Hard Truths About Open Source Community
who are we?

Matt Butcher
Principal Software Developer
Helm Co-Founder
Author
>200 FOSS Projects

Karen Chu
Community Manager
CNCF Ambassador
Meet-up Organizer
Virtual Kubelet
Service Mesh Interface
what it's all about?

• We’ve done many open source projects, large and small.
• We’ve collaborated with many people.
• Open source is hard to do well, and success in the open source world can even come with some heavy cost. Today, we wanted to share what we’ve learned about managing open source projects.
#1: more than code needs to be open

**NOT OPEN**
- Google Docs
- Internal issue queues

**OPEN**
- HackMD for meeting notes, docs, etc.
- Zoom for community dev calls
- Tracking issues in public
- Open governance model

**TOUGH CALLS**
- Slack
- YouTube
#2: multiple projects, same team

<table>
<thead>
<tr>
<th>PROS</th>
<th>CONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>• You know what you’re doing with the</td>
<td>• It can be hard to keep up with issue queues, conference CFPs,</td>
</tr>
<tr>
<td>same people – starting a project gets</td>
<td>community/dev calls, and communication across multiple teams</td>
</tr>
<tr>
<td>easier each time because you’ve built</td>
<td>• People assume you work on a single project full time (as opposed</td>
</tr>
<tr>
<td>up a formula</td>
<td>to being spread out on multiple projects)</td>
</tr>
<tr>
<td>• You can replay the formula each time</td>
<td></td>
</tr>
<tr>
<td>(e.g. you can run branding and feature</td>
<td></td>
</tr>
<tr>
<td>sets together)</td>
<td></td>
</tr>
<tr>
<td>• It’s possible, but long term, you’ll</td>
<td></td>
</tr>
<tr>
<td>have to increase the number of people</td>
<td></td>
</tr>
<tr>
<td>responsible (ideally with shared</td>
<td></td>
</tr>
<tr>
<td>ownership from the community) instead</td>
<td></td>
</tr>
<tr>
<td>of keeping the small team</td>
<td></td>
</tr>
</tbody>
</table>
#3: big community ≠ less work

- The bigger the project grows, the more people will be involved
- We thought by now, Helm would be driving itself... and it's not
- We thought that success meant lots of pull requests, and lots of pull request meant that we could sit back and observe. We were SOOOO wrong.
- What we discovered is...
  - The amount of Helm work goes up and down between the release.
  - Having correct expectations helps with planning.
  - After releases, you can go into "bug fixing mode" but at some point you have to decide whether to start feature coding again.
  - "The tyranny of the shiny": people's desire for new things makes them suspicious of stable software.

AS A COMMUNITY GROWS, GOVERNANCE BECOMES A BIG DEAL...
#3: big community ≠ less work

OPEN DECISION MAKING

- Realizing the vision of a project can sometimes be hard to balance with allowing the community to share the responsibility of growing the project.

- Importance of roadmaps and proposals – it helps the community/partners tie their roadmaps to specific milestones.

- The more successful a project is, the more people expect you to be open and document things.

- How much input do you give the community?
#4: bad actors in the community

- Lot of people spreading negativity
- Lots of misinformation
- Harassment

HOW CAN YOU DEAL WITH PEOPLE WHO ARE BAD ACTORS?

- Have a **Code of Conduct** – be consistent/uniform in enforcing your code; when laws are violated, get legal involved for advice
- Go on the record and make it clear bad actors can’t act or say things like that – norm bad behavior out of existence
- In a few cases, problems can be resolved by removing bad actors from the community
#5: branding is actually quite important

- Consistency branding across our projects – we use design to convey the relationship between Helm, Brigade, and Draft
#5: branding is actually quite important

- Project Github page vs. Project website
#5: branding is actually quite important

- CNAB website branding vs. Helm website branding
#5: branding is actually quite important

- We recommend trademarking your logo

<table>
<thead>
<tr>
<th>Helm</th>
<th>Phippy</th>
</tr>
</thead>
<tbody>
<tr>
<td>logo</td>
<td>mascot</td>
</tr>
<tr>
<td>“fair use”</td>
<td>Creative Commons (CC-BY)</td>
</tr>
<tr>
<td>grants people the right to use as appropriate</td>
<td>want others to share her/make her their own</td>
</tr>
<tr>
<td>not to be used in a negative context</td>
<td>can be used in a negative context</td>
</tr>
</tbody>
</table>
#6: success comes with criticism

**CRITICISM**

- When you pioneer a new space, you may enjoy the benefits early, but critics will catch up
- Helm was originally heavily criticized in several ways

**THINGS TO KEEP IN MIND**

- Seeing people post negative things can bring team morale down
- Helm Classic = 20 dl/mo | Helm 2.0 = 500 dl/mo | Helm 2.14/3.0 = 155,000 dl/mo
  - When you there are a lot of people in the community, someone is going to be unhappy
  - It may feel like there are lot of critics, it might also be a small portion in the grand scheme of things
- You not always hearing from the people who are satisfied
- Remember – there are two perspectives: some people who don't like your new way of doing things vs. there are some people who are always upset
#6: success comes with criticism

**HOW TO COMPETE, NOT CRITIQUE**

- Instead of tearing competing projects down, make your point by differentiate yourself from them in a way both projects can support

- DO NOT shame people into using something else and mislead about advantages/disadvantages

**AFTERMATH OF CRITICISM**

- Own up to criticism that’s true and fix it instead of sweeping it under the rug

- Even if you fix it, sometimes criticism stays around because bad news is stickier and the perception of your project may be tainted

- Be prepared to repeatedly share fixes for years before people realize it’s no longer an issue
#7: pick sustainable tooling

- Freemium/limited products may seem great but can end up causing lots of churn in your projects

**WHERE WE WENT WRONG...**

- Travis CI/CD pipelines
- Slack
- Waffle IO
- Tools we thought were free

**WHERE WE WENT RIGHT...**

- Netlify
- Zoom
- Helm based resources – instead of using personal/employee Azure accounts, we requested Microsoft donated Azure credits to the Cloud Native Computing Foundation where our project lives

**IF YOUR INTENTION IS TO KEEP COSTS DOWN, JUST KEEP IN MIND...**

- You will want to avoid taking away a tool many are using down the line
- You may have to switch tools when you become more successful – don’t wait too long to do it
lead with openness

• Open source is indeed about "putting yourself out there"
• You don't need to have all the answers or be the voice of authority.
• People want to help and want to contribute good ideas. And sometimes you have to validate that users have legitimate needs that your tool cannot meet.
• Tact is important: Say everything carefully. In issue queues. In blog posts. In chat. When you model that behavior, it will become part of the project culture.
• The single most important thing you can do is not write code, it’s protect your community.
7 Hard Truths About Open Source Community

Questions?

Matt Butcher, Microsoft Azure
Karen Chu, Microsoft Azure