Sarah Zhao
CSC 630
Dr. Zufelt
19 January 2018

On Maintainer Burnout

The rise of the open source movement has greatly benefited the technological world by making software products more affordable and accessible, allowing for rapid software development. In 2008, the founding of GitHub and the launch of StackOverflow standardized open-sourcing code and made open-source projects available to a wider audience. Programmers who previously worked on a handful of open-source projects could now easily contribute to hundreds. With these new resources, the open-source community of experienced programmers extended to amateur developers as well. Today on GitHub, anyone can contribute to an open-source project by raising issues or submitting pull requests.

With a large community of enthusiastic contributors to open-source software, many projects starting as small ideas grow popular and may receive hundreds of contributions a day. This onslaught of issues and pull requests from users may seem impossibly cluttered and unmanageable, but there are individuals in the open-source community who successfully manage and organize such hectic projects: project maintainers. In addition to contributing to the project with their own code and ideas, maintainers address user contributions one by one and run thorough tests on pull requests to ensure that merging will cause no issues with the original code. Without maintainers behind projects, the open-source community and the technological world depending on open-source products would quickly fall apart. Yet the majority of open-source users take the hard work of project maintainers for granted. Maintainers often manage their projects for free on their own time while balancing a full-time job. With little acknowledgment and an ever-growing number of issues and pull requests from users, open-source maintainers

often find themselves feeling guilty and stressed from all of the unaddressed user contributions. With no time or financial support and with a heavy emotional burden, increasingly more maintainers are growing disenchanted with both their open-source projects and the open-source community in general. Many, like Ryan Bigg, an award-winning author for technical books who also writes documentation for Ruby on Rails, are relinquishing their positions as maintainers of open-source projects.[1] This phenomenon, known as maintainer burnout, is quickly becoming a prominent problem in the open-source community. With the potential imminent deterioration of the open-source community due to increasing maintainer burnout, it is imperative to raise attention and seek solutions to burnout today.

While maintainer burnout has obvious implications in the open-source coding community as unmaintained projects become outdated and bug-ridden, maintainer burnout also greatly impacts corporations and governments that employ open-source products. Without well-functioning open-source software, these highly-influential organizations would not only fall behind technologically, but also risk missing serious bugs that may threaten user security. OpenSSL's Heartbleed is an example of such a bug. Initially released in 1998, OpenSSL is an open-source cryptographic software library that quickly gained popularity in the developer community. By 2014, roughly two-thirds of Web servers used OpenSSL for securing the passing of sensitive or private information over the Internet.[2] Despite the project's newfound popularity, only a small group of volunteers maintained the project, and the bulk of the work rested on two people's shoulders: Steve Marquess, a security consultant to the U.S. Department of Defense

[1] Ryan Bigg, "Open Source Work," *Ryan Bigg* (blog), entry posted November 16, 2015, accessed January 16, 2018, http://ryanbigg.com/2015/11/open-source-work.

[2] Dan Goodin, "Critical crypto bug in OpenSSL opens two-thirds of the Web to eavesdropping," *ArsTechnica*, accessed January 16, 2018, https://arstechnica.com/information-technology/2014/04/critical-crypto-bug-in-openssl-opens-two-thirds-of-the-web-to-eavesdropping/.

who joined the project in the early 2000s, and Stephen Henson, who worked full-time on OpenSSL since its initial release in 1998. Shortly after starting work on OpenSSL, Marquess realized that the project's revenue could not fully support even one person's work salary – in fact, although Henson was essentially in charge of maintaining the code, his salary was only a fifth of Marquess's government salary.[3] Marquess worked to increase revenue by establishing the OpenSSL Software Foundation (OSF); however, the money earned by these endeavors was still barely enough to pay Stephen Henson's single salary. With few resources and little support, Henson and Marquess, regardless of their talents, were bound to miss a bug or two in the immense code.

In 2014, two engineers separately discovered a major flaw in OpenSSL's software. The bug, later named Heartbleed, had gone unnoticed since a 2011 update and essentially allowed hackers to intercept supposedly secured sensitive data being passed to vulnerable web servers. Among those affected by this major security compromise were Instagram, Gmail, and Netflix.[4] Although Heartbleed was a serious security issue for most of the Web and attracted much negative media attention to the bug, OpenSSL's Heartbleed ironically improved the project's funding. The media reports drew the world's attention to the team behind OpenSSL and the project's lack of support and resources, and OpenSSL finally received the funding it desperately needed. Yet even this success may be fleeting once Heartbleed loses its limelight as Marquess remarks in an email interview, "I don't know that we [the OpenSSL team] can hold this together for more than a couple years."[5]

---

[3] Nadia Eghbal, *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure*, 11, accessed January 16, 2018, http://www.fordfoundation.org/media/2976/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure.pdf.

[4] Eghbal, *Roads and Bridges*, 13.

[5] Eghbal, *Roads and Bridges*, 15.

OpenSSL's struggles with funding highlights one of the main issues that open-source maintainers face today: the lack of time and financial support for open-source work. David MacIver, creator of Hypothesis, an open-source Python library for creating unit tests, addresses this problem as he says that he was able to spend time on Hypothesis because he had enough time and money only "because [he] spent the latter half of last year with double the salary [he] was used to, half the living expenses [he] was used to, and too borderline depressed to spend it on anything interesting. *These are not reasonable requirements.*"[6] Indeed, it is unreasonable for maintainers to remain underappreciated and often unpaid while sacrificing so much for the betterment of the wider community. Although the term open-source is sometimes mistaken to have the connotation of being "free" as in "free of charge," there are many open-source business models that have succeeded in ameliorating open-source maintainers' financial demands.

One of these business models is providing consulting services for users of the product. Red Hat, a software company providing open-source software products,[7] successfully sells subscriptions to product support, training, and other services. However, Nadia Eghbal, the author of *Roads and Bridges*, writes, "many are quick to point out that it is an anomaly unlikely to be repeated again" as "Red Hat benefited from first mover advantage for the technology it supports."[8] Moreover, this business model would be especially difficult for only one or two maintainers to uphold as it would detract from the time devoted to improving the open-source project itself.

Crowdfunding is another way maintainers could gain some financial support while not

---

[6] David R. MacIver, "It's OK for your open source library to be a bit shitty," entry posted April 8, 2015, accessed January 16, 2018, http://www.drmaciver.com/2015/04/its-ok-for-your-open-source-library-to-be-a-bit-shitty/.

[7] "Red Hat," Wikipedia, last modified January 3, 2018, accessed January 16, 2018, https://en.wikipedia.org/wiki/Red_Hat.

[8] Eghbal, *Roads and Bridges*, 92.

having to spend extra time earning money. Developers crowdfund by hosting fundraising events on websites like Kickstarter or Indiegogo to help facilitate work on open-source projects. Andrew Godwin, a Django core developer, describes the idea of crowdfunding as "twofold–to guarantee the project a solid period of work and at least 80 or so hours of coding time, as well as to try and show the world that open source software really can pay for developers' time." [9] While crowdfunding can effectively address smaller and more specific issues, such as bug fixes and new features, this business model would not be feasible for long-term projects that need continuous funding and support.[10]

A long-term business model that could potentially provide the funding and support needed is obtaining corporate sponsorship for a project. With corporate sponsorship, a company or other organization pays a developer to devote time on an open-source project. This paid open-source work could potentially solve both the time and money issue for maintainers with few repositories. However, companies usually fund developers to work on only one or two open-source projects, and this limitation would hinder the work of the majority of maintainers, who contribute to many more repositories. Another possible problem with corporate sponsorship is the possibility that the organization funding the work may exercise certain controls over the project. Strictly open-source maintainers may then view this company-modified version of the product as no longer pure and open-source, and thus they may refuse such corporate sponsorship.[11] Lastly, few companies actually do find sponsorships for developers to spend time on open-source code, which would mean not many opportunities for such a large community of

[9] Eghbal, *Roads and Bridges*, 98.

[10] Eghbal, *Roads and Bridges*, 98-99.

[11] Eghbal, *Roads and Bridges*, 103-104.

open-source developers.[12] While none of these three models present the ideal solution for the financial issues of an open-source maintainer, employing a mixture of these models, such as offering consulting on an open-source product or setting up a crowdfunding event for a new feature, could help alleviate the time and financial strain of many open-source maintainers.

In addition to the lack of time and financing for open-source work, the guilt and stress experienced by open-source maintainers is also a prominent cause of maintainer burnout that must also be addressed when concerned with the entire issue at hand. "More than anything, you feel the guilt: the guilt of knowing that you *could* have helped someone solve their problem, but instead you let their issue rot for months before closing it. Or the guilt of knowing that someone opened their first pull request *ever* on your repo, but you didn't have time to respond to it, and because of that, you may have permanently discouraged them from open source. You feel guilty for the work that you do, for the work that you didn't do, and for not recruiting more people to share in your unhappy guilt-ridden experience."[13] Lawson explains in detail on his blog how, when swamped with an impossible amount of work, maintainers may fall into a downward spiral as they may feel like their work is too shallow and unhelpful as they simply do not have the time to address every single issue and pull request with care.

In GitHub issues and repositories, such as isaacs/github issue#167: *Help open-source maintainers stay sane,* developers and other open-source contributors encumbered by the regrets and stress of maintaining open-source projects are discussing ways to improve GitHub user interface as well as ways to maintain a healthier mindset. In addition to the basic

---

[12] Eghbal, *Roads and Bridges*, 103.

[13] Nolan Lawson, "What it feels like to be an open-source maintainer," *Read the Tea Leaves: Software and Other Dark Arts, by Nolan Lawson* (blog), entry posted March 5, 2017, accessed January 16, 2018, https://nolanlawson.com/2017/03/05/what-it-feels-like-to-be-an-open-source-maintainer/.

recommendations of having well-written code of conduct, contributing, and perhaps frequently asked questions files in the repository, users have suggested ways that can help filter issues based off of date, number of user views and comments, or read/unread files.[14] In this manner, maintainers would not have to open their repositories to the equivalence of being confronted with "a line of a few hundred people […] patiently waiting for [the maintainer] to answer their questions, complaints, pull requests, and feature requests."[15] Others recommend sharing the work load with the community and also not pushing oneself too hard to reach the impossible "super-hero image open-source maintainers are given." [16]

Many of these suggestions for improving the lives of open-source maintainers, whether they be tailored towards the financial or emotional aspects leading to burnout, seem to entail outsiders' acknowledgement or help. From open-source users' demand for consulting to individuals' or companies' active funding of open-source work and from user-sorted issues to contributor support on the code, we, as users of or contributors to open-source code, play an essential role in helping make life easier for the maintainers holding up this community. Despite the countless useful technology stemming from the open-source community, the community remains run and maintained by humans with all our humanly flaws. Now, with the technological advancements in various avenues of research like those in artificial intelligence, perhaps the answer to maintainer burnout lies not only in us helping one another as a community to improve technology, but also in how technology can help us with open-source projects.

---

[14] Posts to GitHub, "Help open-source maintainers stay sane," April 12, 2014, accessed January 16, 2018, https://github.com/isaacs/github/issues/167.

[15] Lawson, "What it feels," *Read the Tea Leaves: Software and Other Dark Arts, by Nolan Lawson* (blog).

[16] Balupton, Posts to GitHub, "Help open-source maintainers stay sane," April 12, 2014, accessed January 16, 2018, https://github.com/isaacs/github/issues/167.

Bibliography

Anonymous post to GitHub web forum, "Help open-source maintainers stay sane," April 12, 2014. Accessed January 16, 2018. https://github.com/isaacs/github/issues/167.

Bigg, Ryan. "Open Source Work." *Ryan Bigg* (blog). Entry posted November 16, 2015. Accessed January 16, 2018. http://ryanbigg.com/2015/11/open-source-work.

Eghbal, Nadia. *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure*. Accessed January 16, 2018. http://www.fordfoundation.org/media/2976/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure.pdf.

Goodin, Dan. "Critical crypto bug in OpenSSL opens two-thirds of the Web to eavesdropping." *ArsTechnica*. Accessed January 16, 2018. https://arstechnica.com/information-technology/2014/04/critical-crypto-bug-in-openssl-opens-two-thirds-of-the-web-to-eavesdropping/.

Lawson, Nolan. "What it feels like to be an open-source maintainer." *Read the Tea Leaves: Software and Other Dark Arts, by Nolan Lawson* (blog). Entry posted March 5, 2017. Accessed January 16, 2018. https://nolanlawson.com/2017/03/05/what-it-feels-like-to-be-an-open-source-maintainer/.

MacIver, David R. "It's OK for your open source library to be a bit shitty." Entry posted April 8, 2015. Accessed January 16, 2018. http://www.drmaciver.com/2015/04/its-ok-for-your-open-source-library-to-be-a-bit-shitty/.

Marquess, Steve. "Of Money, Responsibility, and Pride." *Speeds and Feeds: Personal Musings of Steve Marquess* (blog). Entry posted April 12, 2014. Accessed January 16, 2018. http://veridicalsystems.com/blog/of-money-responsibility-and-pride/index.html.

"Red Hat." Wikipedia. Last modified January 3, 2018. Accessed January 16, 2018. https://en.wikipedia.org/wiki/Red_Hat.