主　题:	Re: Help Open Source Projects Better Evolve -- Need Your Help
发件人:	[redacted]
收件人:	"谭鑫" <tanxin16@pku.edu.cn>

First of all, this is great work :)

Second, there's a few people that I'd like to share this with (mostly researchers within Protocol Labs that have been looking at research on open source sustainability), would you mind if I shared it with them?

Now, let's talk about a few differences between the workflow adopted here and Node.js, a little bit of the history, because I think it can illuminate some of the factors that help scale the process:

> Commit right is a critical aspect of open source communities [29]. Communities are very cautious when giving someone a commit right [5].

While this is certainly true historically, there's been a lot of movement on this particular aspect of open source culture in newer communities on GitHub. Starting with the level community, projects that been experimenting with a model that gives commit rights away almost immediately. https://github.com/Level/community/blob/master/CONTRIBUTING.md#contributing

These experiments have exposed some of the historical aspects of tightly controlled commit rights that may no longer apply.

1. git is not cvs/subversion. In the old days of centralized version managers mistakes were common and costly. You could make a mistake and set back the project for days fixing it. By comparison, it's almost impossible to mess up a git repo. In the last 4 years of onboarding hundreds of committers to Node.js (i'll get into details later) I know of only one time that something was merged that actually messed up the repo and need to be fixed and the issue was resolved in ours.
2. "commit bit" and "governance" are related but don't need to be a single tier in the governance structure.

Node.js has what you might call a "3 tier" governance structure.

1. New contributors (people without commit rights)
2. Committers (often referred to as "Collaborators" using GitHub's terminology)
3. Technical Steering Committee members

The key to the success and scale of Node.js' contribution model is in how these tiers relate. New contributors are offered onboarding as committers quite early, after just a few contributions. Still, there are many more contributors who send something small and are never on-boarded as committers. This would be a huge burden for the maintainers if it weren't for the fact that there are more than 100 active committers to do reviews, many of which have relatively little experience but can still sufficiently review for style and grammar issues and basic code problems, which frees up the time of more experienced developers to handle only the reviews that are more technical.

In Node.js, nobody can merge their own Pull Request. This means that *at least* a single person reviewed code, but it's almost always the case that many people actually review it. All contributions will "be merged unless there are objections" and *any* committer can object and ask for a change, but the onus is on reviewers to point out issues. That vast majority of contributions are merged with only a few adjustments, the Technical Steering Committee is never involved in these contributions. It's only in the small number of contributions which cannot find an easy consensus that the PR is sent to the TSC which tries to reach a consensus but will fall back to a majority rules vote if necessary.

This is a 3 tier system in terms of governance, but in reality it's much more complicated. Individuals with varying degrees of expertise involve themselves in different aspects of review on different sub systems. Everyone is learning and leveling up and there's a constant flow of contributors to committers, committers to the TSC, and often TSC members that decide to move on to other work and move into Alumni status. The review process is part quality assurance and part education, and the educational aspects of the process are the only thing that ensures Node.js has a stead flow of new and competent maintainers.

Coming back to your findings.

The bar for contribution to Linux Kernel is so high that it provides a natural filtration mechanism. I doubt that kernel reviewers deal with contributions from people that need a lot of education or adjustment to basic style guide requirements, but for most open source projects on GitHub these contributions represent one of the largest burdens. Opening up to more maintainers certainly helps, but without another tier of governance you still end up overloading experienced contributors with a lot of remedial review tasks. The only solution I've seen to that problem is by on-boarding committers very early and creating another tier of governance for more senior developers which can focus on resolving only the hardest problems.

What's most interesting to me about the Node.js project is the areas in which this model has had the largest impact. Before we adopted this model it was believed that "outside contributors" (non-committers) would only want to add their pet feature and that they wouldn't be able to significantly impact testing and documentation as these are "boring" tasks. The exact opposite turned out to be true. New contributors are adjusting to both the technical and cultural aspects of a project, so they often will focus on documentation and testing because it's the easiest place to contribute and experience what the contribution workflow is like. A quick glance at the commits in any recent release gives you an ideal of the scale of improvements that happen to docs and tests, which means general stability and approachability improvements https://github.com/nodejs/node/releases

The issues maintainers have in open source projects outside of the Linux Kernel tend to be quite different, but the norm is of a "single maintainer project." Your work shows that moving to a multi-committer model could alleviate that, but it would have to account for some of the differences between average projects and the Linux Kernel, namely that there's a lot more interest for "casual contributors" and that these contributions make up the bulk of their review workload.

[redacted]

From: 谭鑫 <tanxin16@pku.edu.cn>
Sent: Tuesday, July 9, 2019 7:21 PM
[redacted]
Subject: Re: Re: Help Open Source Projects Better Evolve -- Need Your Help

[redacted]

I'm very happy that you are interested in our research.

The attachment is the manuscript of our research. Because we only studied a subsystem of the Linux kernel and confirmed the findings can be applied to the other subsystems within the Linux kernel, we don't know whether or not these findings could bring some insights to other OSS projects. I know Node.js had also experienced the stage of the rapid increase of committers (see the attachment numberofcommitter.png) and you have accumulated a lot of experience on how to scale Node.js. Therefore, we invite you to evaluate our findings, especially the findings of RQ2 (*What factors are crucial for successful implementation and operation of the new model?*).

1. Are these factors we found consistent with the preparations that Node.js did when scaling?
2. Do you think it is necessary to have an automatic tool that helps maintainers select candidate committers?
3. Do you think that our findings will bring some insights to other projects that want to move from a single-committer model to the multi-committer model?

Thank you very much! (Since our research has not yet been officially published, we kindly ask you not to disseminate it.)

Best wishes,

Xin Tan

> -----原始邮件-----
> [redacted]
> **发送时间:**2019-07-10 00:54:34 (星期三)
> **收件人:** "谭鑫" <tanxin16@pku.edu.cn>
> **抄送:**
> **主题:** Re: Help Open Source Projects Better Evolve -- Need Your Help

> Interesting, I’ d love to see the research :)
>
> -[redacted]
>
> ---
>
> **From:** 谭鑫 <tanxin16@pku.edu.cn>
> **Sent:** Tuesday, July 9, 2019 12:43 AM
> **To:** [redacted]
> **Subject:** Help Open Source Projects Better Evolve -- Need Your Help
>
> Hello Mikeal,
>
> I am a Ph.D student from Peking University. I am doing research about review mechanisms in open-source communities to help OSS projects evolve and sustain. I noticed that one of your blogs (https://medium.com/the-node-js-collection/healthy-open-source-967fa8be7951) discussed how to scale Node.js, which brings me much insight.
>
> We also did a lot of work to help projects better evolve. Please allow me to give a brief introduction to our study. The subsystem i915 of the Linux kernel also encountered the problem (maintainers are under heavy workload) like Node.js when it became larger and more complex. To solve this problem, it applied a new model --- the multiple-committer model instead of the single-maintainer model. We designed four dimensions of metrics to evaluate how this model worked. We find that this model conspicuously reduces the burden of the i915 maintainers and works well when more committers are introduced. We propose that overloaded subsystems having trustworthy candidate committers are suitable for adopting this model. The success of the model is closely related to a series of necessary guarantees, i.e., sufficient pre-commit testing, the strict review process, and applying tools to simplify work and avoid errors. We also proposed a network analysis approach to locate candidate committers. These results have been confirmed by the maintainers of other subsystems in the Linux kernel that wish to apply the multiple-committer model. If you want to know the details of this study, we would be happy to discuss this together.
>
> **What do you think of our research questions? Are our findings consistent with the preparations that Node.js did when scaling? Do you think that our findings will bring some insights to other projects that want to move from a single-committer model to the multi-committer model?**
>
> Looking forward to your reply!
>
> Best regards,
>
> Xin Tan