycle of projects through this model's lens and contrast its representation of contributorship with the picture given by other methods of acknowledgment, including GitHub's top committers indicator and contributions derived from actions taken on the platform. We find that community-generated systems of contribution acknowledgment make work like idea generation or bug finding more visible, which generates a more extensive picture of collaboration. Further, we find that models requiring explicit attribution lead to more clearly defined boundaries around what is and what is not a contribution.**

***Index Terms*—open source software, contributions, teams, github**

exclusively of developers [3]. This picture is of course incomplete at best, as is well-known by those involved with OSS [4]. While young projects can thrive under the guidance of lone developers or small unsupported teams, more mature projects usually benefit from contributions to the project that transcend code [4]. These non-code contributions may include, for example: moderating communication channels associated with the project or its issue tracker(s), fielding questions, outreach, infrastructure, governance, funding acquisition, documentation, or even mere *attention* [4, 5]—these contributions are all crucial determinants of a project's continued success [4, 6, 7].

The predominant incentive structure under which OSS operates makes all non-code contributions practically invisible to outsiders. Someone who is well acquainted with a community might be highly aware of who spent a lot of time moderating a community, developing the project's road map, or managing the queue of issues. However, this information is rarely recorded in a standardized way, particularly within software repositories. Instead, OSS projects tend to acknowledge contributions in ad-hoc ways, with mechanisms that include, for example: flat credit files, acknowledgments appearing on a

https://doi.org/10.1109/MSR52588.2021.00036

Again, we'll get back to this.

Google Open Source

That time we facilitated

a series of workshops

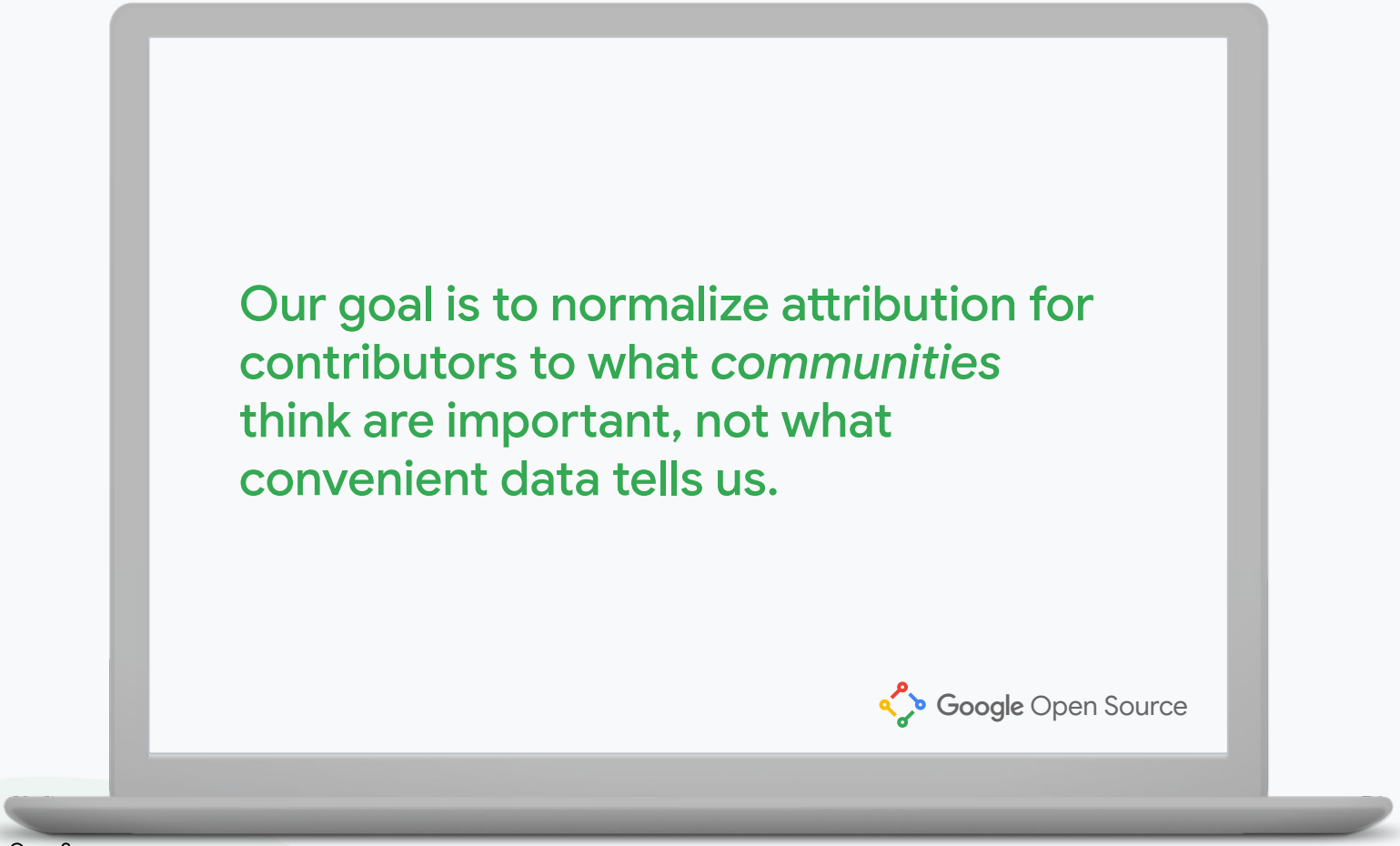
Source control is not the full picture of open source.

We are actively excluding and not counting work:

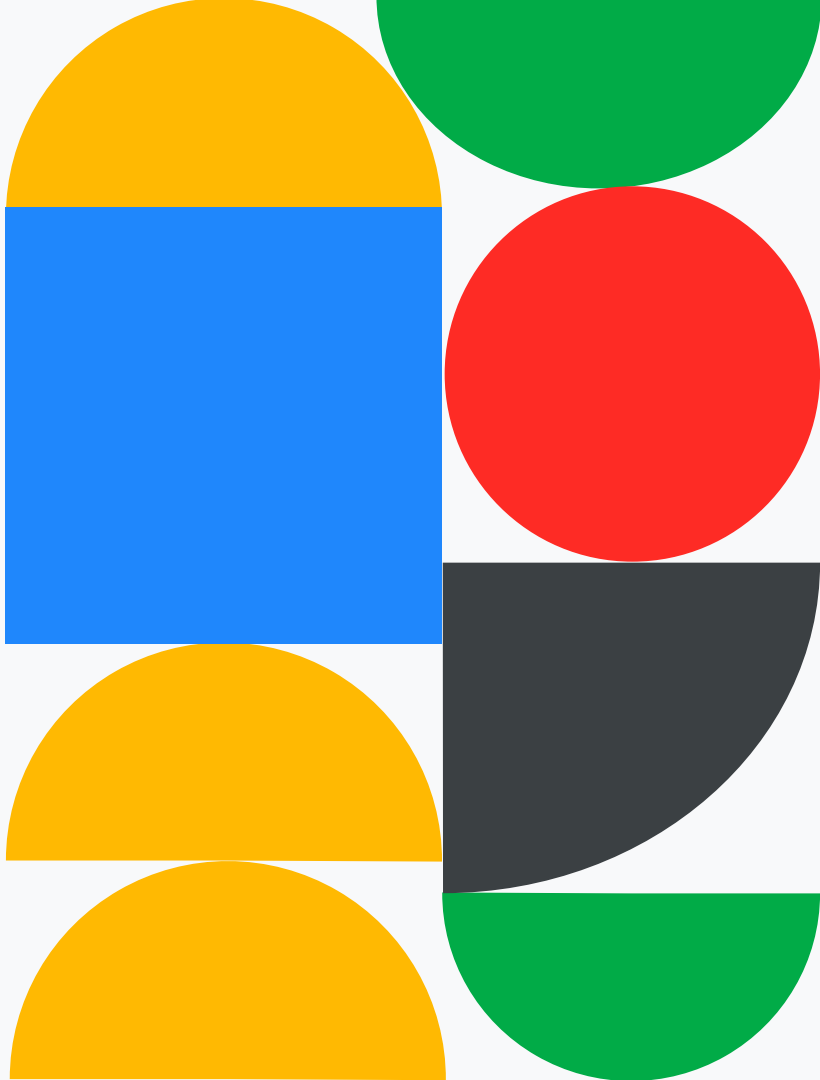
- Strategic decision making
- Project management
- Central build, infrastructure, and maintenance
- Documentation
- Community management
- Governance
- Financial management
- Event planning
- ….the list goes on

See References

Google Open Source


Our goal is to normalize attribution for contributors to what *communities* think are important, not what convenient data tells us.
Google Open Source

30 participants
6 workshops
2 time zones



Workshop focus groups

* Wikipedia definition

Repositories

Can be managed by source control or repository managers

Shared with an open source licence

Events*

A gathering of individuals who meet at an arranged place and time in order to discuss or engage in some common interest.

Organisations*

An entity - such as an institution, governing body, non-profit, association, or project team - comprising of one or more people and having a particular purpose.

Workshop unfocused groups

* Wikipedia definition

Repositories

Can be managed by source control or repository managers

Shared with an open source licence

Events*

A gathering of individuals who meet at an arranged place and time in order to discuss or engage in some common interest.

Organisations*

An entity - such as an institution, governing body, non-profit, association, or project team - comprising of one or more people and having a particular purpose.

Our initial categorisation raised more questions than answers.

What is an organisation?

Does a taxonomy make sense for small open source projects?

Are all repositories code?

What about when a project exceeds the scope of a repository?

What even is a project?

Workshop Outcomes

01

INDETERMINACY

No one definition for an open source "project" or "repository"

02

INCONSISTENCY

No definition for types of participation

03

INABILITY

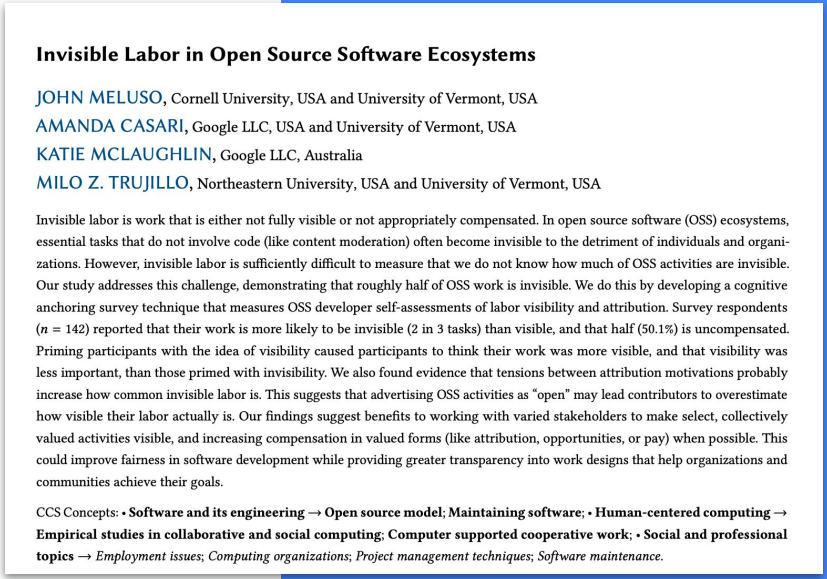
No way to create one universal taxonomy

Google Open Source

That time we surveyed

seven score open source contributors

Invisible labor is sufficiently difficult to measure that **we do not know** how much of [open source software] activities are invisible.



Invisible Labor in Open Source Software Ecosystems

JOHN MELUSO, Cornell University, USA and University of Vermont, USA
AMANDA CASARI, Google LLC, USA and University of Vermont, USA
KATIE MCLAUGHLIN, Google LLC, Australia
MILO Z. TRUJILLO, Northeastern University, USA and University of Vermont, USA

Invisible labor is work that is either not fully visible or not appropriately compensated. In open source software (OSS) ecosystems, essential tasks that do not involve code (like content moderation) often become invisible to the detriment of individuals and organizations. However, invisible labor is sufficiently difficult to measure that we do not know how much of OSS activities are invisible. Our study addresses this challenge, demonstrating that roughly half of OSS work is invisible. We do this by developing a cognitive anchoring survey technique that measures OSS developer self-assessments of labor visibility and attribution. Survey respondents (n = 142) reported that their work is more likely to be invisible (2 in 3 tasks) than visible, and that half (50.1%) is uncompensated. Priming participants with the idea of visibility caused participants to think their work was more visible, and that visibility was less important, than those primed with invisibility. We also found evidence that tensions between attribution motivations probably increase how common invisible labor is. This suggests that advertising OSS activities as "open" may lead contributors to overestimate how visible their labor actually is. Our findings suggest benefits to working with varied stakeholders to make select, collectively valued activities visible, and increasing compensation in valued forms (like attribution, opportunities, or pay) when possible. This could improve fairness in software development while providing greater transparency into work designs that help organizations and communities achieve their goals.

CCS Concepts: • **Software and its engineering** → **Open source model**; **Maintaining software**; • **Human-centered computing** → **Empirical studies in collaborative and social computing**; **Computer supported cooperative work**; • **Social and professional topics** → *Employment issues*; *Computing organizations*; *Project management techniques*; *Software maintenance*.

https://doi.org/10.48550/arXiv.2401.06889

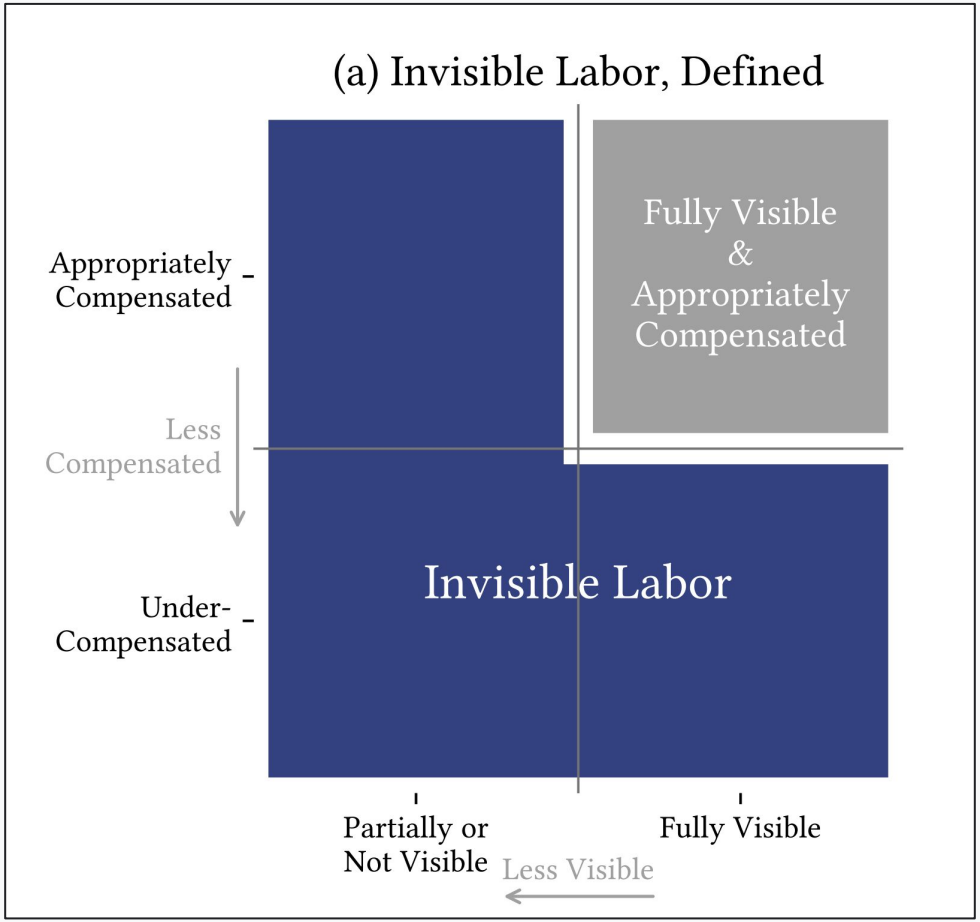


Fig. 1. (a) Invisible labor is labor that is not fully visible and/or undercompensated.

Survey Outcomes

01

INVISIBLE

Work is not visible, not compensated, or both

02

EXTRINSIC vs INTRINSIC

Motivations in tension with attribution

Motivation (🥕🌈)

Extrinsic

Intrinsic

Internalised Extrinsic

Attribution (🌊🦦)

Expressive

Instrumental

Non-Attribution

Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development https://doi.org/10.2307/41703471

03

COGNITIVE ANCHORING

Visibility anchoring shapes perceptions

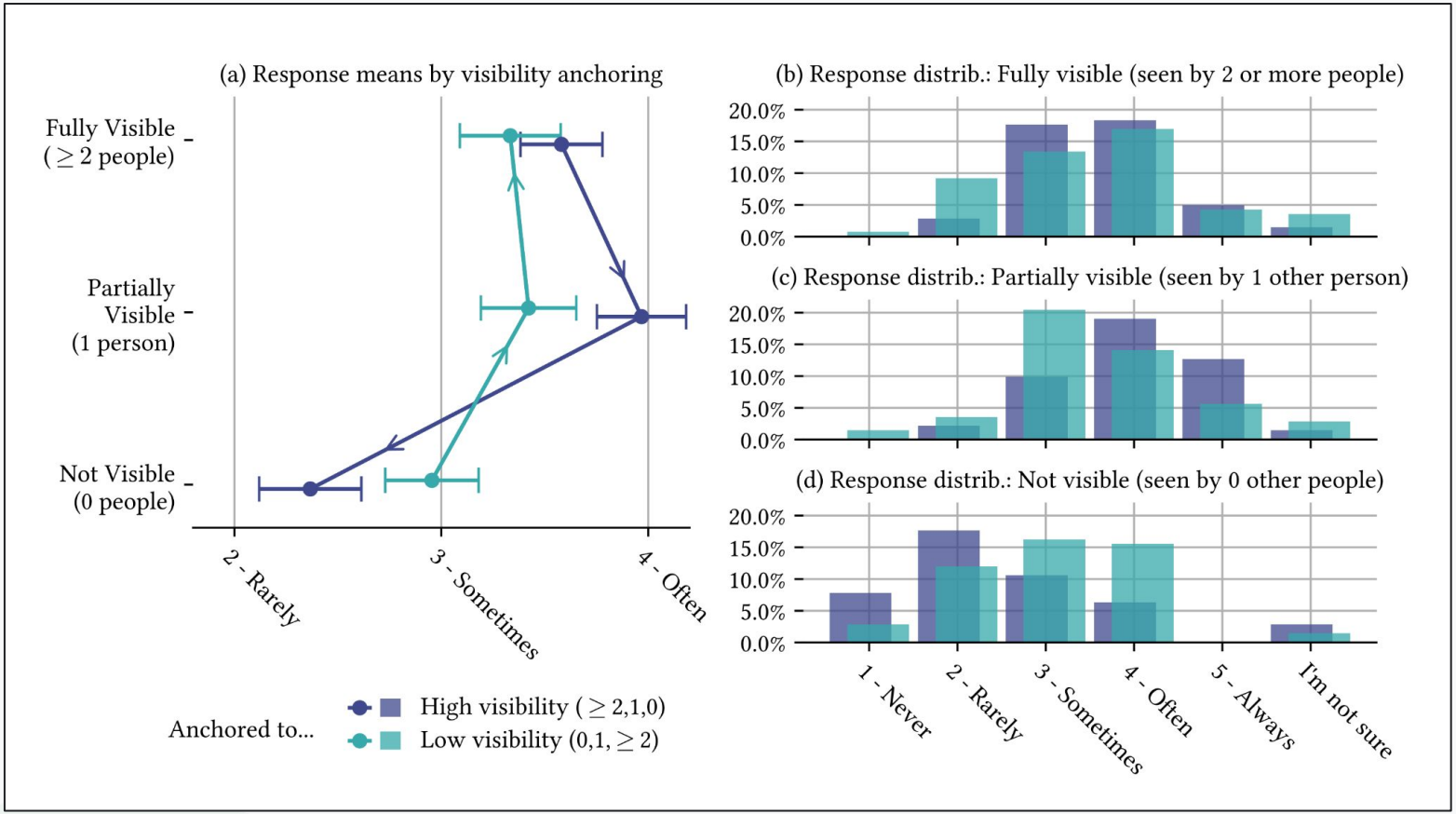


Fig. 4. Responses to "How often did [N people] know that you performed those tasks?"

Google Open Source

What this research means

for open source communities

01

DOCS OR IT DIDN'T HAPPEN

Documenting representation matters

02

RECOGNISE ANTI-PATTERNS

Easy statistics are not always meaningful

03

THE FEMINIST APPROACH

Ensure appropriate compensation for work

Google Open Source

Thank you.

Resources: whodoesthe.dev

Katie McLaughlin
@glasnt on all the things

Amanda Casari
@amcasari on all the things