

Research Says.....

Insights on Building, Leading, and Sustaining Open Source

Dr. Kaylea Champion – kaylea@uw.edu

<https://www.kayleachampion.com>

<https://communitydata.science>

University of Washington

FOSSY—August 2, 2024



What can research tell us about building, leading, and sustaining open source?

- Getting oriented

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
- What challenges do we face?

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
- What challenges do we face?
- What are our strategies to respond?

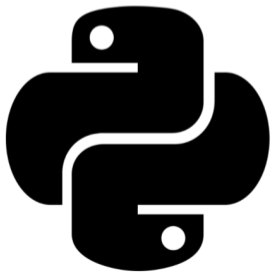
What can research tell us about building, leading, and sustaining open source?

- Getting oriented
- What challenges do we face?
- What are our strategies to respond?
- Takeaways





[AI Generated Using Canva.]



The world runs on open source software – including academic research.

- Digital infrastructure

The world runs on open source software – including academic research.

- Digital infrastructure
- Tools, languages

The world runs on open source software – including academic research.

- Digital infrastructure
- Tools, languages
- Setting, data source

The world runs on open source software – including academic research.

- Digital infrastructure
- Tools, languages
- Setting, data source
- Putting this research into practice can be a challenge – thank you for being here to help bridge this gap!

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- **What challenges do we face?**

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- **What challenges do we face?**
- What are our strategies to respond?
- Takeaways

Free Software is a lot of work!

- Scaling is hard



Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 175 (April 2021), 28 pages. <https://doi.org/10.1145/3449249>

Free Software is a lot of work!

- Scaling is hard
- Community might not be our original goal



Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 175 (April 2021), 28 pages. <https://doi.org/10.1145/3449249>

Free Software is a lot of work!

- Scaling is hard
- Community might not be our original goal
- User support can be a strain



Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 175 (April 2021), 28 pages.
<https://doi.org/10.1145/3449249>

Free Software is a lot of work!

- Scaling is hard
- Community might not be our original goal
- User support can be a strain
- Events build trust, improve teamwork, and make the experience more fun



Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 175 (April 2021), 28 pages. <https://doi.org/10.1145/3449249>

Free Software is a lot of work!

- Scaling is hard
- Community might not be our original goal
- User support can be a strain
- Events build trust, improve teamwork, and make the experience more fun
- Beware catastrophic success



Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 175 (April 2021), 28 pages. <https://doi.org/10.1145/3449249>

Free Software is a lot of work!

- Scaling is hard
- Community might not be our original goal
- User support can be a strain
- Events build trust, improve teamwork, and make the experience more fun
- Beware catastrophic success
- Automate, automate, automate



Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 175 (April 2021), 28 pages. <https://doi.org/10.1145/3449249>

Some critical infrastructure is neglected

- Leftpad and CrowdStrike remind us how interdependent we are.

bigskip Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

Some critical infrastructure is neglected

- Leftpad and CrowdStrike remind us how interdependent we are.
- Vulnerabilities like Heartbleed and Log4Shell indicate misalignment

bigskip Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

Some critical infrastructure is neglected

- Leftpad and CrowdStrike remind us how interdependent we are.
- Vulnerabilities like Heartbleed and Log4Shell indicate misalignment
- Thousands of important packages are low quality.

bigskip Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

Some critical infrastructure is neglected

- Leftpad and CrowdStrike remind us how interdependent we are.
- Vulnerabilities like Heartbleed and Log4Shell indicate misalignment
- Thousands of important packages are low quality.
- Misalignment risk increases with time.

bigskip Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- **What are our strategies to respond?**

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- **What are our strategies to respond?**
- Takeaways

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.

<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

- A sustainable community can produce popular but low quality software.

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.

<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

- A sustainable community can produce popular but low quality software.
- Great software might not outlast its maintainer.

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.

<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

- A sustainable community can produce popular but low quality software.
- Great software might not outlast its maintainer.
- We need both strong communities **and** alignment of importance and quality/security.

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.

<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

- A sustainable community can produce popular but low quality software.
- Great software might not outlast its maintainer.
- We need both strong communities **and** alignment of importance and quality/security.
- How do we do that?

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.
<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

- A sustainable community can produce popular but low quality software.
- Great software might not outlast its maintainer.
- We need both strong communities **and** alignment of importance and quality/security.
- How do we do that?
 - Awareness

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.
<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

What makes a difference?

What can we do to prevent and counter misalignment?

How do we build sustainable communities?

- A sustainable community can produce popular but low quality software.
- Great software might not outlast its maintainer.
- We need both strong communities **and** alignment of importance and quality/security.
- How do we do that?
 - Awareness
 - Intervention

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.
<https://doi.org/10.1109/SANER50967.2021.00043>

Alami, A., Pardo, R., & Linåker, J. (2024). Free open source communities sustainability: Does it make a difference in software quality? Empirical Software Engineering, 29(5), 114. <https://doi.org/10.1007/s10664-024-10529-6>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis
- Lifecycle analysis

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis
- Lifecycle analysis
- Four essential steps:

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.
<https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis
- Lifecycle analysis
- Four essential steps:
 - Measure importance

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.
<https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis
- Lifecycle analysis
- Four essential steps:
 - Measure importance
 - Measure quality (or security)

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399.
<https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis
- Lifecycle analysis
- Four essential steps:
 - Measure importance
 - Measure quality (or security)
 - Specify a relationship

Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

How can we detect misalignment of maintenance efforts?

- Ecosystem analysis
- Lifecycle analysis
- Four essential steps:
 - Measure importance
 - Measure quality (or security)
 - Specify a relationship
 - Measure deviations

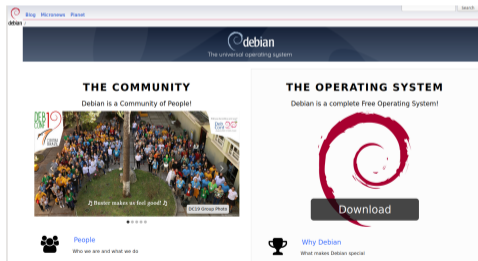
Champion, K., & Hill, B. M. (2021). Underproduction: An approach for measuring risk in open source software. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 388–399. <https://doi.org/10.1109/SANER50967.2021.00043>

Champion, K., & Hill, B. M. (2024). Sources of underproduction in open source software. The IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). <https://doi.org/10.48550/arXiv.2401.11281>

Example: Misalignment Analysis In Action

A worked example of this process in action!

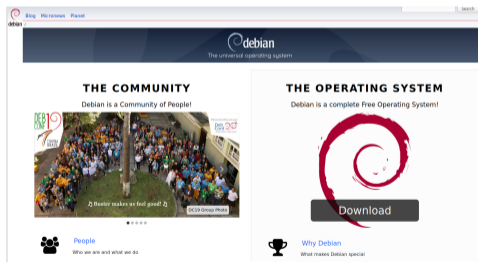
- Measure importance (**installations**)



Example: Misalignment Analysis In Action

A worked example of this process in action!

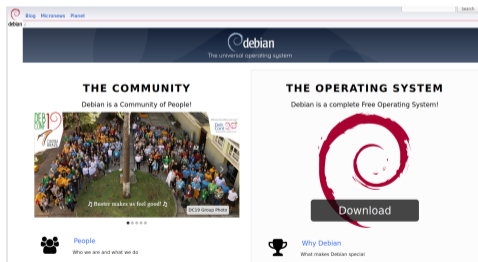
- Measure importance (**installations**)
- Measure quality (or security) (**pylint output**)



Example: Misalignment Analysis In Action

A worked example of this process in action!

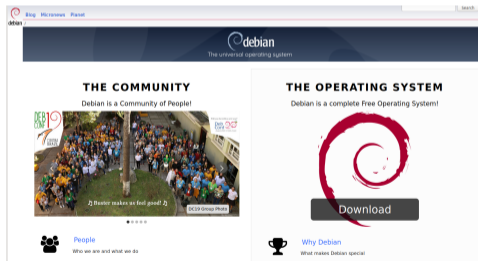
- Measure importance (**installations**)
- Measure quality (or security) (**pylint output**)
- Specify a relationship (**alignment of importance and quality**)



Example: Misalignment Analysis In Action

A worked example of this process in action!

- Measure importance (**installations**)
- Measure quality (or security) (**pylint output**)
- Specify a relationship (**alignment of importance and quality**)
- Measure deviations (**gaps and slopes**)



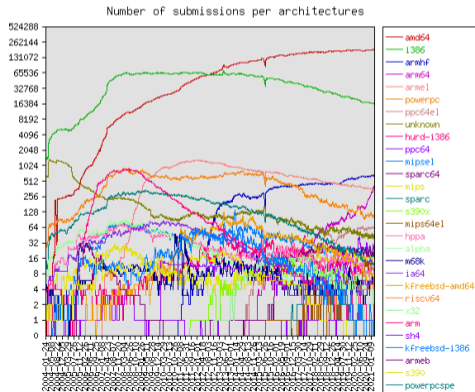
Example: Measuring Importance

POPCON

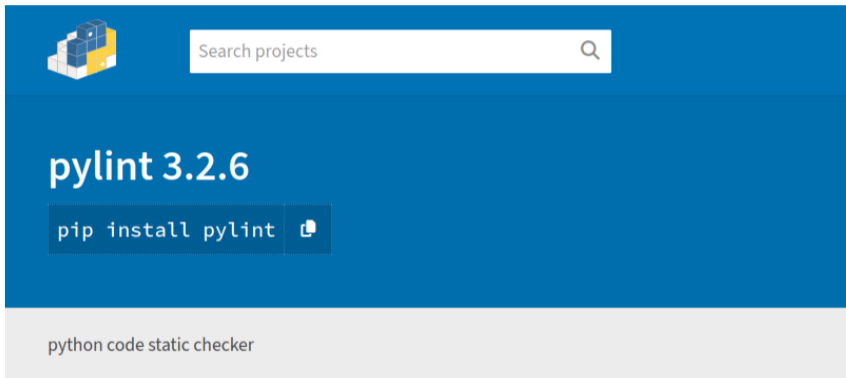
About Debian Getting Debian Support Developers' Corner

debian / debian popularity contest / all reports



Debian Popularity Contest



Example: Measuring Quality



The screenshot shows a project page for 'pylint 3.2.6'. At the top left is a logo consisting of several 3D cubes in blue, yellow, and white. To the right of the logo is a search bar with the text 'Search projects' and a magnifying glass icon. Below the search bar, the project name 'pylint 3.2.6' is displayed in large white text. Underneath the name is a dark blue button containing the command 'pip install pylint' and a copy icon. At the bottom of the page, there is a light gray bar with the text 'python code static checker'.

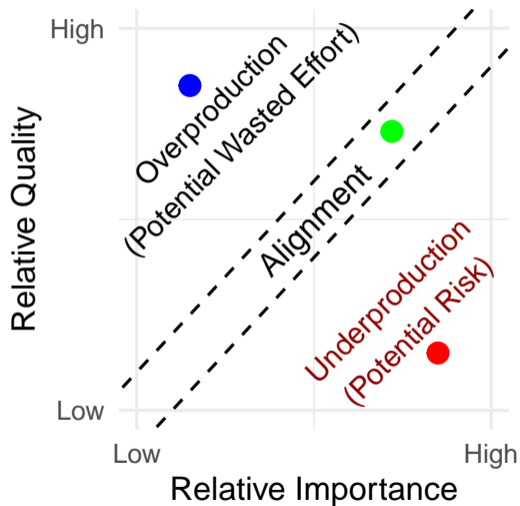
 Search projects 

pylint 3.2.6

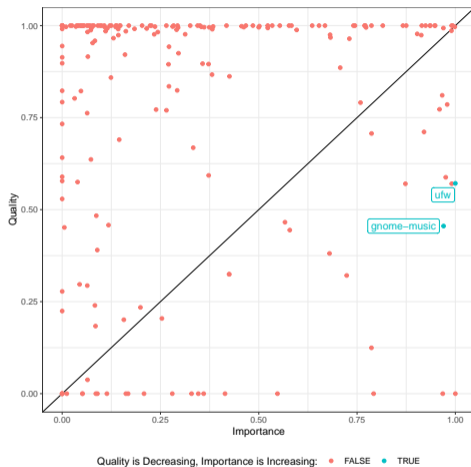
```
pip install pylint
```

python code static checker

Example: Specifying a relationship

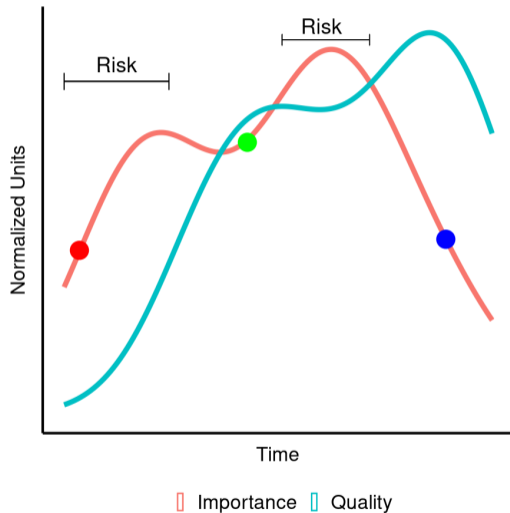


Example: Measuring Deviations

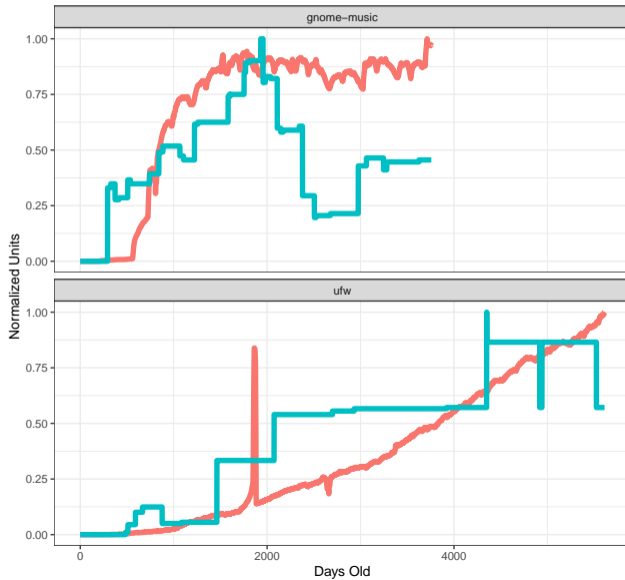


n = 201; 19% of these (39) are underproduced today.

Example: Measuring Deviations



Example: Measuring Deviations



Four Intervention Challenges

- Newcomers

Four Intervention Challenges

- Newcomers
- Funding

Four Intervention Challenges

- Newcomers
- Funding
- Governance

Four Intervention Challenges

- Newcomers
- Funding
- Governance
- Organizing our work

Intervention Challenge 1: Newcomers

This is a very large area of work!

- Newcomers are necessary for projects to be sustained

Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and Retaining OSS Contributors with a Maintainer Dashboard. In Software Engineering in Society (ICSE-SEIS'22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 5 pages.

<https://doi.org/10.1145/3510458.3513020>

Wenxin Xiao, Hao He, Weiwei Xu, Xin Tan, Jinhao Dong, and Minghui Zhou. 2022. Recommending Good First Issues in GitHub OSS Projects. In 44th International Conference on Software Engineering (ICSE '22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3510003.3510196>

H. Horiguchi, I. Omori and M. Ohira, "Onboarding to Open Source Projects with Good First Issues: A Preliminary Analysis," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 501-505, doi: 10.1109/SANER50967.2021.00054.

Intervention Challenge 1: Newcomers

This is a very large area of work!

- Newcomers are necessary for projects to be sustained
- Newcomers can be recruited with social media and directed using issue tagging.

Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and Retaining OSS Contributors with a Maintainer Dashboard. In Software Engineering in Society (ICSE-SEIS'22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 5 pages.

<https://doi.org/10.1145/3510458.3513020>

Wenxin Xiao, Hao He, Weiwei Xu, Xin Tan, Jinhao Dong, and Minghui Zhou. 2022. Recommending Good First Issues in GitHub OSS Projects. In 44th International Conference on Software Engineering (ICSE '22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3510003.3510196>

H. Horiguchi, I. Omori and M. Ohira, "Onboarding to Open Source Projects with Good First Issues: A Preliminary Analysis," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 501-505, doi: 10.1109/SANER50967.2021.00054.

Intervention Challenge 1: Newcomers

This is a very large area of work!

- Newcomers are necessary for projects to be sustained
- Newcomers can be recruited with social media and directed using issue tagging.
- But numbers alone aren't enough.

Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and Retaining OSS Contributors with a Maintainer Dashboard. In Software Engineering in Society (ICSE-SEIS'22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 5 pages.

<https://doi.org/10.1145/3510458.3513020>

Wenxin Xiao, Hao He, Weiwei Xu, Xin Tan, Jinhao Dong, and Minghui Zhou. 2022. Recommending Good First Issues in GitHub OSS Projects. In 44th International Conference on Software Engineering (ICSE '22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3510003.3510196>

H. Horiguchi, I. Omori and M. Ohira, "Onboarding to Open Source Projects with Good First Issues: A Preliminary Analysis," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 501-505, doi: 10.1109/SANER50967.2021.00054.

Intervention Challenge 1: Newcomers

This is a very large area of work!

- Newcomers are necessary for projects to be sustained
- Newcomers can be recruited with social media and directed using issue tagging.
- But numbers alone aren't enough.
- It's the experienced folks who make a big difference in sustainability...

Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and Retaining OSS Contributors with a Maintainer Dashboard. In Software Engineering in Society (ICSE-SEIS'22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 5 pages.

<https://doi.org/10.1145/3510458.3513020>

Wenxin Xiao, Hao He, Weiwei Xu, Xin Tan, Jinhao Dong, and Minghui Zhou. 2022. Recommending Good First Issues in GitHub OSS Projects. In 44th International Conference on Software Engineering (ICSE '22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3510003.3510196>

H. Horiguchi, I. Omori and M. Ohira, "Onboarding to Open Source Projects with Good First Issues: A Preliminary Analysis," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 501-505, doi: 10.1109/SANER50967.2021.00054.

Intervention Challenge 1: Newcomers

This is a very large area of work!

- Newcomers are necessary for projects to be sustained
- Newcomers can be recruited with social media and directed using issue tagging.
- But numbers alone aren't enough.
- It's the experienced folks who make a big difference in sustainability...
- ...and how do we get experienced folks?

Nadia Eghbal. 2020. Working In Public: The Making and Maintenance of Open Source Software. Stripe Press.

Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and Retaining OSS Contributors with a Maintainer Dashboard. In Software Engineering in Society (ICSE-SEIS'22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 5 pages.

<https://doi.org/10.1145/3510458.3513020>

Wenxin Xiao, Hao He, Weiwei Xu, Xin Tan, Jinhao Dong, and Minghui Zhou. 2022. Recommending Good First Issues in GitHub OSS Projects. In 44th International Conference on Software Engineering (ICSE '22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3510003.3510196>

H. Horiguchi, I. Omori and M. Ohira, "Onboarding to Open Source Projects with Good First Issues: A Preliminary Analysis," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 501-505, doi: 10.1109/SANER50967.2021.00054.

Intervention Challenge 2: Funding

- Large umbrella organizations are showing flat or declining activity levels.

Cassandra Overney, Jens Meinicke, Christian Kästner, and Bogdan Vasilescu. 2020. How to not get rich: an empirical study of donations in open source. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20). Association for Computing Machinery, New York, NY, USA, 1209–1221.

<https://doi.org/10.1145/3377811.3380410>

Cełkowski T, Jemielniak D, Macikowski K (2021) Free and Open Source Software organizations: A large-scale analysis of code, comments, and commits frequency. PLoS ONE 16(9): e0257192. <https://doi.org/10.1371/journal.pone.0257192>

Xunhui Zhang, Tao Wang, Yue Yu, Qiubing Zeng, Zhixing Li, and Huaimin Wang. 2022. Who, What, Why and How? Towards the Monetary Incentive in Crowd Collaboration: A Case Study of Github's Sponsor Mechanism. In CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3491102.3501822>

Yuxia Zhang, Mian Qin, Klaas-Jan Stol, Minghui Zhou, and Hui Liu. 2024. How Are Paid and Volunteer Open Source Developers Different? A Study of the Rust Project. In 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3597503.3639197>

Intervention Challenge 2: Funding

- Large umbrella organizations are showing flat or declining activity levels.
- Sponsorship may have little to no impact; perhaps modest short-term individual impact (tweeting may help).

Cassandra Overney, Jens Meinicke, Christian Kästner, and Bogdan Vasilescu. 2020. How to not get rich: an empirical study of donations in open source. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20). Association for Computing Machinery, New York, NY, USA, 1209–1221.

<https://doi.org/10.1145/3377811.3380410>

Chełkowski T, Jemielniak D, Macikowski K (2021) Free and Open Source Software organizations: A large-scale analysis of code, comments, and commits frequency. PLoS ONE 16(9): e0257192. <https://doi.org/10.1371/journal.pone.0257192>

Xunhui Zhang, Tao Wang, Yue Yu, Qiubing Zeng, Zhixing Li, and Huaimin Wang. 2022. Who, What, Why and How? Towards the Monetary Incentive in Crowd Collaboration: A Case Study of Github's Sponsor Mechanism. In CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3491102.3501822>

Yuxia Zhang, Mian Qin, Klaas-Jan Stol, Minghui Zhou, and Hui Liu. 2024. How Are Paid and Volunteer Open Source Developers Different? A Study of the Rust Project. In 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3597503.3639197>

Intervention Challenge 2: Funding

- Large umbrella organizations are showing flat or declining activity levels.
- Sponsorship may have little to no impact; perhaps modest short-term individual impact (tweeting may help).
- Paid contributors contribute differently than volunteers.

Cassandra Overney, Jens Meinicke, Christian Kästner, and Bogdan Vasilescu. 2020. How to not get rich: an empirical study of donations in open source. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20). Association for Computing Machinery, New York, NY, USA, 1209–1221.

<https://doi.org/10.1145/3377811.3380410>

Chełkowski T, Jemielniak D, Macikowski K (2021) Free and Open Source Software organizations: A large-scale analysis of code, comments, and commits frequency. PLoS ONE 16(9): e0257192. <https://doi.org/10.1371/journal.pone.0257192>

Xunhui Zhang, Tao Wang, Yue Yu, Qiubing Zeng, Zhixing Li, and Huaimin Wang. 2022. Who, What, Why and How? Towards the Monetary Incentive in Crowd Collaboration: A Case Study of Github's Sponsor Mechanism. In CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3491102.3501822>

Yuxia Zhang, Mian Qin, Klaas-Jan Stol, Minghui Zhou, and Hui Liu. 2024. How Are Paid and Volunteer Open Source Developers Different? A Study of the Rust Project. In 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3597503.3639197>

Intervention Challenge 2: Funding

- Large umbrella organizations are showing flat or declining activity levels.
- Sponsorship may have little to no impact; perhaps modest short-term individual impact (tweeting may help).
- Paid contributors contribute differently than volunteers.
- Figuring out how to spend funds may be even more challenging than raising them (event / travel is a common approach here).

Cassandra Overney, Jens Meinicke, Christian Kästner, and Bogdan Vasilescu. 2020. How to not get rich: an empirical study of donations in open source. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20). Association for Computing Machinery, New York, NY, USA, 1209–1221.

<https://doi.org/10.1145/3377811.3380410>

Chełkowski T, Jemielniak D, Macikowski K (2021) Free and Open Source Software organizations: A large-scale analysis of code, comments, and commits frequency. PLoS ONE 16(9): e0257192. <https://doi.org/10.1371/journal.pone.0257192>

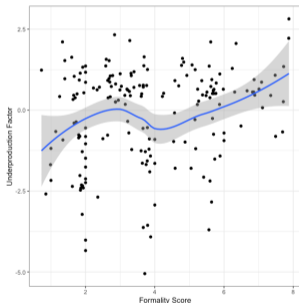
Xunhui Zhang, Tao Wang, Yue Yu, Qiubing Zeng, Zhixing Li, and Huaimin Wang. 2022. Who, What, Why and How? Towards the Monetary Incentive in Crowd Collaboration: A Case Study of Github's Sponsor Mechanism. In CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3491102.3501822>

Yuxia Zhang, Mian Qin, Klaas-Jan Stol, Minghui Zhou, and Hui Liu. 2024. How Are Paid and Volunteer Open Source Developers Different? A Study of the Rust Project. In 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages.

<https://doi.org/10.1145/3597503.3639197>

Intervention Challenge 3: Governance

- We may not always follow our own rules when it comes to granting power.



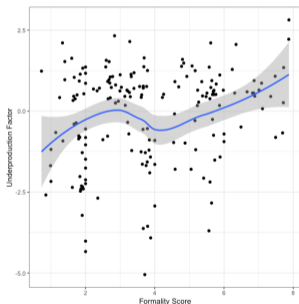
Gaughan, M., Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*.

Mahasweta Chakraborti, Curtis Atkisson, Stefan Stănciulescu, Vladimir Filkov, and Seth Frey. 2024. Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11– 16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 26 pages.

<https://doi.org/10.1145/3613904.3641980>

Intervention Challenge 3: Governance

- We may not always follow our own rules when it comes to granting power.
- Formality overall increases risk or use of formal processes not significant.



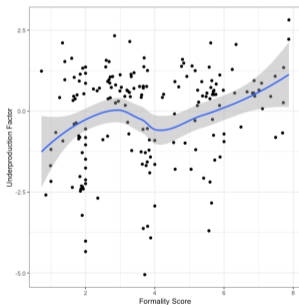
Gaughan, M., Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*.

Mahasweta Chakraborti, Curtis Atkisson, Stefan Stănciulescu, Vladimir Filkov, and Seth Frey. 2024. Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11– 16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 26 pages.

<https://doi.org/10.1145/3613904.3641980>

Intervention Challenge 3: Governance

- We may not always follow our own rules when it comes to granting power.
- Formality overall increases risk or use of formal processes not significant.
- But! Sharing power is associated with lower underproduction.



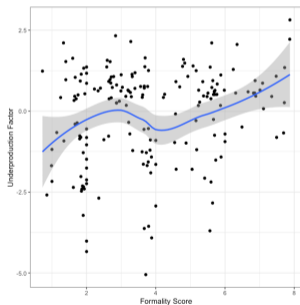
Gaughan, M., Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*.

Mahasweta Chakraborti, Curtis Atkisson, Stefan Stănciulescu, Vladimir Filkov, and Seth Frey. 2024. Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11– 16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 26 pages.

<https://doi.org/10.1145/3613904.3641980>

Intervention Challenge 3: Governance

- We may not always follow our own rules when it comes to granting power.
- Formality overall increases risk or use of formal processes not significant.
- But! Sharing power is associated with lower underproduction.
- More policies on a topic doesn't mean more effort.



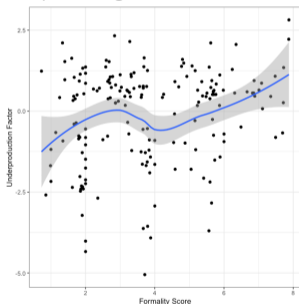
Gaughan, M., Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*.

Mahasweta Chakraborti, Curtis Atkisson, Stefan Stănciulescu, Vladimir Filkov, and Seth Frey. 2024. Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11– 16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 26 pages.

<https://doi.org/10.1145/3613904.3641980>

Intervention Challenge 3: Governance

- We may not always follow our own rules when it comes to granting power.
- Formality overall increases risk or use of formal processes not significant.
- But! Sharing power is associated with lower underproduction.
- More policies on a topic doesn't mean more effort.
- Reporting can be burdensome.



Gaughan, M., Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*.

Mahasweta Chakraborti, Curtis Atkisson, Stefan Stănciulescu, Vladimir Filkov, and Seth Frey. 2024. Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11– 16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 26 pages.

<https://doi.org/10.1145/3613904.3641980>

Intervention Challenge 4: Organizing our work

Coordination can be costly!

- Automation is helpful but can run up bills (CI/CD waste)

Irina Zaks, Timothy Lehnen, Aaron Zaks (2024). Improving Velocity of Code Contributions in Open Source. Computer (IEEE computer society magazine) DOI: 10.1109/MC.2023.3327893 Nimmi Rashinika Weeraddana, Mahmoud Alfadel, and Shane McIntosh.

2024. Dependency-Induced Waste in Continuous Integration: An Empirical Study of Unused Dependencies in the npm Ecosystem. Proc. ACM Softw. Eng. 1, FSE, Article 116 (July 2024), 24 pages. <https://doi.org/10.1145/3660823>

Intervention Challenge 4: Organizing our work

Coordination can be costly!

- Automation is helpful but can run up bills (CI/CD waste)
- Document decisions and acceptance criteria

Irina Zaks, Timothy Lehnen, Aaron Zaks (2024). Improving Velocity of Code Contributions in Open Source. Computer (IEEE computer society magazine) DOI: 10.1109/MC.2023.3327893 Nimmi Rashinika Weeraddana, Mahmoud Alfadel, and Shane McIntosh.

2024. Dependency-Induced Waste in Continuous Integration: An Empirical Study of Unused Dependencies in the npm Ecosystem. Proc. ACM Softw. Eng. 1, FSE, Article 116 (July 2024), 24 pages. <https://doi.org/10.1145/3660823>

Intervention Challenge 4: Organizing our work

Coordination can be costly!

- Automation is helpful but can run up bills (CI/CD waste)
- Document decisions and acceptance criteria
- “Review then commit” slows pace.

Irina Zaks, Timothy Lehnen, Aaron Zaks (2024). Improving Velocity of Code Contributions in Open Source. Computer (IEEE computer society magazine) DOI: 10.1109/MC.2023.3327893 Nimmi Rashinika Weeraddana, Mahmoud Alfadel, and Shane McIntosh.

2024. Dependency-Induced Waste in Continuous Integration: An Empirical Study of Unused Dependencies in the npm Ecosystem. Proc. ACM Softw. Eng. 1, FSE, Article 116 (July 2024), 24 pages. <https://doi.org/10.1145/3660823>

The story so far...

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- What are our strategies to respond?
 - Four areas of challenging interventions: newcomers, funding, governance, organizing work

The story so far...

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- What are our strategies to respond?
 - Four areas of challenging interventions: newcomers, funding, governance, organizing work
 - **Three areas of immediate action: data, toxicity, and valuing one another**

The story so far...

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- What are our strategies to respond?
 - Four areas of challenging interventions: newcomers, funding, governance, organizing work
 - **Three areas of immediate action: data, toxicity, and valuing one another**
- In Summary

Three areas of immediate action

- Collect data and have a plan to use it.

Three areas of immediate action

- Collect data and have a plan to use it.
- Minimize toxicity.

Three areas of immediate action

- Collect data and have a plan to use it.
- Minimize toxicity.
- Value your fellow contributors.

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because...'

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because...'
- Engage with metrics communities and researchers.

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because....'
- Engage with metrics communities and researchers.
 - Reach out! The Community Data Science Collective is a coalition of researchers who care about communities in all their dimensions!

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because...'
- Engage with metrics communities and researchers.
 - Reach out! The Community Data Science Collective is a coalition of researchers who care about communities in all their dimensions!
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because....'
- Engage with metrics communities and researchers.
 - Reach out! The Community Data Science Collective is a coalition of researchers who care about communities in all their dimensions!
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.
- Open communication channels with users!
 - quality

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because...'
- Engage with metrics communities and researchers.
 - Reach out! The Community Data Science Collective is a coalition of researchers who care about communities in all their dimensions!
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.
- Open communication channels with users!
 - quality
 - security

Action area 1: Collect data and have a plan to use it

- Keeping logs and records can help a lot.
 - Historical interpretation and metadata needed!
 - 'Oh, right there data is missing/spiked because...'
- Engage with metrics communities and researchers.
 - Reach out! The Community Data Science Collective is a coalition of researchers who care about communities in all their dimensions!
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.
- Open communication channels with users!
 - quality
 - security
 - importance (where is my thing being used?)

Action area 2: Minimize Toxicity

Toxicity matters!

- Toxic comments drive people away and scare off potential newcomers.

Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The “Shut the f**k up” Phenomenon: Characterizing Incivility in Open Source Code Review Discussions. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 353 (October 2021), 35 pages. <https://doi.org/10.1145/3479497>

Huilian Sophie Qiu, Bogdan Vasilescu, Christian Kastner, Carolyn Egelman, Ciera Jaspan, Emerson Murphy-Hill. 2022. Detecting Interpersonal Conflict in Issues and Code Review: Cross Pollinating Open- and Closed-Source Approaches. In *Software Engineering in Society (ICSE-SEIS'22)*, May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 15 pages. <https://cmustrudel.github.io/papers/seis22pushback.pdf>

M. Chouchen et al., “Anti-patterns in Modern Code Review: Symptoms and Prevalence,” 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 531-535, doi: 10.1109/SANER50967.2021.00060.

Jack Jamieson, Naomi Yamashita, and Eureka Foong. 2024. Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3597503.3623340>

Action area 2: Minimize Toxicity

Toxicity matters!

- Toxic comments drive people away and scare off potential newcomers.
- Experienced contributors look for toxicity and avoid joining because of it.

Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The “Shut the f**k up” Phenomenon: Characterizing Incivility in Open Source Code Review Discussions. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 353 (October 2021), 35 pages. <https://doi.org/10.1145/3479497>

Huilian Sophie Qiu, Bogdan Vasilescu, Christian Kastner, Carolyn Egelman, Ciera Jaspan, Emerson Murphy-Hill. 2022. Detecting Interpersonal Conflict in Issues and Code Review: Cross Pollinating Open- and Closed-Source Approaches. In *Software Engineering in Society (ICSE-SEIS'22)*, May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 15 pages. <https://cmustrudel.github.io/papers/seis22pushback.pdf>

M. Chouchen et al., “Anti-patterns in Modern Code Review: Symptoms and Prevalence,” 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 531-535, doi: 10.1109/SANER50967.2021.00060.

Jack Jamieson, Naomi Yamashita, and Eureka Foong. 2024. Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3597503.3623340>

Action area 2: Minimize Toxicity

Toxicity matters!

- Toxic comments drive people away and scare off potential newcomers.
- Experienced contributors look for toxicity and avoid joining because of it.
- Shallow, disengaged, toxic code reviews waste time and drive people away.

Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The “Shut the f**k up” Phenomenon: Characterizing Incivility in Open Source Code Review Discussions. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 353 (October 2021), 35 pages. <https://doi.org/10.1145/3479497>

Huilian Sophie Qiu, Bogdan Vasilescu, Christian Kastner, Carolyn Egelman, Ciera Jaspan, Emerson Murphy-Hill. 2022. Detecting Interpersonal Conflict in Issues and Code Review: Cross Pollinating Open- and Closed-Source Approaches. In *Software Engineering in Society (ICSE-SEIS'22)*, May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 15 pages. <https://cmustrudel.github.io/papers/seis22pushback.pdf>

M. Chouchen et al., “Anti-patterns in Modern Code Review: Symptoms and Prevalence,” 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 531-535, doi: 10.1109/SANER50967.2021.00060.

Jack Jamieson, Naomi Yamashita, and Eureka Foong. 2024. Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3597503.3623340>

Action area 2: Minimize Toxicity

Toxicity matters!

- Toxic comments drive people away and scare off potential newcomers.
- Experienced contributors look for toxicity and avoid joining because of it.
- Shallow, disengaged, toxic code reviews waste time and drive people away.
- Emphasizing the shared goal of the project helps retention and joining.

Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The “Shut the f**k up” Phenomenon: Characterizing Incivility in Open Source Code Review Discussions. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 353 (October 2021), 35 pages. <https://doi.org/10.1145/3479497>

Huilian Sophie Qiu, Bogdan Vasilescu, Christian Kastner, Carolyn Egelman, Ciera Jaspan, Emerson Murphy-Hill. 2022. Detecting Interpersonal Conflict in Issues and Code Review: Cross Pollinating Open- and Closed-Source Approaches. In *Software Engineering in Society (ICSE-SEIS'22)*, May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 15 pages. <https://cmustrudel.github.io/papers/seis22pushback.pdf>

M. Chouchen et al., “Anti-patterns in Modern Code Review: Symptoms and Prevalence,” 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 531-535, doi: 10.1109/SANER50967.2021.00060.

Jack Jamieson, Naomi Yamashita, and Eureka Foong. 2024. Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3597503.3623340>

Action Area 3: Valuing Contributors

- Focused code contributors build sustainability.

Wenxin Xiao, Hao He, Weiwei Xu, Yuxia Zhang, and Minghui Zhou. 2023. How Early Participation Determines Long-Term Sustained Activity in GitHub Projects?. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '23), December 3–9, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3611643.3616349>

E. Dias, P. Meirelles, F. Castor, I. Steinmacher, I. Wiese and G. Pinto, “What Makes a Great Maintainer of Open Source Projects?” 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 982-994, doi: 10.1109/ICSE43902.2021.00093.

Action Area 3: Valuing Contributors

- Focused code contributors build sustainability.
- Wide-ranging non-code contributors build sustainability.

Wenxin Xiao, Hao He, Weiwei Xu, Yuxia Zhang, and Minghui Zhou. 2023. How Early Participation Determines Long-Term Sustained Activity in GitHub Projects?. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '23), December 3–9, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3611643.3616349>

E. Dias, P. Meirelles, F. Castor, I. Steinmacher, I. Wiese and G. Pinto, “What Makes a Great Maintainer of Open Source Projects?” 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 982-994, doi: 10.1109/ICSE43902.2021.00093.

Action Area 3: Valuing Contributors

- Focused code contributors build sustainability.
- Wide-ranging non-code contributors build sustainability.
- Technical chops are needed, but **communication** is what characterizes great maintainers.

Wenxin Xiao, Hao He, Weiwei Xu, Yuxia Zhang, and Minghui Zhou. 2023. How Early Participation Determines Long-Term Sustained Activity in GitHub Projects?. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '23), December 3–9, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3611643.3616349>

E. Dias, P. Meirelles, F. Castor, I. Steinmacher, I. Wiese and G. Pinto, “What Makes a Great Maintainer of Open Source Projects?” 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 982-994, doi: 10.1109/ICSE43902.2021.00093.

Action Area 3: Valuing Contributors

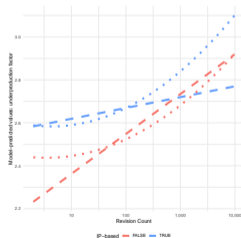
- Focused code contributors build sustainability.
- Wide-ranging non-code contributors build sustainability.
- Technical chops are needed, but **communication** is what characterizes great maintainers.
- Retaining people in the community helps fight misalignment (cue example!).

Wenxin Xiao, Hao He, Weiwei Xu, Yuxia Zhang, and Minghui Zhou. 2023. How Early Participation Determines Long-Term Sustained Activity in GitHub Projects?. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '23), December 3–9, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3611643.3616349>

E. Dias, P. Meirelles, F. Castor, I. Steinmacher, I. Wiese and G. Pinto, “What Makes a Great Maintainer of Open Source Projects?” 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 982-994, doi: 10.1109/ICSE43902.2021.00093.

Action Area 3: Valuing Contributors (another worked example)

- Underproduction in Wikipedia

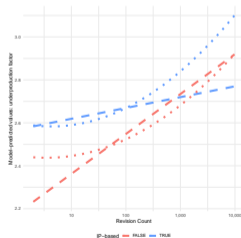


K Champion, BM Hill (2024) Countering underproduction of peer produced goods. *New Media & Society*

Aaron Shaw, Benjamin M. Hill, Laboratories of Oligarchy? How the Iron Law Extends to Peer Production, *Journal of Communication*, Volume 64, Issue 2, April 2014, Pages 215–238, <https://doi.org/10.1111/jcom.12082>

Action Area 3: Valuing Contributors (another worked example)

- Underproduction in Wikipedia
- Quality measure: article structural completeness.

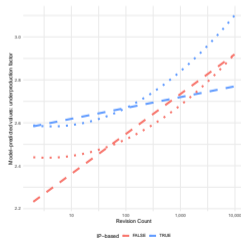


K Champion, BM Hill (2024) Countering underproduction of peer produced goods. *New Media & Society*

Aaron Shaw, Benjamin M. Hill, Laboratories of Oligarchy? How the Iron Law Extends to Peer Production, *Journal of Communication*, Volume 64, Issue 2, April 2014, Pages 215–238, <https://doi.org/10.1111/jcom.12082>

Action Area 3: Valuing Contributors (another worked example)

- Underproduction in Wikipedia
- Quality measure: article structural completeness.
- Importance measure: readership.

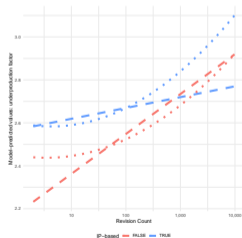


K Champion, BM Hill (2024) Countering underproduction of peer produced goods. *New Media & Society*

Aaron Shaw, Benjamin M. Hill, Laboratories of Oligarchy? How the Iron Law Extends to Peer Production, *Journal of Communication*, Volume 64, Issue 2, April 2014, Pages 215–238, <https://doi.org/10.1111/jcom.12082>

Action Area 3: Valuing Contributors (another worked example)

- Underproduction in Wikipedia
- Quality measure: article structural completeness.
- Importance measure: readership.
- As contributors persist, task selection tends toward underproduced articles: those low quality, important goods.

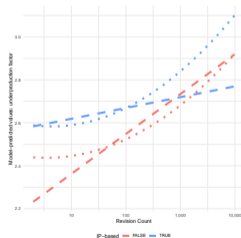


K Champion, BM Hill (2024) Countering underproduction of peer produced goods. *New Media & Society*

Aaron Shaw, Benjamin M. Hill, Laboratories of Oligarchy? How the Iron Law Extends to Peer Production, *Journal of Communication*, Volume 64, Issue 2, April 2014, Pages 215–238, <https://doi.org/10.1111/jcom.12082>

Action Area 3: Valuing Contributors (another worked example)

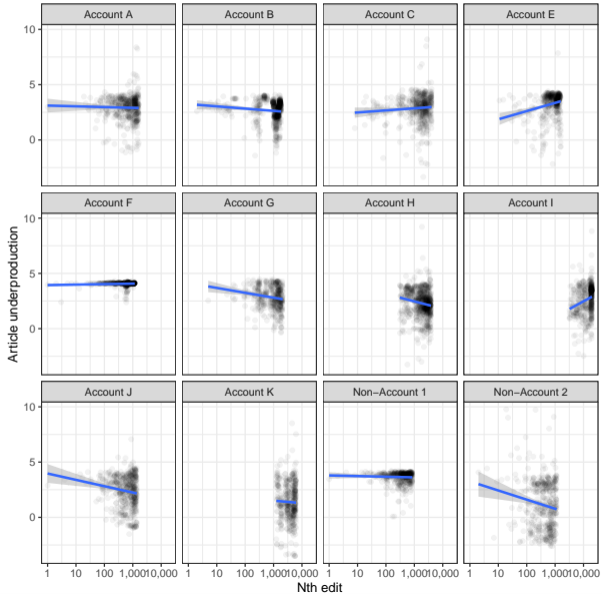
- Underproduction in Wikipedia
- Quality measure: article structural completeness.
- Importance measure: readership.
- As contributors persist, task selection tends toward underproduced articles: those low quality, important goods.
- Caveat: Retention helps counter underproduction....but there's a risk of oligarchy forming.



K Champion, BM Hill (2024) Countering underproduction of peer produced goods. *New Media & Society*

Aaron Shaw, Benjamin M. Hill, Laboratories of Oligarchy? How the Iron Law Extends to Peer Production, *Journal of Communication*, Volume 64, Issue 2, April 2014, Pages 215–238, <https://doi.org/10.1111/jcom.12082>

Action Area 3: Valuing contributors (another worked example)



The story so far...

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- What are our strategies to respond?
 - Four areas of challenging interventions: newcomers, funding, governance, organizing work
 - Three areas of immediate action: data, toxicity, and valuing one another

What can research tell us about building, leading, and sustaining open source?

- Getting oriented
 - The world runs on free software.
 - Extensive research record on FLOSS and FLOSS communities.
- What challenges do we face?
 - Free software requires a lot of effort to build and sustain.
 - Our efforts can be misaligned: important communities can fall apart, and highly important software can be low quality.
- What are our strategies to respond?
 - Four areas of challenging interventions: newcomers, funding, governance, organizing work
 - Three areas of immediate action: data, toxicity, and valuing one another
- **In Summary**

In Summary!

- Building and maintaining free software is a huge endeavor.

In Summary!

- Building and maintaining free software is a huge endeavor.
- Critical components can be neglected and communities can be hard to sustain.

In Summary!

- Building and maintaining free software is a huge endeavor.
- Critical components can be neglected and communities can be hard to sustain.
- We need to actively align our efforts with public needs and do so sustainably.

In Summary!

- Building and maintaining free software is a huge endeavor.
- Critical components can be neglected and communities can be hard to sustain.
- We need to actively align our efforts with public needs and do so sustainably.
- Formality, extensive reports, and incautiously seeking funds and newcomers might not help us.

In Summary!

- Building and maintaining free software is a huge endeavor.
- Critical components can be neglected and communities can be hard to sustain.
- We need to actively align our efforts with public needs and do so sustainably.
- Formality, extensive reports, and incautiously seeking funds and newcomers might not help us.
- Gather great data, share power, minimize toxicity, and encourage one another.

Thank you!

`kaylea@uw.edu—@kaylea@social.coop`

`https://kayleachampion.com`

`https://communitydata.science—@communitydata@social.coop`

Actively seeking new funding and collaborators!

We gratefully acknowledge support from the Sloan Foundation through the Ford/Sloan Digital Infrastructure Initiative, Sloan Award 2018-11356 as well as the National Science Foundation (Grant IIS-2045055). This work was conducted using the Hyak supercomputer at the University of Washington as well as research computing resources at Northwestern University.



[Washington Coast. John Murphy. (kingair42) via flickr, CC BY-SA 2.0]