Demystifying Python Packaging

Leah Wasser, Executive Director & Founder

pyOpenSci builds diverse community around scientific open source software

peer review, training & mentorship

2018 — Founded

2019 — Started peer review

2021 — Sloan Foundation Funding

Fall 2022 — Launched as independent organization

Leadership

Executive council

Tracy Teal
Board chair

Karen Cranston

DEIA & education council

Ariane Sasso

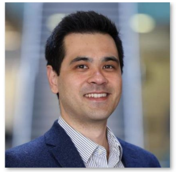
Agustina Pesce

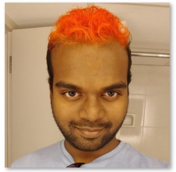
Advisory council

Chris Holdgraf
UC Berkeley

Inessa Pawson
Numpy, OpenTeams Incubator

Pradyun Gedam
PyPA, PSF, Bloomberg Python Infrastructure


Leonardo Uieda
University of Liverpool


Yuvi Panda

Filipe Fernandes

Lindsey Heagy
@simpeg, @geoscixyz, @ubcgif, @2i2c_org

Martin Fleischmann

Ivan Ogasawara
xmnlab

https://www.pyopensci.org/our-community/

Outline

- Packaging confusion
- Packaging Guide: pyOpenSci's process
- Software peer review + community -> encourage standards

What would [Geopandas] do?

Many model package structure following packages built by trusted maintainers

My experience with versioning

setuptools_scm

- Talked to lots of people
- Read (*dated*) blogs
- Lots of experimentation.

Python Packaging

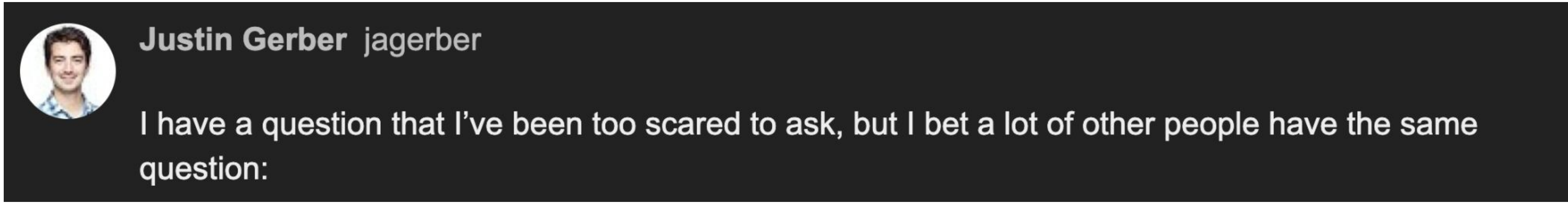
1. Documentation
2. Community disagreement
3. Many (*competing?*) tool options

Challenge 1: Documentation

Packaging challenges

- Dated
- Assumes technical knowledge
    - Asks users to make decisions about things they aren't expert in.

pyOpenSci



Justin Gerber jagerber

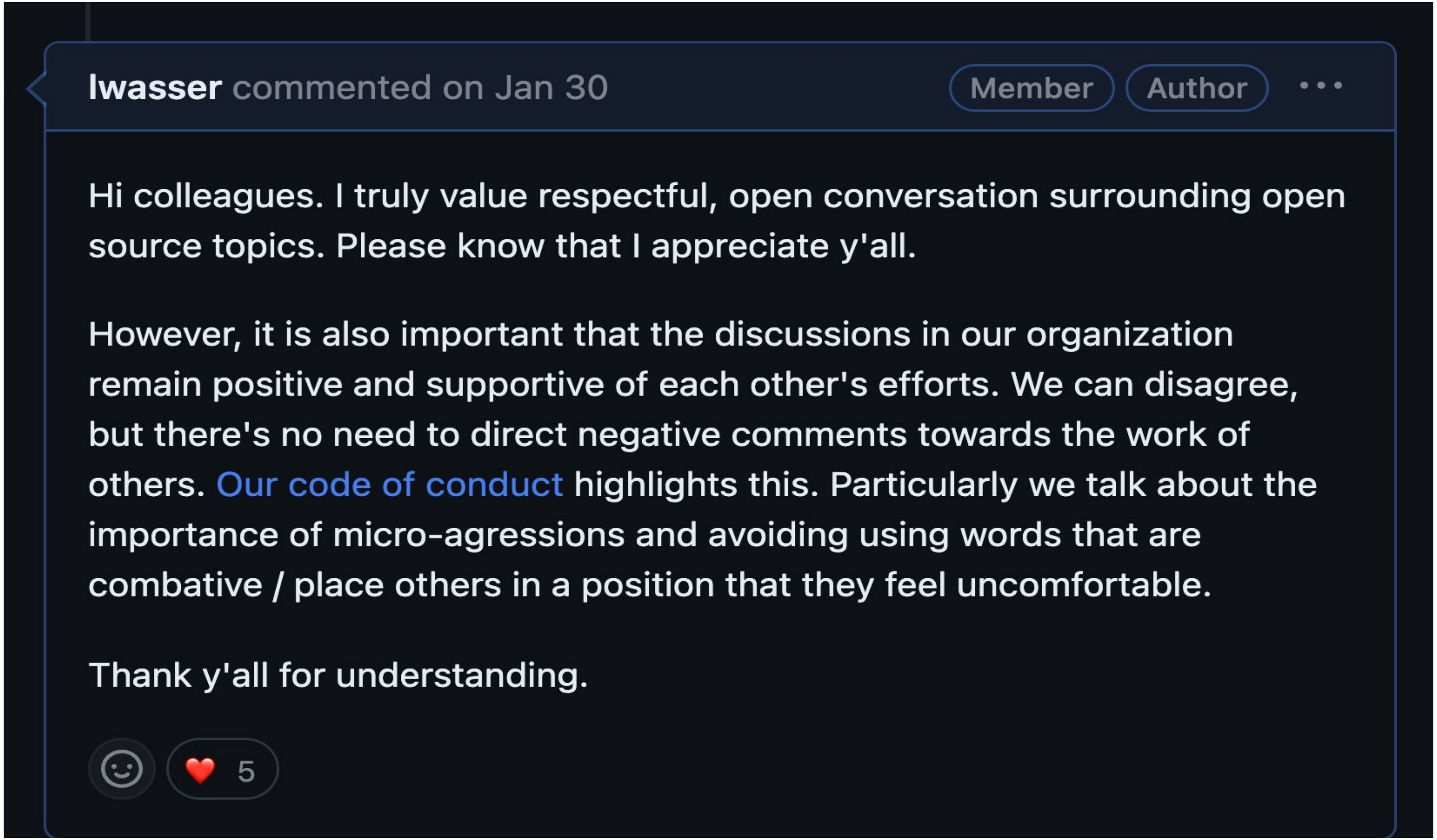
I have a question that I’ve been too scared to ask, but I bet a lot of other people have the same question:

“What is ‘building’ and why do I care about it?”

“The reason I don’t want to select a build tool is because I don’t want to have to learn that much about what “building” means. “

Challenge 2: community disagreement

unwelcoming



lwasser commented on Jan 30 Member Author

Hi colleagues. I truly value respectful, open conversation surrounding open source topics. Please know that I appreciate y'all.

However, it is also important that the discussions in our organization remain positive and supportive of each other's efforts. We can disagree, but there's no need to direct negative comments towards the work of others. Our code of conduct highlights this. Particularly we talk about the importance of micro-agressions and avoiding using words that are combative / place others in a position that they feel uncomfortable.

Thank y'all for understanding.

❤️ 5

Problem 3: Too many tools

- Flit
- PDM
- Hatch
- Hatchling
- Front-ends
- Back-ends
- …

"The reason i don't want to select a build tool is because i don't want to have to learn that much about what "building" means. "

Most people want to know how to create a package

What if???

We prioritized a shared vision: support beginners creating pure Python projects.

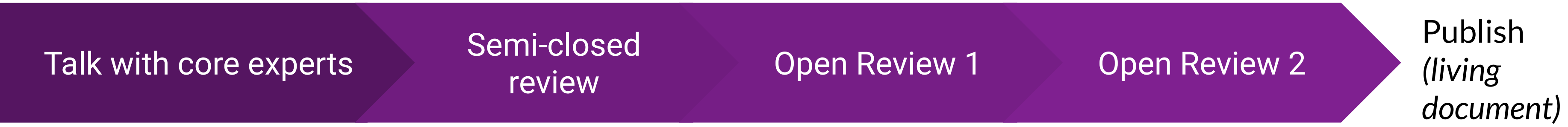
Community Driven Python Packaging Guide

- **Vision & Scope**: content for beginners
- Significant community input
  - Tool maintainers about tool capabilities & development trajectories
- Heavily moderated discussions

pyOpenSci

Python Packaging Guide

Our process



- Write a section of the guide

* Core experts review

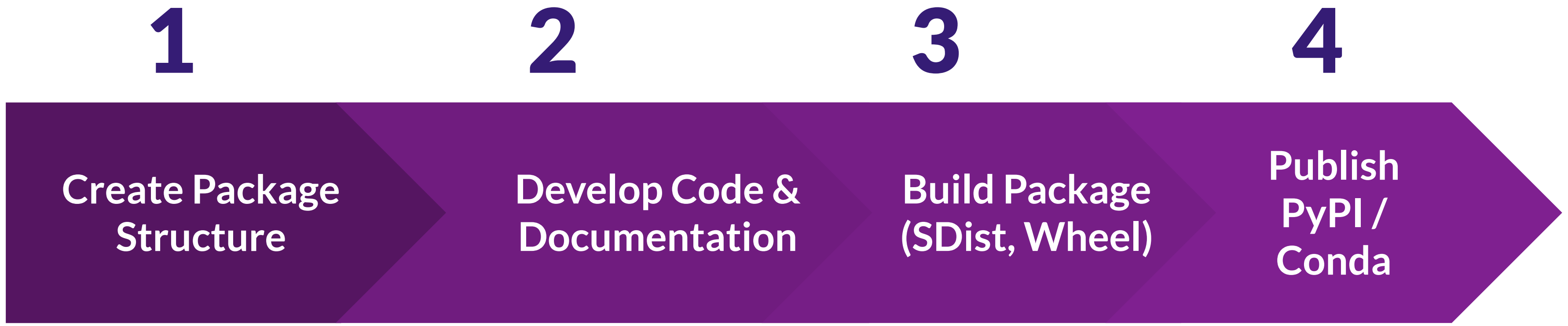
- Ping tool developers and maintainers
- Welcome broad community feedback

- Welcome broad community feedback

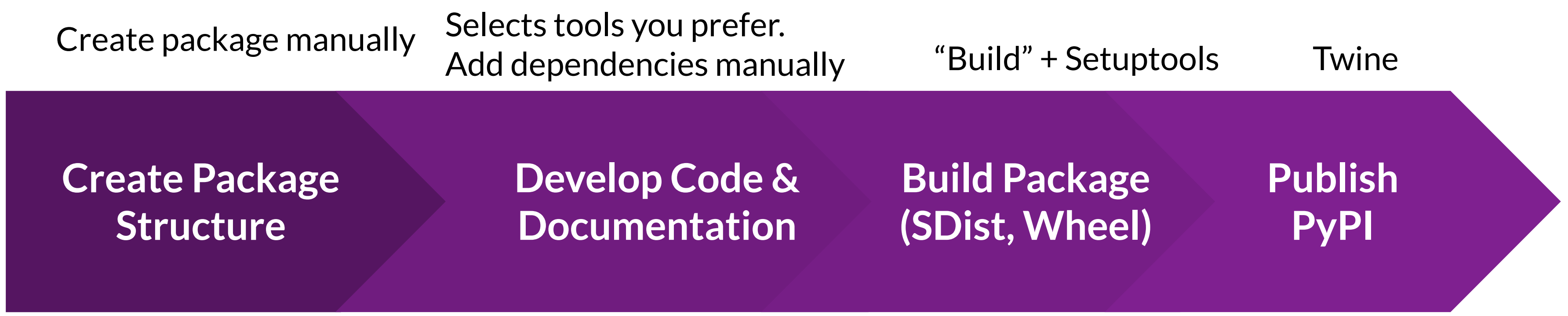
www.pyopensci.org/python-package-guide/

How do you pick a packaging tool?

Pure Python Package



Pure Python Package
Hammer & Wrench Approach


Create package manually
Selects tools you prefer.
Add dependencies manually
“Build” + Setuptools
Twine
Create Package Structure
Develop Code & Documentation
Build Package (SDist, Wheel)
Publish PyPI

Pure Python Package Multi-tool Approach

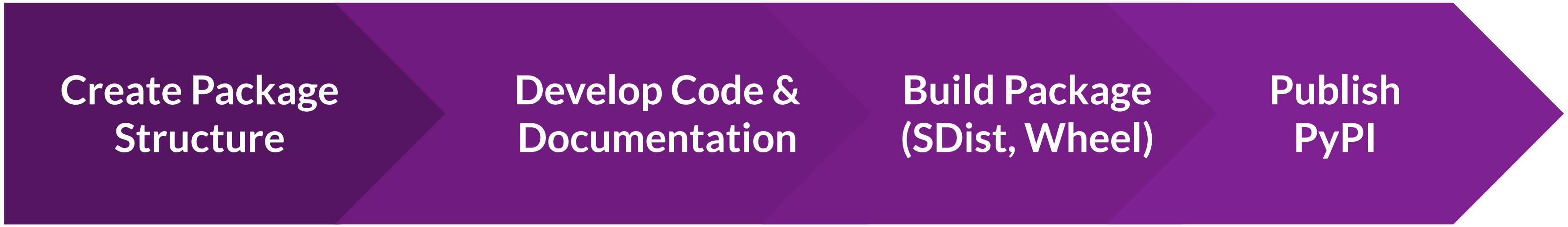
PDM, Hatch, Poetry

PDM Init

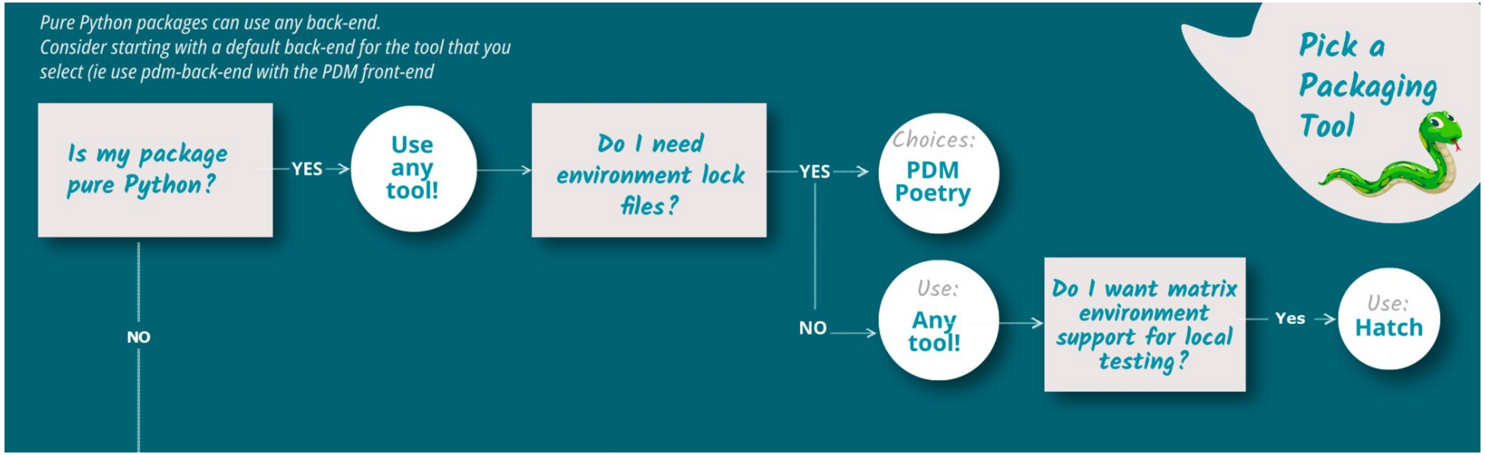
PDM
(environment support, dependency versions)

PDM build

PDM publish



So many tools!



Pure Python packages can use any back-end.
Consider starting with a default back-end for the tool that you select (ie use pdm-back-end with the PDM front-end

Pick a Packaging Tool
Is my package pure Python?
YES
NO
Use any tool!
Do I need environment lock files?
YES
Choices: PDM Poetry
NO
Use: Any tool!
Do I want matrix environment support for local testing?
Yes
Use: Hatch

Software Peer Review

pyOpenSci

Open peer review

- Provide credit (citable entity)
- Packaging standardization
- Guidance on packaging best practices

pyOpenSci Peer Review Process

maintainer(s)
Submit PreSubmission Inquiry on GitHub

maintainer(s)
Submit Tool For Review
GitHub

Conversation

(2 reviewers)
Package Review

pyOS standards

🎉 Package Accepted 🎉

Optional

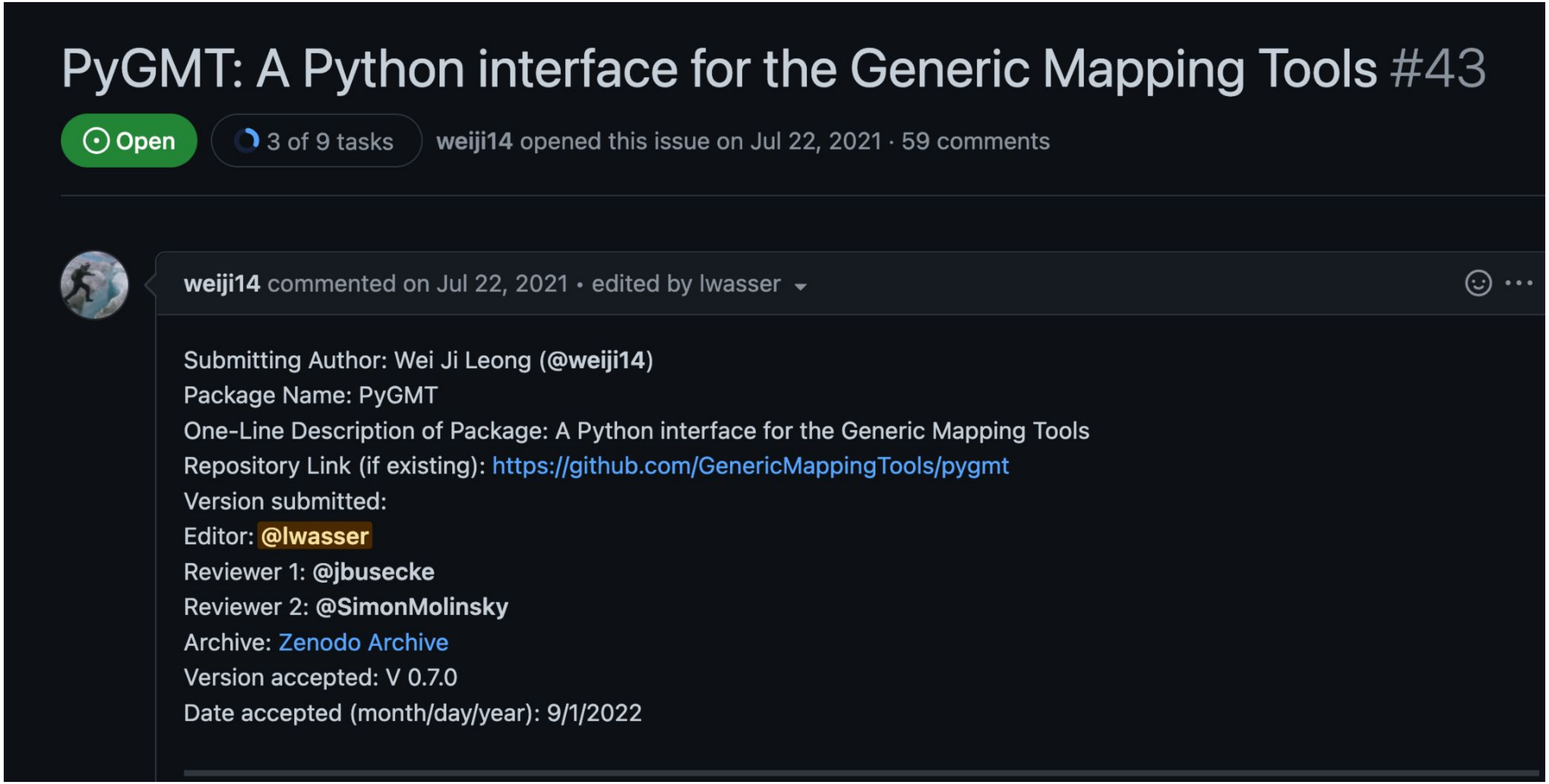
Fast Track in JOSS
(if in scope)

JOSS
The Journal of Open Source Software

✨Package is a part of the pyOS ecosystem✨

- Maintenance check-ins
- Community visibility
- Maintainer still “owns” tool

Review mentorship available!



PyGMT: A Python interface for the Generic Mapping Tools #43

Open · 3 of 9 tasks · **weiji14** opened this issue on Jul 22, 2021 · 59 comments

weiji14 commented on Jul 22, 2021 • edited by lwasser

Submitting Author: Wei Ji Leong (**@weiji14**)
Package Name: PyGMT
One-Line Description of Package: A Python interface for the Generic Mapping Tools
Repository Link (if existing): https://github.com/GenericMappingTools/pygmt
Version submitted:
Editor: **@lwasser**
Reviewer 1: **@jbusecke**
Reviewer 2: **@SimonMolinsky**
Archive: Zenodo Archive
Version accepted: V 0.7.0
Date accepted (month/day/year): 9/1/2022

pynteny

Presubmission

github.com/pyOpenSci/software-review/issues/65

- Move metadata setup.py → pyproject.toml
- Moved docs from wiki → sphinx
- Helped with install issues

pyOpenSci

pyOpenSci Editorial Team

Ariane Sasso
Editor
Hasso Plattner Institute

David Nicholson
Editor in Chief

Chiara Marmo
Editor

Anita Graser
Editor
AIT Austrian Institute of Technology

Alex Batisse
Editor

Jenny Palomino
Editor
Google Cloud Learning Services

pyOpenSci

Community Partnerships

JOSS
The Journal of Open Source Software

Open Science Labs

ROpenSci

Scientific Python

PANGEO

Thank you!

pyOpenSci

ALFRED P. SLOAN FOUNDATION

Community Initiatives
In Service to GREAT IDEAS

leah@pyopensci.org

@pyopensci@fosstodon.org

pyopensci

Ways to get involved:

- [Sign up for discourse to get latest updates!]
- [Sign up to be a reviewer]
- [Submit a package for review]
- Review our peer review guide
- Review our packaging guide
- Post a packaging question on discourse
- Spread the word!