Analyzing the Motivations Behind Open Source Software Development

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Justin Zhang

Spring 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Caitlin D. Wylie, Department of Engineering and Society

Introduction

The Open Source Initiative (2006a), the *de facto* authority on open source (Open Source Initiative, 2006b), defines open-source software (OSS) as software with a license that allows free redistribution, modification, and access to un-obfuscated source code. 96% of codebases contain OSS (Synopsys, 2024), and almost all web servers use some sort of OSS, whether it be with the networking library, backend library, web framework, database, or operating system the web server is run on. Especially to techies like software developers, it seems like that any common software library, programming language, or server that you would use is open-source: closed-source software can feel like the exception.

This all seems too good to be true. If OSS is so useful and generally free, what do the developers of OSS gain from it? And if developers develop OSS for free, how do we know that they care enough to make OSS that's high-quality, secure, and otherwise "good?" It's almost impossible to think of anything else where someone can build a useful thing with completely free tools like one could with a website, or where the process and product of someone's livelihood is all freely available online for anyone to see.

I will be analyzing OSS with the theory of the Social Construction of Technology (SCOT), which argues that human motivations and action shape the development of technology (Bijker & Pinch, 1987). SCOT especially examines humans through the lens of social groups, which each have their own agenda and want different things from a specific technology. A given person doesn't necessarily belong to any social group or is limited to one social group; in fact, the intersection between different social groups can give a lot of insight into the development of OSS. The most obvious social groups relevant to OSS are the developers and the users, but the more specific social groups I will be analyzing are OSS advocates, smaller companies, larger

companies, corporate developers, and freelance developers. Even though open-source software can have its issues, through an analysis with SCOT, we can see how the values of different social groups contribute to open-source software with its development, funding, and security.

The Developers of Open-Source Software

Open-source software is not a recent phenomenon, though it was not always as popular as it is today. Software began in academia and research, which generally facilitates open collaboration. Software was not considered copyrightable until *Apple Computer, Inc. v. Franklin Computer Corp.* in 1974 (Nussbaum, 1984). As software development grew as a field and costs increased, proprietary software became dominant, especially led by Microsoft and IBM (Cantrill, 2012); Bill Gates wrote his now infamous *An Open Letter to Hobbyists* in 1976, accusing hobbyists of the "stealing" and "theft" of their proprietary software. Richard Stallman started the GNU project in 1983 to allow a user to run a computer on solely free and open-source software. Stallman also wrote the GNU General Public License (GPL), under which a lot of OSS continues to be licensed to this day. The Linux kernel was released under an open-source license by Linus Torvalds in 1992 to make it compatible with GNU's licensing, and when combined with GNU, it provides a fully free and open source operating system (Torvalds 1992). A vast majority of companies use OSS, and the use of OSS is only growing (SolutionsHub, 2023).

Advocates like Torvalds and Stallman were and are a significant driver of open source, especially in the late 20th century when open source was less widespread than it is today. Advocates care strongly about the principles behind open source, and they become effective by spreading their vision to a wide audience and changing the behavior of others, specifically to have others help develop OSS. The most widely known OSS advocates themselves tend to be very skilled in development: Torvalds pioneered the development of Linux, the most used

operating system for development today, with the source code repository still under his account on GitHub (Torvalds, 2011/2024). Because of their lower numbers, advocates still need others to develop OSS, which Torvalds has successfully done with Linux. In an interview, Torvalds said "I spend most of my day merging code others wrote," as opposed to writing code himself. Advocates also in part inspire others by maintaining their presence in the public consciousness. xkcd is a webcomic referenced in nearly every computer science class (and some other engineering classes!) I've been in, and it references Stallman enough to justify the whole category of "Comics featuring Richard Stallman" on the Explain xkcd Wiki, with 11 comics in the category (2024).

Whether or not an industry primarily uses proprietary or open-source software isn't always static, especially if the social group of the users is technologically inclined. For music notation software, the three market leaders are Sibelius, Finale, and Dorico (Cliff, 2019). However, MuseScore Studio, a free and open-source music notation software, has seen impressive growth over the past years: the number of its downloads has gone from 200/day in 2008, to 7000/day in 2016, to 12 million total downloads as listed on their website now in 2024 (Pisano, 2008; Weiss, 2016; Muse Group, n.d.). It has gained traction with users that want their voice heard in the software design process and feels that proprietary software like Sibelius and Finale are too complacent in their place as market leaders. Martin Keary runs a YouTube channel called Tantacrul with many popular videos critiquing the UI/UX design of notation software like Sibelius and Finale, and his very public critiques has led Keary to spearhead the design of MuseScore 4 and become the VP of Product at MuseScore Studio (Keary, n.d.-a, n.d.-b). Especially because of Keary's background both as a musician a software developer, and others

with similar background that contribute to MuseScore Studio's development, the open-source option for music notation software only continues to grow and become better (Keary, n.d.-a).

The size of the intersection between users and possible developers of a type of software has the greatest influence on whether open-source or closed-source software dominates a specific industry. For software libraries mainly consumed by software developers (like web frameworks and networking libraries), OSS dominates. Out of the top ten programming languages coders have used in the past year, eight are open source (Stack Overflow, 2023). However, for software whose end users aren't necessarily software developers, like music notation software and Microsoft Office, OSS is not as popular. This supports the idea in SCOT that social groups are core to how a technology is developed: for software where there exists a large social group acting as both producer and consumer (i.e. an large intersection between users and developers), that social group has a vested interest in developing software with more direct communication with their consumer peers. This is best facilitated by personally helping to develop OSS, rather than looking to be hired by a company that develops closed-source software. Even if such an individual in a social group only cares about their own benefit, OSS allows the transparency and flexibility for that person to work on a feature of their software. This can be seen with the contributors to the Linux kernel. Linus Torvalds, the lead developer of Linux, said in an interview, "Every release, about half the people involved send just one patch, and a lot of them never show up again. They may have something small they wanted to fix that they cared about. … They were not interested in doing anything more" (Torvalds & Hohndel, 2023).

The tech-savvy software developers tend to advocate for and use open-source software more than other groups, mainly because they have the skills to make improvements to the open-source software they use as they see fit. This is why software developers use text editing

software like Vim or NeoVim, editors that have almost infinite customizability but require some technical skill to unlock their full potential, as opposed to editors like Word which has a lot of features but is much less customizable. This trend even applies to developers at different stages of learning: in the 2023 Stack Overflow Developer Survey, 23% of professional developers use Vim, while only 17% of people learning to code use Vim. Out of all editors, Neovim also has the highest proportion of users that want to continue using it at 81%, even beating out the most used and popular Visual Studio Code at 77%.

Proprietary software is still booming, especially in industries where technology isn't necessarily the main focus. As we saw earlier, in music notation software, the proprietary products, Sibelius, Finale, and Dorico, are still the most commonly used (Cliff, 2019). The Microsoft Office suite, with closed-source software such as Word and Excel, is widely used both educationally and professionally, despite the existence of open-source alternatives such as LibreOffice and OpenOffice. Windows is also used by half of software developers and has three-quarters of the operating system market share, and it is and always has been closed-source (Stack Overflow, 2023; Sherif, 2024).

Nonetheless, almost all software has an open-source version, such as LibreOffice for Microsoft Office, since the intersection between users and possible developers is almost never 0. Because of the much smaller intersection between social groups, however, LibreOffice is significantly behind Microsoft Office in market share, with its "10s of millions" of users being less than ten percent of the more than a billion Office users (LibreOffice, n.d.; Callaham, 2016). OSS users that persist despite alternatives with vastly greater technical support are usually OSS advocates that are willing to sacrifice convenience for the transparency of open source. As one

person says on Reddit, they view Office as “invading privacy” and thus advocate for LibreOffice instead (WhereWillIt3nd, 2023).

Funding of Open-Source Software

Funding is and was the biggest barrier to OSS. There’s not much to dislike about OSS in principle: consumers (which includes most developers) benefit from the free, no-strings-attached redistribution of software. However, developers still need to make a living and find strategies for funding other than just charging for their software. In *An Open Letter to Hobbyists* (1976), Bill Gates’s main plea to hobbyists that stole Microsoft’s software was that paying for their software was the only way quality software for hobbyists could be made: “Who can afford to do professional work for nothing? What hobbyist can put 3-man years into … his product and distribute it for free?” Although his letter wasn’t attacking open source but rather piracy, Gates’s argument stands for why he believed his software should be paid for. The most direct way of supporting yourself is charging for what you create, but OSS developers need to be more creative than this.

Sometimes the problem of funding can end up leading to open-source software turning closed-source. AWS used the then open-source Elasticsearch software to power their own cloud service offering that directly competed against the same cloud service offering from Elastic, Elasticsearch’s parent company. Because AWS’s Elasticsearch took revenue away from Elastic, Elastic decided to continue developing Elasticsearch as a closed-source software product (Banon, 2021). AWS then forked the Elasticsearch source code, creating a copy of it, and continues to maintain it as an open-source project with the name AWS OpenSearch. They are now direct competitors, with Elasticsearch making claims about its faster performance compared to AWS

(Elastic, n.d.). OSS companies and developers thus need to be careful with monetization to not be pushed out by larger companies that could the same OSS to compete.

The social groups of large companies and developers of larger projects generally have an easier time with funding OSS. Once enough people buy into a certain OSS, like with Linux, OSS companies/developers can solicit money from sponsorships or provide technical support for a price (Cox 2023). Larger companies that mainly focus on proprietary software can also still have incentives to maintain OSS projects. Meta benefits from supporting React, an open-source web framework, because of its vast contributor base and its prevalence in web development as a whole. This can make finding React software developers easier since so many people use React already, and interested non-Meta developers can still contribute to React and help make it better for everyone, including Meta. AWS maintains s2n-tls, an open-source networking library that doesn't have nearly as many non-company (non-AWS) contributors as React, but maintaining it as OSS builds customer trust (GitHub, 2024a). As long as these companies share enough of the same motivations of the broad OSS developer social group, such as collaboration for Meta and transparency with s2n-tls, these large companies, as their own social group, can fund their own OSS projects.

Finally, for developers working on smaller hobby projects, funding isn't as big of a problem, since they get personal gain from working on the project and mostly aren't working on an OSS project full-tim. As surveyed by Stack Overflow (2023), 70% of developers code for hobby projects outside of work. A notable example of a hobby project is Vim, the most popular command-line text editor, with around 22% of developers using it (Stack Overflow, 2023). Bram Moolenaar, most commonly referred to as just "Bram" by users of Vim, was the single core maintainer of Vim and wrote most of its source code, since it was the culmination of his personal

vision (*Contributors to Vim/Vim*, n.d.). Vim's only monetary solicitation is an ask to support children in Uganda, displayed when the program is first opened. The passion and wants of a single individual can drastically change the OSS space, and lead to benefits for everyone, as the advocates and original designed of OSS intended. Because of the open-source format, developers can also remix existing OSS. Neovim is a fork of Vim created in 2015 designed to make the program more flexible and extensible, and it coexists alongside Vim in the code editor space (Neovim, n.d.-a). It's funded mainly through sponsors via GitHub, and although funding does go directly to some contributors, it's not nearly as much as a full-time developer salary (Neovim, n.d.-b; Open Collective, 2024). OSS funding exists on a spectrum, with developers being able to get funding of this type for part-time work, which also makes it easier for developers to transition to large full-time role for a project. Even smaller projects are various Neovim plugins, most of which have no financial backing; they are developed and maintained by people that use those plugins and care about their development for their own sake.

Security of Open-Source Software

When an OSS vulnerability is publicized, it can be especially alarming. Heartbleed was a software vulnerability that allowed anyone to get passwords, addresses, encryption keys, and other sensitive information from more than a third of the world's websites (Synopsys, 2020). Kerner (2014) estimated the cost of Heartbleed to be $500 million. Surprisingly, it is possible to track down who wrote the single line of code responsible for Heartbleed, since it was a vulnerability in an OSS library, OpenSSL (Henson, 2014). However unlikely, this person could have intentionally hidden Heartbleed in a way that would get it past code reviews by other contributors. All OSS, which again most codebases depend on in at least some way, is vulnerable to such bad actors (Synopsys, 2024). However, security flaws in OSS like Heartbleed have such

a large impact because OSS is so widely used, not necessarily because OSS is worse at security. For example, in March 2024 alone, there were 70 new known security vulnerabilities for Microsoft products published in the CVE database (*Microsoft : Security Vulnerabilities, CVEs Published in March 2024*, n.d.). This number is likely even deflated because of the lower transparency into closed-source, proprietary software.

All stakeholders advocate for the security of OSS, but generally, larger stakeholders of OSS care more. Large companies like AWS want high security for their systems. AWS in the past decade is replacing most of their internal networking libraries, even dedicating their own resources to replace OSS like OpenSSL with their own open-source networking library, s2n-tls, because of how large projects with many features like OpenSSL can cause oversights that lead to disasters like Heartbleed (Schmidt, 2015). OpenSSL itself is also a project with large stakeholders, and the discovery of Heartbleed was only announced the day the OpenSSL update fixing the bug was released (Henson, 2014; Synopsys, 2020). Smaller stakeholders, on the other hand, don't care as much about security and only use OSS as an easy-to-access software tool. This can lead to irresponsible use of OSS, especially when people don't update OSS to its latest versions to avoid dependency issues. In commercial codebases analyzed by Synopsys in 2023, 89% had OSS more than 4 years out of date, and 84% had at least one vulnerability.

With proper use, OSS generally can result in secure software, despite most OSS projects allowing contributions from anyone (with approval). Anyone that wants to can look at the source code of OSS to look for security vulnerabilities, so potential security vulnerabilities can be shared more openly and faster than with closed-source software. For example, Synopsys publicizes its own Open Source Security and Risk Analysis report every year, which is only possible because of open-source source code. Closed-source software, on the other hand,

prevents potential attackers from analyzing code to perform attacks, but security audits are less trustworthy. Open source generally encourages innovation in the field of security, resulting in breakthroughs like blockchain and cryptocurrency, decentralized platforms with security and community ownership as core tenets.

Conclusion

Open-source software is already ingrained in our technology and in our lives. A core idea of SCOT is closure, where social groups perceive a problem as being solved and innovation slows (Bijker & Pinch, 1987). During the early formative years of software development, individual OSS contributors had a massive impact on the future of OSS since their ideas were so different from their status quo, with much of their core OSS concepts and artifacts (like licenses) still lasting to this day. OSS has become more and more integrated into technology now, evidenced by its widespread use with the many examples already discussed. Many people learning how to code start with OSS, such as with React discussed above. Proprietary, closed-source software, however, does dominate certain types of applications, especially when the most relevant social groups to those applications don't have the interest/human capital available to develop OSS. Overall, both OSS and proprietary software have achieved a decent amount of closure: both types of software have their uses for the social groups responsible for each.

Non-technical users generally don't need to fret about the quality and security of OSS. There is a reason most software either is open source or relies on open-source software. Using SCOT, we can see that despite potential drawbacks, open-source software is useful to almost all social groups; everyone has a stake in OSS. As the world becomes more technological, OSS will only become more prevalent, and we should be optimistic in its future.

References:

Banon, S. (2021, January 19). *Amazon: NOT OK - why we had to change Elastic licensing*. Elastic Blog. https://www.elastic.co/blog/why-license-change-aws

Bijker, W., & Pinch, T. (1987). The Social Construction of Facts and Artifacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other. In *Social Studies of Science* (Vol. 14, pp. 17–50).

Bryan Cantrill. (2012, July 26). *Corporate Open Source Anti-patterns*. Fórum Internacional Software Livre. https://www.youtube.com/watch?v=Pm8P4oCIY3g

Callaham, J. (2016, March 31). *There are now 1.2 billion Office users and 60 million Office 365 commercial customers*. Windows Central. https://www.windowscentral.com/there-are-now-12-billion-office-users-60-million-office-365-commercial-customers

Category:Comics featuring Richard Stallman—Explain xkcd. (n.d.). Retrieved April 29, 2024, from https://www.explainxkcd.com/wiki/index.php/Category:Comics_featuring_Richard_Stallman

Cliff, T. (2019, December 1). *Tech Reviews: Dorico Pro3*. Music Teacher. https://www.musicteachermagazine.co.uk/review/article/tech-reviews-dorico-pro3

Contributors to vim/vim. (n.d.). GitHub. Retrieved April 30, 2024, from https://github.com/vim/vim/graphs/contributors

Cox, M. (2023, August 8). OpenSSL Finances. *OpenSSL Blog*. https://www.openssl.org/blog/blog/2023/08/08/finances/index.html

Elastic. (n.d.). *Amazon OpenSearch vs. Elasticsearch*. Elastic. https://www.elastic.co/amazon-opensearch-service

Gates, B. (1976, February 3). *An Open Letter to Hobbyists*. https://archive.nytimes.com/www.nytimes.com/library/cyber/surf/072397mind-letter.html

GitHub. (2024a, February 8). *Contributors to aws/s2n-tls*. GitHub. https://github.com/aws/s2n-tls/graphs/contributors

GitHub. (2024b, February 8). *Contributors to facebook/react*. https://github.com/facebook/react/graphs/contributors

Henson, S. (2014, April 7). *Add heartbeat extension bounds check*. Git.Openssl.Org. https://git.openssl.org/gitweb/?p=openssl.git;a=commit;h=96db9023b881d7cd9f379b0c154650d6c108e9a3

Keary, M. (n.d.-a). *About Me*. Tantacrul. Retrieved April 29, 2024, from https://www.martinkeary.com/about-me

Keary, M. (n.d.-b). *Videos [Youtube Channel]*. Youtube. Retrieved April 29, 2024, from https://www.youtube.com/@Tantacrul/videos

Kerner, S. M. (2014, April 19). *Heartbleed SSL Flaw's True Cost Will Take Time to Tally*. eWEEK. https://www.eweek.com/security/heartbleed-ssl-flaw-s-true-cost-will-take-time-to-tally/

Kopriva, J. (n.d.). *Heartbleed, BlueKeep and other vulnerabilities that didn't disappear just because we don't talk about them anymore*. SANS Internet Storm Center. https://isc.sans.edu/diary/26798

LibreOffice. (n.d.). *Who uses LibreOffice?* LibreOffice. Retrieved February 27, 2024, from https://www.libreoffice.org/discover/who-uses-libreoffice/

Linus Torvalds & Dirk Hohndel. (2023, December 7). *Keynote: Linux Torvalds, Creator of Linux & Git, in Conversation with Dirk Hohndel* [Video]. https://www.youtube.com/watch?v=OvuEYtkOH88

Maracke, C. (2019). Free and Open Source Software and FRAND-based patent licenses. *The Journal of World Intellectual Property*, *22*(3–4), 78–102. https://doi.org/10.1111/jwip.12114

Marko, K. (2021, April 21). *Elastic vs. AWS highlights open source monetization dilemma*. TechTarget. https://www.techtarget.com/searchcloudcomputing/opinion/Elastic-vs-AWS-highlights-open-source-monetization-dilemma

Meta. (n.d.). *React*. Meta Open Source. https://opensource.fb.com/projects/react/

Microsoft: Security vulnerabilities, CVEs published in March 2024. (n.d.). CVEdetails.Com. Retrieved April 30, 2024, from https://www.cvedetails.com/vulnerability-list/vendor_id-26/Microsoft.html?page=1&year=2024&month=3&order=1&trc=70&sha=5a52033953366e274c031af1d12435fd5e441fb3

Muse Group. (n.d.). *MuseScore Studio*. Muse Group. Retrieved March 20, 2024, from https://www.mu.se/musescore-studio

Neovim. (n.d.-a). *About*. Neovim. https://neovim.io/charter/

Neovim. (n.d.-b). *Sponsors*. Neovim. https://neovim.io/sponsors/

Nussbaum, J. L. (1984). *Apple Computer, Inc. V. Franklin Computer Corporation Puts the Byte Back into Copyright Protection for Computer Programs*. *14*, 281–308.

Open Collective. (2024, January 16). *Transactions*. Neovim - Open Collective. https://opencollective.com/neovim/transactions

Open Source Initiative. (2006a, July 7). *The Open Source Definition*. Open Source Initiative. https://opensource.org/osd

Open Source Initiative. (2006b, September 19). *About*. Open Source Initiative. https://opensource.org/about

Pisano, J. M. (2008, October 7). *MuseScore, A Free Open-Souce Music Composition And Notation Program—MusTech.Net—Music Education & Technology*. MusTech.Net. https://web.archive.org/web/20221207193403/http://mustech.net/2008/10/musescore-a-free-open-souce-music/

React. (n.d.). *Meet the Team*. React. https://react.dev/community/team

Schmidt, S. (2015, June 30). *Introducing s2n-tls, a New Open Source TLS Implementation | AWS Security Blog*. AWS. https://aws.amazon.com/blogs/security/introducing-s2n-a-new-open-source-tls-implementation/

SolutionsHub. (2023, January 30). *What is Open-Source Software?* EPAM SolutionsHub. https://solutionshub.epam.com/blog/post/what-is-open-source-software

Stack Overflow. (2023). *Stack Overflow Developer Survey 2023*. Stack Overflow. https://survey.stackoverflow.co/2023/

Synopsys. (2020, June 3). *The Heartbleed Bug*. https://heartbleed.com/

Synopsys. (2023). *Open Source Security and Risk Analysis 2023*.

Synopsys. (2024). *2024 Open Source Security and Risk Analysis Report*.

Thompson, C. (2023, February 14). *How Rust went from a side project to the world's most-loved programming language*. MIT Technology Review. https://www.technologyreview.com/2023/02/14/1067869/rust-worlds-fastest-growing-programming-language/

Torvalds, L. (1992, January 15). *Release notes for Linux 0.12*. The Linux Kernel Archives. https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12

Torvalds, L. (2024). *Torvalds/linux* [C]. https://github.com/torvalds/linux (Original work published 2011)

Twitter. (2023, March 31). *Twitter's Recommendation Algorithm* [X Engineering]. https://blog.twitter.com/engineering/en_us/topics/open-source/2023/twitter-recommendation-algorithm

Vaughan-Nichols, S. (2015, October 15). *Can the Internet exist without Linux?* ZDNET. https://www.zdnet.com/home-and-office/networking/can-the-internet-exist-without-linux/

Weiss, I. (n.d.). *MuseScore in 2016: The year in review*. MuseScore.Org. Retrieved March 20, 2024, from https://musescore.org/en/user/101731/blog/2016/12/27/musescore-2016-year-review

WhereWillIt3nd. (2023, July 29). *[Serious] Does LibreOffice really have much of a future?* [Reddit Post]. R/Libreoffice. www.reddit.com/r/libreoffice/comments/15cvjq0/serious_does_libreoffice_really_have_much_of_a/