Growing Open Source Seeds

Kenneth Reitz
Hi.
github.com/kennethreitz

- ~18 serious projects.
- 100+ experiments.
- OSX-GCC-Installer: 56TB of downloads.
- Requests: 3+ million downloads.
Other Interests...

• Street Photography
• Synthesizers and Music Production
• World Travel (~140,000 miles last year)
• Public Speaker (29 events last year)
• Classic Video Games!
Python Software Foundation
Once upon a time...
The Facebook SDK

- The Facebook SDK was rarely updated.
- One day, the library fully stopped working.
- Issues were opened, people offering to help.
- HN got wind, one issue had 50+ comments.
Facebook’s response was...

Disabling the Issue Tracker
That’s not open source.

It’s public source.
Public Source Project

- Organization takes a chunk of code and slaps an open source license on it, hoping it’ll be useful to someone else.
- Clearly better than keeping it closed source...
- Often abandoned due to lack of interest, burnout, or change of focus.
- Motivations are unclear.
Gittip
What is Gittip?

• Platform for sustainable crowd funding.
• It takes open source to the extreme.
• Striving to be the world’s first truly “open company”.
Extreme Open Source

• There’s a GitHub issue for everything.
• Major decisions are voted for on GitHub.
• Interviews with journalists are live-streamed.
• All formal discussions with other groups are publicly documented and shared.
I’m not building Gittip, I’m building the community that’s building Gittip.

— Chad Whitacre
Projects like Gittip are...

Shared Investment Projects
Shared Investments

• Shared ownership, extreme transparency.

• New contributors can get involved by following a documented process.

• Low risk. High bus-factor.

• See also: Python, Django, Firefox, jQuery...
  (projects that aim to change the world)
Requests
Requests
HTTP for Humans

```python
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf-8'
>>> r.encoding
'utf-8'
```
HTTP for Humans

- One of the most installed PyPi packages.
- Downloaded over 3,300,000 times.
- Comprised of contributions by 100+ developers.
- Considered by many to be an example of great open source.
• There’s a key difference between Requests and Gittip/Django...

• Zero of the project’s decisions are made by the community — they are made by me.
Between public source and shared investment:

Dictatorship Project
Dictatorship Projects

- A totalitarian BDFL that owns everything.
- Dictator is responsible for all decisions.
- Community feedback is encouraged, but users with feedback should have no expectation of change.
Requests’ Dictatorship

• Requests’ value lies in its extreme opinions.

• Few people involved allows for quick, precise iteration.

• Those opinions could be diluted if more people were involved with the decision making process.

• Tragedy of the commons.
There are downsides...

- The bus factor is extremely low.
- The risk of burnout is extremely high.
- If the BDFL loses sight of his goals, the project is ruined.
Lessons Learned
BE CORDIAL
OR
BE ON YOUR WAY
Contributors: Be Cordial

- Keep all interactions with a maintainer as respectful as possible.
- They have likely donated a significant amount of time and energy into their project.
- They don’t owe you a moment of their time.
Maintainers: Be Cordial

- You have the crucial responsibility of being immensely thankful to all contributors.
- They are the lifeblood of your project.
- Ignore non-constructive feedback.
- Some people just take things too seriously.
Maintainers: Be Cordial

- Be careful with the words you chose. Contributors sometimes take what you say very personally.

- Take the opportunity to educate the user.

- This could be their first ever interaction with an open source project.

- A little bit of kindness goes a long way.
Avoiding Burnout.
Sustainability

- Sustainability is one of the biggest challenges of open source.
- Everyone has a limited amount of time.
- It’s easy to become the bottleneck of your own projects.
Open source provides a unique opportunity for the trifecta of purpose, mastery and autonomy.

By recognizing the power of these factors, we can keep ourselves motivated and continue to increase our impact.

— Wes Beary
Meet My Minions

@lukasaoz @sigmavirus24

These guys are amazing.
Learn to Do Less

• When an issue or pull request comes into the repo, these guys usually triage it.

• This saves an immense amount of time.

• I can focus my time on larger issues, while they help contributors make the best of their time and efforts.
As I fork another of his projects, it occurs to me that I don’t program in the Python ecosystem: I program in the @kennethreitz ecosystem.

— Cory Benfield
Just Say No
Learn to Say No

- Saying ‘No’ is extremely difficult.
- People ask for crazy features. They send seemingly practical pull requests. They are trying to help.
- If you say yes often, your project will be ruined. Tragedy of the Commons.
Simplicity is always better than functionality.

— Pieter Hintjens
Learn to Say No

• Saying ‘No’ is extremely important.

• If a new pull request adds some nice features, but adds complexity, say no.

• I want as few lines of code in my projects as possible.
Simple Code is Good

- Code solves problems created by humans.
- The less code, the less to maintain.
- Negative diffs are the best diffs.
- Small, sharp, distributed services.
Complex Code is Bad

- Tight coupling, monolithic codebases.
- Lurking, growing technical debt.
- Maintenance burden is high.
- Self-serving instead of problem-solving.
Say ‘No’ to All The Things!
Open source makes the world a better place.
Don’t make it complicated.
Questions?
github.com/kennethreitz