The challenges of maintaining a popular open-source project

EuroPython 2018 — July 27, 2018
Raphael Pierzina
Thanks to the organizers and volunteers for a great event! 🏴‍☠️
$ whoami
• Maintainer and core developer of pytest and Cookiecutter and related projects

• I also write, speak, and tweet about these and other open source projects

• I currently live in Berlin, Germany and work as a Senior Test Engineer for Firefox at Mozilla
https://raphael.codes/
@hackebrot
Agenda

1. Maintaining a popular open-source project
2. Challenges of a growing community
3. Frustrations for maintainers
4. Take-aways for maintainers
Maintaining a popular open-source project
Cookiecutter

- Command-line utility that creates projects from templates
- Project templates can be in any programming language or markup format
- Cross-platform: Windows, Mac, and Linux are officially supported
- Works with CPython 2.7, 3.4, 3.5, 3.6 and PyPy
https://github.com/audreyr/cookiecutter
Community

• Free and open-source software: permissive BSD-3 license
• 180 individual contributors from around the world
• More than 1000 public templates on GitHub
• Multiple talks at community conferences
Templates

- https://github.com/audreyr/cookiecutter-pypackage
- https://github.com/pydanny/cookiecutter-django
- https://github.com/pytest-dev/cookiecutter-pytest-plugin
- ...

@hackebrot
Contributing

• Development of Cookiecutter is community-driven

• Connect with other Cookiecutter contributors and users on Gitter

• Everyone is invited to contribute! 😞🍪
$ pip install cookiecutter
$ cookiecutter gh:pytest-dev/cookiecutter-pytest-plugin
full_name [Raphael Pierzina]:
email [raphael@hackebrot.de]: raphael@mozilla.com
github_username [hackebrot]:
plugin_name [foobar]: emoji
module_name [emoji]:
short_description [A simple plugin to use with pytest]: Add emojis to the test report log
version [0.1.0]:
pytest_version [3.5.0]: 3.6.3
Select docs_tool:
1 - mkdocs
2 - sphinx
3 - none
Choose from 1, 2, 3 [1]:
Select license:
1 - MIT
2 - BSD-3
3 - GNU GPL v3.0
4 - Apache Software License 2.0
5 - Mozilla Public License 2.0
Choose from 1, 2, 3, 4, 5 [1]: 5
My contributions

• Created and published several Cookiecutter templates

• Converted the test suite from unittest to pytest

• Helped users with problems on the issue tracker and chat

• Reviewed pull requests from other contributors
My contributions

• Developed a number of features and bug fixes

• Created poyo to overcome installation issues of YAML parsers

• Created jinja2-time to make it easier to retrieve the current time

• Created pytest-cookies to make testing templates easier
Cookiecutter

@hackebrot
pytest-cookies

hackebrot
125 commits 1,699 ++ 569 --

#1

Twitter: @hackebrot
poyo
Becoming a maintainer

• Was granted the commit bit by the core team

• Started managing releases on GitHub and pushing to PyPI

• Continued to develop bug fixes and new features
Being a maintainer

• Review pull requests and triage issues
• Speak about Cookiecutter at EuroPython, PyData and local user groups
• Explored various ways of getting funding for Cookiecutter
• ...

@hackebrot
Emails 📪
Challenges of a growing community
Community

• Cookiecutter attracts users at a faster rate than contributors

• Cookiecutter attracts contributors at a faster rate than people, who have the desire and time available to maintain the project

• Thousands of users, hundreds of contributors, 5 maintainers
Learnings: Community

• Make your project easier to contribute to
• Adopt a Code of Conduct and enforce it
• Have empathy towards others and yourself
• Write good documentation
• Lead by example

@hackebrot
Project scope

• People use your software in ways in which you might not have intended or anticipated, but they are still valid use cases

• Newcomers encounter problems with Cookiecutter templates and create issues on the Cookiecutter issue tracker
Learnings: Project scope

- Encourage building tools on top of your project

- Ask contributors to develop automated tests and write documentation for new features in the same pull request

- Write documentation describing the project and its scope
Breaking changes

• Maintaining backwards compatibility becomes increasingly important
• All changes must be carefully considered and thoroughly reviewed
• Releases to PyPI, Conda, Mac Homebrew, Debian, …
Learnings: Breaking changes

- Be careful about adding new features
- Test your projects for supported Python versions and platforms
- Write documentation describing the project and its scope
- Say “NO” 😞
Workflows

• Asynchronous communication makes it difficult to make decisions

• We established guidelines for merging code to ensure new code is maintainable, idiomatic Python, tested and documented

• Lack of tools for maintainers on GitHub
Learnings: Workflows

- Set up meetings on IRC/Gitter/Slack or video calls for discussions that are hard to have over email or GitHub issue comments
- Document decisions made in a GitHub issue or the documentation
- Work towards making yourself redundant
- Reach out to other maintainers and suggest product enhancements

@hackebrot
Maintainers are humans too

• Maintainers are humans and volunteer their free time to lead the project

• They don’t always have a time or resources to dedicate to the project

• Every maintainer had their own reasons and motivations for taking on that responsibility and they might have changed over time
Learnings: Maintainers are humans too

• Stick to best-practices, use automated testing and code checks, add automation and integrations

• Document the process of bringing on new maintainers and guidelines for reviewing and merging changes

• Do not tolerate toxic and abusive behavior from community members!
Expectations

• Managing expectations is really hard! 😞
Learnings: Expectations

• Write documentation for different audiences
Frustrations for maintainers
Bike-shedding 🚴‍♂️
“We need this at work…” 😞
“You broke our deployment” 😞
“...and this ‘poyo’ introduction is a great example how direction of cookiecutter can seem immature to my eyes from times to times” 😞
“...can’t you simply...” 😐
“Not sure how much ‘sponsorship’ is needed, but here is a site for allowing the crowd to contribute as I know my organization would be interested in helping...”
Unseen work: running sprints, ordering stickers, exploring sponsorship, submitting talks 😞
Suggested solutions oftentimes require even more work from maintainers 😞
People do not respect boundaries and send direct emails rather than opening GitHub issues 📧
Maintaining open-source projects becomes work at some point, but without compensation or clear boundaries 🖥️
Empathy 😞
Take-aways for maintainers
Set expectations
Work towards making yourself redundant
Say “NO”!
Lead by example!
Do not tolerate toxic and abusive behavior from community members!
Allow yourself to take breaks or even walk away
Take care of yourself!
Thank you!
https://speakerdeck.com/hackebrot/the-challenges-of-maintaining-a-popular-open-source-project