

On average, the same dev will write cleaner code
when they commit to a clean repo,
and messier code when they commit to a messy one.

Spinellis, D., Louridas, P., Kechagia, M; and Sharma, T. (2024) Broken windows: Exploring the applicability of a controversial theory on code quality. *ICSME 2024* Preprint: <https://arxiv.org/abs/2410.13480>

According to my research...

The latest insights on building, leading, and sustaining open source

Dr. Kaylea Champion – kaylea@uw.edu

<https://www.kayleachampion.com>

<https://communitydata.science>

University of Washington

PuPPy—December 18, 2024





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

How do researchers use open source?

- Subject and Object

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)
 - as a setting and data source

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)
 - as a setting and data source
 - as important in its own right

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)
 - as a setting and data source
 - as important in its own right
 - as part of larger social and technical phenomena

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)
 - as a setting and data source
 - as important in its own right
 - as part of larger social and technical phenomena
- Research owes a lot to open source!

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)
 - as a setting and data source
 - as important in its own right
 - as part of larger social and technical phenomena
- Research owes a lot to open source!
- Putting this research into practice can be a challenge.

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

How do researchers use open source?

- Subject and Object
 - as a tool (like everyone else)
 - as a setting and data source
 - as important in its own right
 - as part of larger social and technical phenomena
- Research owes a lot to open source!
- Putting this research into practice can be a challenge.
- Thank you for being here to help bridge this gap!

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.

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?**

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?**
- What are our strategies to respond?
- Takeaways

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.
 -entitled and hostile users



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.
 -entitled and hostile users
 -entitled and hostile contributors



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.
 -entitled and hostile users
 -entitled and hostile contributors
 -unhelpful problem reports



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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.
 -entitled and hostile users
 -entitled and hostile contributors
 -unhelpful problem reports
- 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>

Challenge 1: Free Software is a lot of work!

- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.
 -entitled and hostile users
 -entitled and hostile contributors
 -unhelpful problem reports
- 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>

Challenge 1: Free Software is a lot of work!

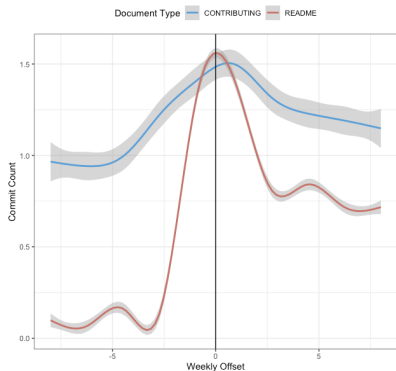
- Code is fun...until it's not.
 - ...abandoned dependencies
 - ...technical debt
 - ...guilt
- Having people use your stuff is fun....until it's not.
 -entitled and hostile users
 -entitled and hostile contributors
 -unhelpful problem reports
- Events build trust, improve teamwork, and make the experience more fun
- Beware catastrophic success (automate, automate, automate)
- Community might not be the 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>

Many ways to do open source – possible advice misalignment



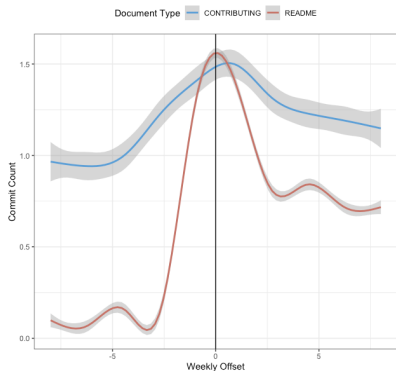
- OSS guides and how-to docs emphasize governance, community, newcomers, and process.

Is this accidental infrastructure? Should community always be the goal?

Note: this work is currently under review.

Matthew Gaughan, Kaylea Champion, Sohyeon Hwang, and Aaron Shaw. (2025.) The introduction of README and CONTRIBUTING files in open source software development.

Many ways to do open source – possible advice misalignment



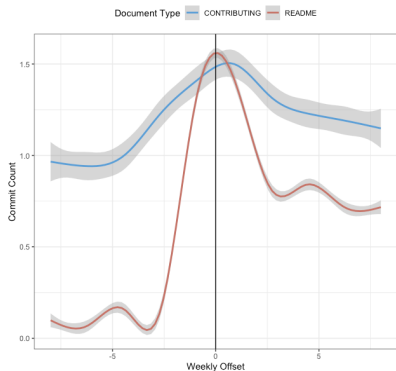
- OSS guides and how-to docs emphasize governance, community, newcomers, and process.
- These guides suggest putting this info in README.md and CONTRIBUTING.md files.

Is this accidental infrastructure? Should community always be the goal?

Note: this work is currently under review.

Matthew Gaughan, Kaylea Champion, Sohyeon Hwang, and Aaron Shaw. (2025.) The introduction of README and CONTRIBUTING files in open source software development.

Many ways to do open source – possible advice misalignment



- OSS guides and how-to docs emphasize governance, community, newcomers, and process.
- These guides suggest putting this info in README.md and CONTRIBUTING.md files.
- Many projects that go on to be core infrastructure do not emphasize these topics in their first README and CONTRIBUTING docs!

Is this accidental infrastructure? Should community always be the goal?

Note: this work is currently under review.

Matthew Gaughan, Kaylea Champion, Sohyeon Hwang, and Aaron Shaw. (2025.) The introduction of README and CONTRIBUTING files in open source software development.

Challenge 2: Some critical infrastructure is neglected

- Leftpad and CrowdStrike remind us how interdependent we are.

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>

Challenge 2: Some critical infrastructure is neglected

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

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>

Challenge 2: 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.

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>

Challenge 2: 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 and with greater numbers of contributors!

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>

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.

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?**

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?**
- Takeaways

Response Strategies

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>

Response Strategies

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>

Response Strategies

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>

Response Strategies

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>

Response Strategies

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>

Response Strategies

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>

Response Strategies

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>

Awareness: 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>

Awareness: 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>

Awareness: 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>

Awareness: 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>

Awareness: 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>

Awareness: 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>

Awareness: 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) (**errors per statement**)

Example: Misalignment Analysis In Action

A worked example of this process in action!

- Measure importance (**installations**)
- Measure quality (or security) (**errors per statement**)
- 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) (**errors per statement**)
- Specify a relationship (**alignment of importance and quality**)
- Measure deviations (**gaps and slopes**)



debian /



The universal operating system

THE COMMUNITY

Debian is a Community of People!



People

Who we are and what we do

THE OPERATING SYSTEM

Debian is a complete Free Operating System!



Download

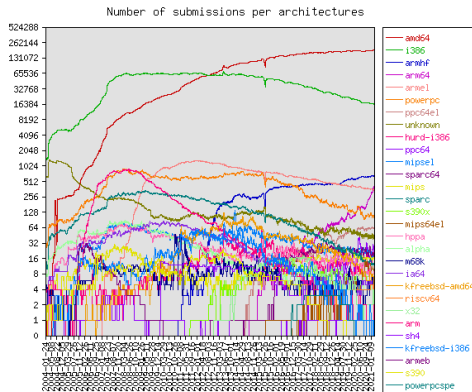
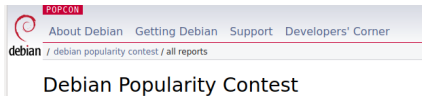


Why Debian

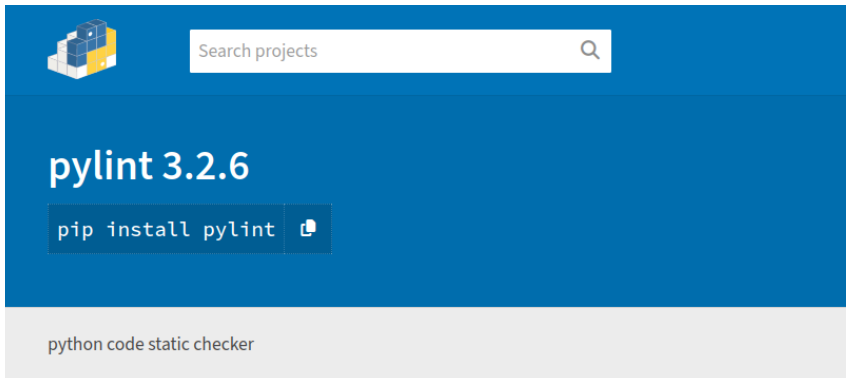
What makes Debian special

- 1 - Measure importance
- 2 - Measure quality / security
- 3 - Specify a relationship
- 4 - Measure deviations



Example: Measuring Importance




Example: Measuring Quality



The image shows the header section of the PyLint project page on PyPI. It features a blue background with a search bar at the top. The search bar contains the text "Search projects" and a magnifying glass icon. Below the search bar, the text "pylint 3.2.6" is displayed in a large, white, sans-serif font. Underneath the version number, there is a dark blue button with the text "pip install pylint" in white, followed by a small white icon of a document with a plus sign. At the bottom of the header, the text "python code static checker" is written in a smaller, white, sans-serif font.

 Search projects 

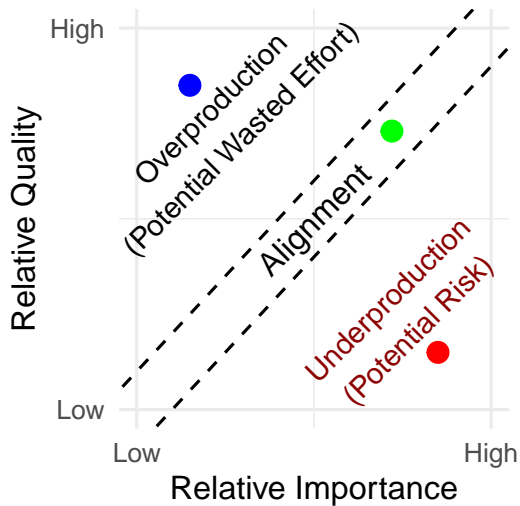
pylint 3.2.6

`pip install pylint` 

python code static checker

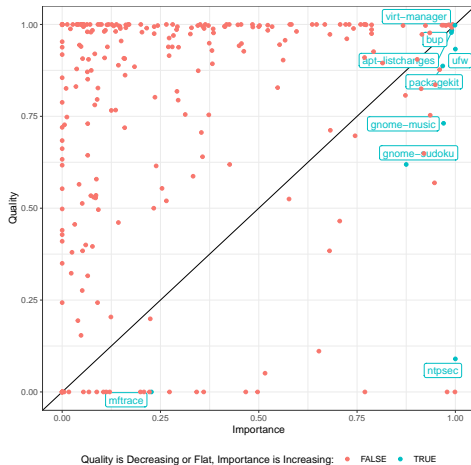
- 1 - Measure importance
- 2 - Measure quality / security
- 3 - Specify a relationship
- 4 - Measure deviations

Example: Specifying a relationship



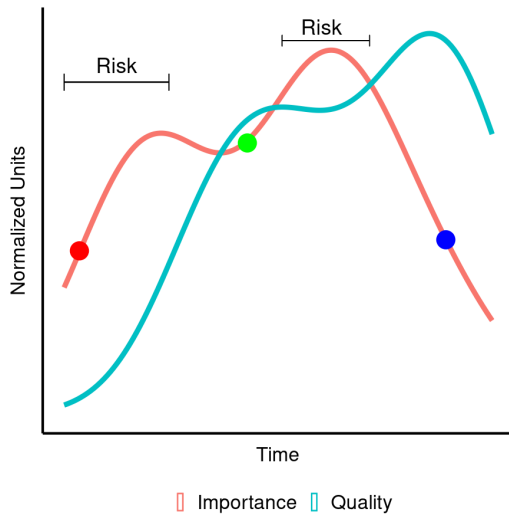
- 1 - Measure importance
- 2 - Measure quality / security
- 3 - Specify a relationship
- 4 - Measure deviations

Example: Measuring Deviations

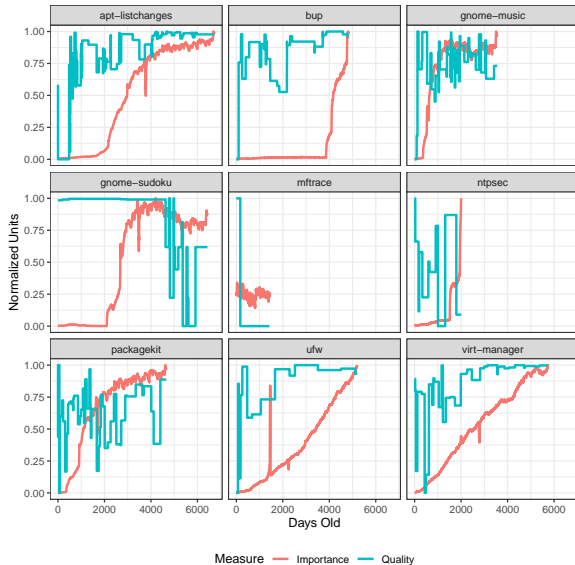


n = 201; 20% of these (42) are underproduced today.

Example: Measuring Deviations



Example: Measuring Deviations



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

W. Xiao, H. He, W. Xu, Y. Zhang, and M. Zhou, "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. <https://dl.acm.org/doi/10.1145/3611643.3616349>

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

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

W. Xiao, H. He, W. Xu, Y. Zhang, and M. Zhou, "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. <https://dl.acm.org/doi/10.1145/3611643.3616349>

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

W. Xiao, H. He, W. Xu, Y. Zhang, and M. Zhou, "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. <https://dl.acm.org/doi/10.1145/3611643.3616349>

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

W. Xiao, H. He, W. Xu, Y. Zhang, and M. Zhou, "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. <https://dl.acm.org/doi/10.1145/3611643.3616349>

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

W. Xiao, H. He, W. Xu, Y. Zhang, and M. Zhou, "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*. <https://dl.acm.org/doi/10.1145/3611643.3616349>

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

W. Xiao, H. He, W. Xu, Y. Zhang, and M. Zhou, "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. <https://dl.acm.org/doi/10.1145/3611643.3616349>

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). <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). <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) <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). <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). <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) <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). <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). <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) <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, hosting, CI/CD 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). <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). <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) <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, hosting, CI/CD is a common approach here).
- New models emerging: GitHub Grants, NSF POSE, Alpha-Omega, Sloan Digital Infrastructure Initiative, Germany's Sovereign Tech Fund

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

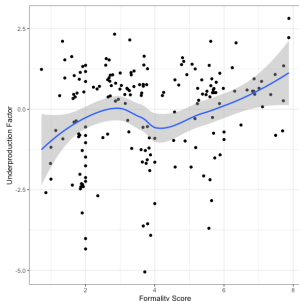
Chełkowski T, Jemielniak D, Macikowski K (2021) Free and Open Source Software organizations: A large-scale analysis of code, comments, and commits frequency. PLoS ONE 16(9): e0257192. <https://doi.org/10.1371/journal.pone.0257192>

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). <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) <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.



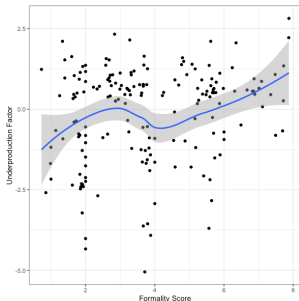
Gaughan, M.; Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*. Chakraborti, M.; Atkisson, C.; Stănculescu, S.; Filkov, V. and Frey, S. 2024. Do We Run How We Say We

Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24) <https://doi.org/10.1145/3613904.3641980>
Gaughan, M.; Champion, K.; Hwang, S.; and Shaw, A. (2025). The Introduction of README and CONTRIBUTING

Files in Open Source Software Development. *Under Review*.

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.



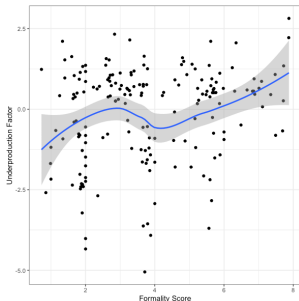
Gaughan, M.; Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*. Chakraborti, M.; Atkisson, C.; Stănculescu, S.; Filkov, V. and Frey, S. 2024. Do We Run How We Say We

Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24) <https://doi.org/10.1145/3613904.3641980>
Gaughan, M.; Champion, K.; Hwang, S.; and Shaw, A. (2025). The Introduction of README and CONTRIBUTING

Files in Open Source Software Development. *Under Review*.

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.



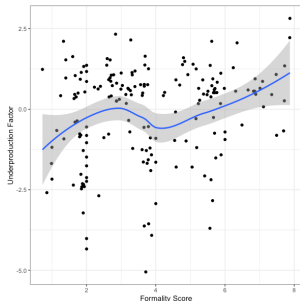
Gaughan, M.; Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*. Chakraborti, M.; Atkisson, C.; Stănculescu, S.; Filkov, V. and Frey, S. 2024. Do We Run How We Say We

Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24) <https://doi.org/10.1145/3613904.3641980>
Gaughan, M.; Champion, K.; Hwang, S.; and Shaw, A. (2025). The Introduction of README and CONTRIBUTING

Files in Open Source Software Development. *Under Review*.

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.



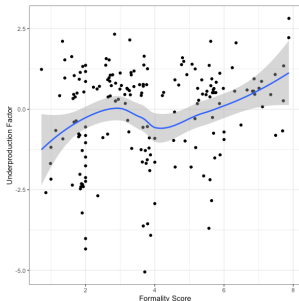
Gaughan, M.; Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*. Chakraborti, M.; Atkisson, C.; Stănculescu, S.; Filkov, V. and Frey, S. 2024. Do We Run How We Say We

Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24) <https://doi.org/10.1145/3613904.3641980>
Gaughan, M.; Champion, K.; Hwang, S.; and Shaw, A. (2025). The Introduction of README and CONTRIBUTING

Files in Open Source Software Development. *Under Review*.

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.



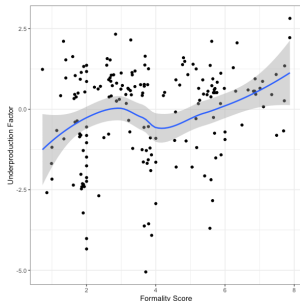
Gaughan, M.; Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*. Chakraborti, M.; Atkisson, C.; Stănculescu, S.; Filkov, V. and Frey, S. 2024. Do We Run How We Say We

Run? Formalization and Practice of Governance in OSS Communities. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24) <https://doi.org/10.1145/3613904.3641980>
Gaughan, M.; Champion, K.; Hwang, S.; and Shaw, A. (2025). The Introduction of README and CONTRIBUTING

Files in Open Source Software Development. *Under Review*.

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.
- Prevailing advice emphasizes a community model—but many projects don't start that way.



Gaughan, M.; Champion, K.; and Hwang, S. (2024) Engineering Formality and Software Risk in Debian. *IEEE SANER*. Chakraborti, M.; Atkisson, C.; Stănculescu, S.; Filkov, V. and Frey, S. 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) <https://doi.org/10.1145/3613904.3641980>
Gaughan, M.; Champion, K.; Hwang, S.; and Shaw, A. (2025). The Introduction of README and CONTRIBUTING

Files in Open Source Software Development. *Under Review*.

Intervention Challenge 4: Organizing our work

Coordination can be costly!

- Automation is helpful but can run up bills (CI/CD waste – look for dependency bloat)

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 – look for dependency bloat)
- Document decisions and acceptance criteria

Irina Zaks, Timothy Lehen, 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 – look for dependency bloat)
- Document decisions and acceptance criteria
- “Review then commit” slows pace. Just make the pull request!

Irina Zaks, Timothy Lehen, 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?
 - Five elements make intervention hard: newcomers, funding, governance, organizing work,
hard to find good advice

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?
 - Five elements make intervention hard: newcomers, funding, governance, organizing work, hard to find good advice
 - **Three areas of immediate action**

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?
 - Five elements make intervention hard: newcomers, funding, governance, organizing work, hard to find good advice
 - **Three areas of immediate action**
- In Summary

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 is hard without metadata and changelogs.

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

- Keeping logs and records can help a lot.
 - Historical interpretation is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'

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

- Keeping logs and records can help a lot.
 - Historical interpretation is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'
- 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 is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'
- Engage with metrics communities and researchers.
 - Reach out to the Community Data Science Collective (or me!).

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

- Keeping logs and records can help a lot.
 - Historical interpretation is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'
- Engage with metrics communities and researchers.
 - Reach out to the Community Data Science Collective (or me!).
 - 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 is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'
- Engage with metrics communities and researchers.
 - Reach out to the Community Data Science Collective (or me!).
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.
- Open bi-directional communication channels between users and development communities!
 - quality

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

- Keeping logs and records can help a lot.
 - Historical interpretation is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'
- Engage with metrics communities and researchers.
 - Reach out to the Community Data Science Collective (or me!).
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.
- Open bi-directional communication channels between users and development communities!
 - quality
 - security

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

- Keeping logs and records can help a lot.
 - Historical interpretation is hard without metadata and changelogs.
 - 'Oh, right there data is missing/spiked because we changed out the infrastructure....'
- Engage with metrics communities and researchers.
 - Reach out to the Community Data Science Collective (or me!).
 - Consider scarf.sh and the CHAOSS project for lots of metrics-related resources.
- Open bi-directional communication channels between users and development communities!
 - 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. <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). <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) <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. <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). <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) <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. <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)*. <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) <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. <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). <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) <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).

<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, doi: 10.1109/ICSE43902.2021.00093.

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

- Focused code contributors build sustainability.
- Wide-roaming 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).

<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, doi: 10.1109/ICSE43902.2021.00093.

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

- Focused code contributors build sustainability.
- Wide-roaming non-code contributors build sustainability.
- Good maintainers have technical chops...

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).

<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, doi: 10.1109/ICSE43902.2021.00093.

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

- Focused code contributors build sustainability.
- Wide-roaming non-code contributors build sustainability.
- Good maintainers have technical chops...
...great maintainers have technical chops and **communicate**.

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).

<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, doi: 10.1109/ICSE43902.2021.00093.

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

- Focused code contributors build sustainability.
- Wide-roaming non-code contributors build sustainability.
- Good maintainers have technical chops...
...great maintainers have technical chops and **communicate**.
- Retaining people in the community helps fight misalignment.

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).

<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, doi: 10.1109/ICSE43902.2021.00093.

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

- Focused code contributors build sustainability.
- Wide-roaming non-code contributors build sustainability.
- Good maintainers have technical chops...
...great maintainers have technical chops and **communicate**.
- Retaining people in the community helps fight misalignment.
...but there's a risk of oligarchy forming.

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).

<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, doi: 10.1109/ICSE43902.2021.00093.

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>

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**

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.
- Aligning development efforts with public needs deserves focus.

In Summary!

- Building and maintaining free software is a huge endeavor.
- Critical components can be neglected and communities can be hard to sustain.
- Aligning development efforts with public needs deserves focus.
- Formality, extensive reports, and incautiously seeking resources (funds, 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.
- Aligning development efforts with public needs deserves focus.
- Formality, extensive reports, and incautiously seeking resources (funds, newcomers) might not help us.
- Gather great data, share power, minimize toxicity, and encourage one another.



Some questions I want to work on

- How can we engage in graceful sunsetting – walking away, shutting down, facing our own end?

Some questions I want to work on

- How can we engage in graceful sunsetting – walking away, shutting down, facing our own end?
- How might we apply the underproduction analysis approach to a broader range of settings and measures, and what interventions work to counter underproduction?

Some questions I want to work on

- How can we engage in graceful sunseting – walking away, shutting down, facing our own end?
- How might we apply the underproduction analysis approach to a broader range of settings and measures, and what interventions work to counter underproduction?
- What are the impacts of big shifts in public attention on open source ecosystems?

Some questions I want to work on

- How can we engage in graceful sunsetting – walking away, shutting down, facing our own end?
- How might we apply the underproduction analysis approach to a broader range of settings and measures, and what interventions work to counter underproduction?
- What are the impacts of big shifts in public attention on open source ecosystems?
- How can we hold space for casual, no-commitment open source work?

Some questions I want to work on

- How can we engage in graceful sunsetting – walking away, shutting down, facing our own end?
- How might we apply the underproduction analysis approach to a broader range of settings and measures, and what interventions work to counter underproduction?
- What are the impacts of big shifts in public attention on open source ecosystems?
- How can we hold space for casual, no-commitment open source work?
- “Public data” doesn’t mean “anything goes”

Some questions I want to work on

- How can we engage in graceful sunsetting – walking away, shutting down, facing our own end?
- How might we apply the underproduction analysis approach to a broader range of settings and measures, and what interventions work to counter underproduction?
- What are the impacts of big shifts in public attention on open source ecosystems?
- How can we hold space for casual, no-commitment open source work?
- “Public data” doesn’t mean “anything goes”
- “Model collapse” – what about social model collapse?

Thank you!

`kaylea@uw.edu—@kaylea@social.coop`

`https://kayleachampion.com`

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

Actively seeking new funding and collaborators!

- 1 - Measure importance
- 2 - Measure quality / security
- 3 - Specify a relationship
- 4 - Measure deviations

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.

