Positive Thoughts on the GitHub Acquisition

2018

2 min read • 423 words

I met (and became friends with) the GitHub crew when they were merely 8 employees, hacking away on their newly popular product. I’ve followed the team/product extremely closely over the years, with keen interest, and maintain relationships with some of the founders.

I’m also one of the top users on GitHub (depending on how you measure) — so I feel as though my opinion on this acquisition is worth sharing to the community.

This Will Be Good

Microsoft is a good company, now. They've been showing a lot of good–will towards both the developer community and specifically the open source community over the past several years. Their releases of SQL Server for Linux, .NET Core, Linux support on Azure, VSCode, and other projects (like financially supporting Requests!) have done nothing but prove to me that—demonstrating the evolution that sometimes I was ahead of my time in recognizing:

1. Microsoft understands that the world wants open source.
2. Microsoft understands the ethos of open source software, and doesn’t want to capitalize upon it by changing it.
3. Microsoft has the best of intentions, and I trust them.

That’s the gist of what I’m getting at here — there’s a vocal part of the developer community that is very concerned about this acquisition, and I think some of them have some valid concerns, namely—

- Microsoft is not known for tailoring good user experiences, unlike GitHub.

Rest assured, GitHub will be a separate functional unit from Microsoft, and will continue to operate as it has been, design-centered and all (shout out to the legacy of @kneath).

Git Community Implications

Because of the number of community members upset about the acquisition, a number of people are looking to host their code/projects elsewhere.

I think this is actually very healthy for the git ecosystem. Git is a bit over-saturated with GitHub usage — it’s a completely decentralized system, and people should be encouraged to run their own infrastructure for it, instead of “just use GitHub”.

So, I’m looking forward to GitHub still being the sane reasonable default, but not today’s reality of “uh, why aren’t you on GitHub?”.

Moving? GitLab vs. Phabricator

Anyone looking at GitLab as a replacement — I heavily recommend looking into running a Phabricator instance instead. It’s utterly fantastic software — Facebook uses it to manage their entire engineering workforce.

I personally run a Phabricator instance over at code.kennethreitz.org that mirrors all of my important repositories. In addition, I use it for higher-level project management.

So — in closing — have no fear & check out Phabricator.