Jungwoo Park

CS 630

Zufelt

11 January 2017

Forks

Forks and open source software seem to be inextricably linked. In fact, the freedom to legally fork without prior approval of those currently developing, managing, or distributing the software is written in the tenets of both free software and the open source movement:

> "The freedom to distribute copies of your modified versions to others (freedom 3). By doing this, you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this." -- The Free Software Definition[1]
>
> "3. Derived Works: The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software." -- The Open Source Definition[2]

The history of open source is full of software forks, and the reasons for forking a project are manifold. Some contributors fork because they want to take the project in a different technical direction, others fork because they are unsatisfied with the pace of the original maintainers, and still

[1] Free Software Foundation, "What is Free Software?," GNU Operating System, accessed January 19, 2018, https://www.gnu.org/philosophy/free-sw.en.html.

[2] Open Source Initiative, "The Open Source Definition," Open Source Initiative, accessed January 19, 2018, https://opensource.org/osd-annotated.

others fork due to irreconcilable personal differences. David A Wheeler notes[3] four possible outcomes of a fork, with examples:

1. The death of a fork (example: libc/glibc). This is by far the most common case, as it is easy to declare a fork, but difficult to continue independent development.
2. A re-merging of the fork (example: gcc/egcs). This is where the projects rejoin each other.
3. The death of the original (example: XFree86 gets replaced by X.org).
4. Successful branching -- both succeed, typically catering to different communities (example: GNU emacs/xemacs).

Many members of the community fully support forks and believe in their ability to catalyze change in open source software development. For example, according to The New Stack, Linus Torvalds, creator of the Linux kernel, wholly advocates forking: “Every time there is a fork, and I think forks are actually good things, it means somebody sees a need and a technical reason to do something different from the standard kernel.”[4] Other developers view forking as a strictly detrimental process due to its inherent tendency to divide the group of people working on a given project. Eric S. Raymond, in his essay *Homesteading the Noosphere*,[5] states that “The most important characteristic of a fork is that it spawns competing projects that cannot later exchange code, splitting the potential developer community.” He also notes in an entry for the Jargon File:

> Forking is considered a Bad Thing--not merely because it implies a lot of wasted effort in the future, but because forks tend to be

[3] David A. Wheeler, "Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!," David A. Wheeler, accessed January 19, 2018, https://www.dwheeler.com/oss_fs_why.html#forking.

[4] Swapnil Bhartiya, "May the Fork Be with You: A Short History of Open Source Forks," The New Stack, accessed January 19, 2018, https://thenewstack.io/may-fork-short-history-open-source-forks/.

[5] Eric S. Raymond, "Homesteading the Noosphere," Dvara, accessed January 19, 2018, http://www.dvara.net/hk/homesteading.pdf.

> accompanied by a great deal of strife and acrimony between the successor groups over issues of legitimacy, succession, and design direction.[6]

The community has yet to reach a consensus on whether or not forks should be encouraged. This paper examines three forks of open source software and draws conclusions about the role forks ought to play in the open source community.

Forking can allow developers to give neglected open source projects the support they need to succeed. OpenOffice.org, commonly known as OpenOffice, is a discontinued open source office suite created by Sun Microsystems in 2002. Members of the OpenOffice community who did not work at Sun Microsystems had wanted a more egalitarian form of the project for many years. Oracle Corporation purchased Sun in early 2010 and acquired OpenOffice in the process. Users initially welcomed the change in ownership; Sun's contributions to OpenOffice had been declining for some time, and the company's decision to release OpenOffice code to IBM under a proprietary contract, rather than under an open source license upset many contributors.[7] As time passed, however, OpenOffice community members grew increasingly concerned by Oracle's behavior towards open source software, Oracle's withdrawal of developers, and the company's lack of activity on or visible commitment to OpenOffice.[8] Simply put, OpenOffice did not fit anywhere in Oracle's plans. If project was to survive, something had to change -- quickly.

[6] Eric S. Raymond, "Forked," The Jargon File, accessed January 19, 2018, http://catb.org/jargon/html/F/forked.html.

[7] Simon Phipps, "OpenOffice.org and Contributor Agreements," LWN.net, accessed January 19, 2018, https://lwn.net/Articles/443989/.

[8] Richard Hillesley, "LibreOffice - A Fresh Page for OpenOffice," The H Open, accessed January 19, 2018, https://web.archive.org/web/20131206203304/http://www.h-online.com/open/features/LibreOffice-A-fresh-page-for-OpenOffice-1097358.html.

On September 2010, OpenOffice community members formed The Document Foundation as the host of LibreOffice, a new derivative of OpenOffice. In its initial announcement, The Document Foundation stated its concerns that Oracle would either discontinue OpenOffice or place restrictions on it as an open source project, like it had done on a different project by Sun, as the main reasons for forking.[9]Oracle demanded that all members of the OpenOffice.org Community Council involved with The Document Foundation step down from the council, claiming a conflict of interest.[10]

To Oracle's dismay, however, the passion behind The Document Foundation's bold move resonated with OpenOffice's user base. Many core developers quickly joined the The Document Foundation, and LibreOffice quickly became the default office suite for many major distributions; in August 2016, LibreOffice had an estimated 120 million users.[11] In April 2011, Oracle ended its development of OpenOffice.[12] Oracle most likely would have eventually closed OpenOffice down regardless of The Document Foundation, and LibreOffice is a prime example of how forking allows community members of a project to keep software they cared about alive.

Not all forks, however, result in happy endings. Systemd is one such fork that resulted in catastrophe. Calling systemd controversial could not be more of an understatement; the fork almost split the entire Linux community into two. MakeUseOf[13] and Hackernoon[14] provide records of the

[9] Barry Collins, "OpenOffice Group Breaks Away from Oracle," Alphr, accessed January 19, 2018, http://www.alphr.com/news/361516/openoffice-group-breaks-away-from-oracle.

[10] Gavin Clarke, "OpenOffice Files Oracle Divorce Papers," The Register, accessed January 19, 2018, https://www.theregister.co.uk/2010/09/28/openoffice_independence_from_oracle/.

[11] Italo Vignoli, "The Documentation Foundation Celebrates its First Anniversary," LibreOffice, accessed January 19, 2018, http://blog.documentfoundation.org/2011/09/28/the-document-foundation-celebrates-its-first-anniversary/.

[12]Ryan Paul, "Oracle gives up on OpenOffice after community forks the project," arsTechnica, accessed January 19, 2018, https://arstechnica.com/information-technology/2011/04/oracle-gives-up-on-ooo-after-community-forks-the-project/.

[13] Matthew Hughes, "Open Source Software and Forking: The Good, The Great and The Ugly," MUD, accessed January 19, 2018, https://www.makeuseof.com/tag/forking-good-great-ugly/.

fork’s messy history. Until recently, whenever a computer running Linux was booted up, it would first launch a program called init. Init would run until the computer was eventually shut down, and is responsible for launching the computer and handling orphaned processes. Init has a glaring flaw, however: its speed. Init launches tasks serially, or one at a time, meaning that for a new task to start, it first has to wait for the last one to finish. In 2015, when most computers had gigabytes of RAM, init was simply too slow. A new init system called systemd was proposed to solve this. Many members of the Linux community were unhappy with systemd for various reasons. Some people did not like the fact that systemd had massive scope creep. Specifically, they disliked that it tried to reimplement many existing services instead of improving or integrating existing ones. Other users and developers did not like the idea of software unnecessarily relying on systemd interfaces to work at all. For example, GNOME, a desktop environment, began to rely on systemd interfaces even though it technically did not need to. Still others did not like the management of the project. Finally, instead of doing one thing, and doing it well, systemd attempted to do multiple things besides simply boot the computer. In the eyes of the fork’s critics, this broke an important design philosophy of how system tools for Linux should work.

At this point, the Linux world split into two camps. One desperately wanted to ditch the old, slow, and outdated init daemon, while the other wanted nothing to do with systemd. Things got messy very quickly. Blogs declared the ‘end of Linux’ as they knew it.[15] Linus Torvalds, creator of the Linux kernel, went so far as to ban a key systemd developer from working on the kernel.[16] As more and more Linux distributions announced they would be moving to systemd, a group of

[14] David Clinton, "A brief overview and history of systemdâ—âthe Linux process manager," Hackernoon, accessed January 19, 2018, https://hackernoon.com/a-brief-overview-and-history-of-systemd-the-linux-process-manager-ca508bee4a33.

[15]"Systemd-redux," blog dot lusis, accessed January 19, 2018, http://blog.lusis.org/blog/2014/11/20/systemd-redux/.

[16] accessed January 19, 2018, https://lkml.org/lkml/2014/4/2/420.

frustrated programmer took matters into their own hands and launched a systemd-free fork of Debian, a Unix-like operating system, called Devuan. Although the Linux community has since recovered, its members do not forget the tumultuous, acrimonious struggle with systemd that almost brought one of the most culturally and commercially significant open source projects to its knees.

When malicious external factors threaten the integrity of a project, a fork can serve as a powerful method of reuniting users and developers against unsavory practices. According to *The Register*[17] and *Quartz*[18], one startup company learned of the unifying potential of forks the hard way. Kite is a development tools startup that makes a Python programming plugin, called Kite, for various code editors to boost developer productivity through automatic code completion and other enhancements. In order to market the plugin, Kite first looked to autocomplete-python, a widely-used, Jedi-based autocomplete engine for Atom.[19] The company purchased the plugin from its original developer, allowing its employees immediate push access for the project. After the acquisition, Kite developers began surreptitiously integrating their completion engine into the plugin. They added a screen that offers an ostensible choice between their engine and Jedi, which used to be the default, when a user attempts to install the plugin.[20] The catch is that the selection menu's interface design and wording clearly favor Kite over Jedi, which is presented as an unnamed, disparaged alternative. Many users and contributors of autocomplete-python took notice,[21] and the controversy created further outrage when it was revealed that Kite uploads users' code to their

[17] Thomas Claburn, "Python autocomplete-in-the-cloud tool Kite pushes into projects, gets stabbed with a fork," The Register, accessed January 19, 2018, https://www.theregister.co.uk/2017/07/25/kite_flies_into_a_fork/.

[18] Keith Collins, "This startup learned the hard way that you do not piss off open-source programmers," Quartz, accessed January 19, 2018, https://qz.com/1043614/this-startup-learned-the-hard-way-that-you-do-not-piss-off-open-source-programmers/.

[19] https://github.com/autocomplete-python/autocomplete-python

[20] https://github.com/autocomplete-python/autocomplete-python/pull/250

[21] https://github.com/autocomplete-python/autocomplete-python/issues/308

servers, raising privacy and security concerns. Autocomplete-python's GitHub README was eventually updated to state Kite's sponsorship of the package, but the incident had compromised the trust of many people connected to the project.

The autocomplete-python incident, however, was just the beginning of Kite's controversies. A few months after acquiring autocomplete-python, Kite targeted atom-minimap, a different Atom plugin that provides a zoomed-out view of code for easier navigation.[22] In April 2017, a Kite employee added a Kite promotion to atom-minimap that added links to Kite's website as a supposed service to users.[23] Despite heavy backlash from the community, Kite dismissed all requests to disable or remove the promotional links. The incident was exacerbated when Lee Dohm, community manager for Atom, suggested that Kite's links failed to meet his definition of advertising. Unlike the autocomplete-python situation, the users and developers of atom-minimap decided to take matters into their own hands. After further user statements that pointed out that the links were explicitly referred to as promotions in commit messages were ignored, developer Ryan Leckey decided to fork to code and create atom-minimap-plus, a version of the package without Kite.[24] News of Leckey's fork continued to spread, and increasing support for atom-minimap-plus placed further pressure on Kite to give into the developer's demands. Less than a month after the fork's creation, Kite promised to remove all of the changes it had made to atom-minimap. Leckey's package has been deprecated since, but it was never intended to completely overtake atom-minimap; rather, the fork represented a necessary and clear message that voiced the concerns of the ignored and brought the community closer together than ever before. Some users have questioned the necessity of creating ,

[22] https://github.com/atom-minimap/minimap

[23]https://github.com/atom-minimap/minimap/commit/16c11d82b889ce1260342e4fa7d6d1905c0fde45#diff-16b7d75b70953cbf4755508170f2f24c

[24] https://github.com/mehcode/atom-minimap-plus

which admittedly separated people connected to atom-minimap and impeded the original project's development. Had Leckey's fork not have provided a specific alternative to the Kite-controlled atom-minimap to switch to, Kite would not have had sufficient incentive to withdraw from the pull out from the original package. The fact that autocomplete-python is currently sponsored by Kite speaks volumes to the effectiveness of Leckey's decision to create a fork.

Like many other topics in the open source community, forks are neither definitively good nor always beneficial. Although forks generally occur as a response to a large problem that a project is facing, if they are handled poorly or hastily, like in the case of systemd, they can result in wasted resources and schisms within user and developer communities. At the same time, many forks have effectively replaced their predecessors and, in some cases, gone on to become successfully projects in their own right. Contrary to Eric Raymond's warnings, forks should not be discouraged or avoided; while they are not always successful, forks usually reflect the opinions and needs of a community, and no one knows that is best for a project better than its users and developers.

Bibliography

Accessed January 19, 2018. https://lkml.org/lkml/2014/4/2/420.

Bhartiya, Swapnil. "May the Fork Be with You: A Short History of Open Source Forks." The New Stack. Accessed January 19, 2018. https://thenewstack.io/may-fork-short-history-open-source-forks/.

Claburn, Thomas. "Python autocomplete-in-the-cloud tool Kite pushes into projects, gets stabbed with a fork." The Register. Accessed January 19, 2018. https://www.theregister.co.uk/2017/07/25/kite_flies_into_a_fork/.

Clarke, Gavin. "OpenOffice Files Oracle Divorce Papers." The Register. Accessed January 19, 2018. https://www.theregister.co.uk/2010/09/28/openoffice_independence_from_oracle/.

Clinton, David. "A brief overview and history of systemd?—?the Linux process manager." Hackernoon. Accessed January 19, 2018. https://hackernoon.com/a-brief-overview-and-history-of-systemd-the-linux-process-manager-ca508bee4a33.

Collins, Barry. "OpenOffice Group Breaks Away from Oracle." Alphr. Accessed January 19, 2018. http://www.alphr.com/news/361516/openoffice-group-breaks-away-from-oracle.

Collins, Keith. "This startup learned the hard way that you do not piss off open-source programmers." Quartz. Accessed January 19, 2018. https://qz.com/1043614/this-startup-learned-the-hard-way-that-you-do-not-piss-off-open-source-programmers/.

Free Software Foundation. "What is Free Software?" GNU Operating System. Accessed January 19, 2018. https://www.gnu.org/philosophy/free-sw.en.html.

Hillesley, Richard. "LibreOffice - A Fresh Page for OpenOffice." The H Open. Accessed January 19, 2018. https://web.archive.org/web/20131206203304/http://www.h-online.com/open/features/LibreOffice-A-fresh-page-for-OpenOffice-1097358.html.

Hughes, Matthew. "Open Source Software and Forking: The Good, The Great and The Ugly." MUD. Accessed January 19, 2018. https://www.makeuseof.com/tag/forking-good-great-ugly/.

Open Source Initiative. "The Open Source Definition." Open Source Initiative. Accessed January 19, 2018. https://opensource.org/osd-annotated.

Paul, Ryan. "Oracle gives up on OpenOffice after community forks the project." arsTechnica. Accessed January 19, 2018.

https://arstechnica.com/information-technology/2011/04/oracle-gives-up-on-ooo-after-community-forks-the-project/.

Phipps, Simon. "OpenOffice.org and Contributor Agreements." LWN.net. Accessed January 19, 2018. https://lwn.net/Articles/443989/.

Raymond, Eric S. "Forked." The Jargon File. Accessed January 19, 2018. http://catb.org/jargon/html/F/forked.html.

———. "Homesteading the Noosphere." Dvara. Accessed January 19, 2018. http://www.dvara.net/hk/homesteading.pdf.

"Systemd-redux." blog dot lusis. Accessed January 19, 2018. http://blog.lusis.org/blog/2014/11/20/systemd-redux/.

Vignoli, Italo. "The Documentation Foundation Celebrates its First Anniversary." LibreOffice. Accessed January 19, 2018. http://blog.documentfoundation.org/2011/09/28/the-document-foundation-celebrates-its-first-anniversary/.

Wheeler, David A. "Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!" David A. Wheeler. Accessed January 19, 2018. https://www.dwheeler.com/oss_fs_why.html#forking.