Who Will Pay the Piper for Open Source Software Maintenance?

Can We Increase Reliability as We Increase Reliance?

DAN GEER AND GEORGE P. SIENIAWSKI

Dan Geer is the CISO for In-Q-Tel and a security researcher with a quantitative bent. He has a long history with the USENIX Association, including officer positions, program committees, etc. dan@geer.org

George P. Sieniawski is a technologist at In-Q-Tel Labs, which develops open source tools and data sets that address challenges at the intersection of national security, the public interest, and the private sector. He specializes in data visualization research and prototype development for a wide variety of use cases. GSieniawski@iqt.org

"A little neglect may breed mischief ...
for want of a nail, the shoe was lost;
for want of a shoe, the horse was lost;
and for want of a horse, the rider was lost."

—Benjamin Franklin, Poor Richard's Almanac (1758)

As software eats the world and open source eats software, IT supply chains and enterprise risk management postures are evolving. Top-down, CIO-led commercial software procurement is shifting towards bottom-up, developer-driven choices that increasingly involve open source software (OSS) [1]. Security in this context requires visibility, starting with a comprehensive inventory (software bill of materials) as well as an understanding of code provenance (software composition analysis). It also entails application testing, automated vulnerability scanning, instrumentation, and observability, which can provide insights for defenders. For organizations that plan over longer time horizons, however, mitigating OSS risk sometimes means taking on direct responsibility for software maintenance. Little by little, organizations are empowering staff to perform upstream code improvements that the rest of the world can access. When implemented thoughtfully, this pragmatic form of software stewardship can help avoid broken builds, obsolescence, and other potential failure modes.

In a rough count by the authors, we found that at least one-third of Fortune 500 firms have a public Git presence for company-sanctioned OSS activity [2]. While proprietary software use remains widespread, and while many more companies use private repositories for internal collaboration projects, or inner-source, many high-profile enterprise software development efforts are now happening in the open under permissive license terms. A similar pattern appears to be unfolding in the public sector, albeit at a more gradual pace. NASA, the GSA, the Department of Transportation, and the Department of Energy, for instance, have earned high marks on the code.gov agency compliance dashboard for their performance under the Federal Source Code Policy. Other federal agencies are taking more incremental steps in adapting OSS to their missions, and these initiatives are likely to remain a continual work-in-progress. With commercial and governmental enterprises mostly consuming but increasingly producing OSS, and with shared source code resources circulating across both types of Git repos, knowledge spillovers [3] appear to be reshaping a wide variety of software development communities. Silicon Valley is playing a prominent role in this arena, and as the Linux Foundation's Core Infrastructure Initiative recently noted, "some of the most active OSS developers contribute to projects under their Microsoft, Google, IBM, or Intel employee email addresses" [4].

Whether public or private, funding for OSS can help underwrite open innovation, reduce security costs, and amortize technical debt, but Red Hat's Gordon Haff reminds us: "Open source today is not peace, love, and Linux" [5]. Fiscal sponsorship can skew incentives in

Top 50 Packages (for each package manager)	Primary Language	Language Rank,* 2019	Language Rank,* 2018	Average Dependent Projects	Average Direct Contributors
npm	JS	1	1	3,500,000	35
Pip	Python	2	3	78,000	204
Maven	Java	3	2	167,000	99
NuGet	.NET/C++	6	5	94,000	109
RubyGems	Ruby	10	10	737,000	146

Table 1: Concentration of GitHub contributions. *Popularity ranked by number of unique contributors to public and private GitHub repositories tagged with the corresponding primary language. Source: GitHub, *State of the Octoverse* (https://octoverse.github.com/#average-package-contributors-and-dependencies), released Nov. 6, 2019 (a few months before GitHub acquired npm).

unexpected ways since OSS backers are in a position to influence feature prioritization and project governance. As organizations start treating user-driven open source development as a regular operating expense, some developers worry about ecosystem fragmentation, value capture, and selective appropriation of benefits. Indeed, the advent of new software funding vehicles and managed open source subscription plans has drawn comparisons to gentrification and gerrymandering [6]. Consequently, organizations looking to engage with OSS communities around the world need to understand developer motivations, which are distinct from ownership and contract [7] and which involve a mix of pecuniary, reputational, and "own-use"/DIY reasons.

As Internet researcher Nadia Eghbal rightly recognizes, the OSS community's "volunteer culture discourages talk of money" [8]. Moreover, "The pervasive belief, even among stakeholders such as software companies, that open source is well-funded, makes it harder to generate support" for fledgling projects. It also highlights the need to find a balance between bearing private cost and conferring public benefit, which is the crux of open source stewardship. In the years since Eghbal's magisterial study of OSS, developers have become increasingly vocal about funding. Researchers are also beginning to look more closely at the individual contributors whose work underpins today's OSS ecosystem. These efforts have started to shed light on the complex symbiosis—or perhaps commensalism—between community-developed OSS and corporate-backed OSS.

Among other companies, Netflix, JP Morgan, and Airbnb have reaped significant benefits from company-sponsored community-maintained open source, not only in terms of demonstrating technical prowess and cultivating talent, but also in terms of operational impact. Other groups, like the world's largest automakers collaborating on Automotive Grade Linux or the financial sector companies embracing the Hyperledger project, seem to be following suit by forming consortia. GitLab's effort to establish a clear set of principles that enable a diverse OSS contributor community to work as one is another compelling case in point. The company's management promises not to "remove features from the open source codebase in order to make the same feature paid." GitLab also stresses contributors' right to the integrity of their work: "If the wider community contributes a new feature they get to choose if it is open source or source-available (proprietary and paid)" [9]. By explicitly recognizing the value volunteer developers bring to the platform, the company has been able to promote high-quality code contributions while avoiding cannibalization.

GitLab's rivals also appear to be taking a long-term view of OSS risk [10]. In February 2019, Microsoft took a snapshot of the top active public GitHub repositories, depositing physical copies of some of the world's most widely used software in a decommissioned coal mine in the Svalbard archipelago of Norway. The company has already stored copies of the source code for the Linux and Android operating systems in this remote region, along with 6,000 other OSS libraries it considers significant. Part gene bank and part library, this mega-repository is now the largest tenant in the Arctic World Archive, with additional redundancies planned for other locations. Backing up this treasure trove of software is a significant resilience and data loss prevention measure. However, building a nest for Coase's Penguin [11] in Svalbard is by no means sufficient for the vitality of the open source economy. On the contrary, as OSS becomes ever more ubiquitous, active maintenance becomes an increasingly pressing priority. Which brings us to the maintainers.

OSS Maintenance

Although there is "a high correlation between being employed and being a top contributor to" OSS [12], sustaining it takes more than a regular income stream. Long-term commitment to open source stewardship is also essential, as is budgeting time for periodic upkeep. For perspective, consider that 36% of professional developers report never contributing to open source projects, with another 28% reporting less than one open source contribution per year (2019 Stack Overflow Developer Survey). Thus, despite more direct enterprise engagement with open source, risk-averse attitudes towards licensing risk and potential loss of proprietary advantage endure by and large. Consider further Table 1, which shows how concentrated contribution patterns are, particularly in JavaScript, and thus where additional OSS maintenance support could have an outsized impact.

For additional context, Figures 1 and 2 show the geographic and technological mix of contemporary OSS development worldwide. Note that this is not an exhaustive account of OSS growth, merely an indicative snapshot at a single point in time. In addition, keep in mind that this data, sourced from the Open Source Compass, excludes GitHub projects with fewer than 10 watchers. For more detail on these smaller open source projects, which are enjoying intense growth outside of the US, see the *State of the Octoverse* report mentioned in the caption of Table 1.

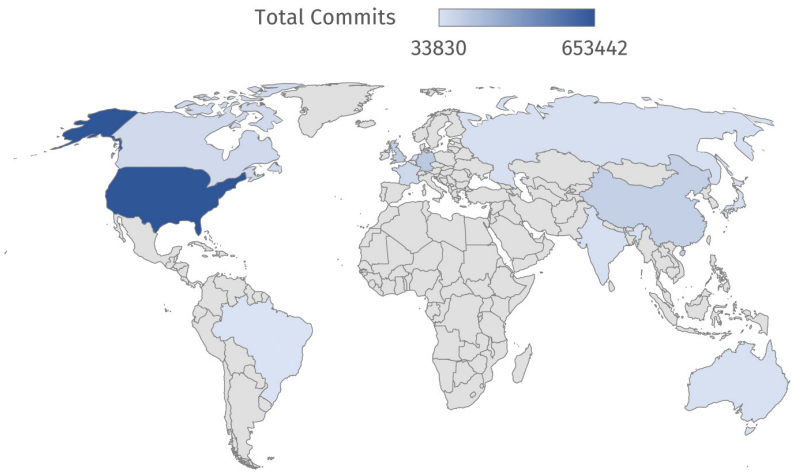

Figure 1: Geographic mix of OSS contributors on GitHub, 1Q19. Source: Open Source Compass (https://opensourcecompass.io/locations); note that this map excludes countries with fewer than 5,000 commits.

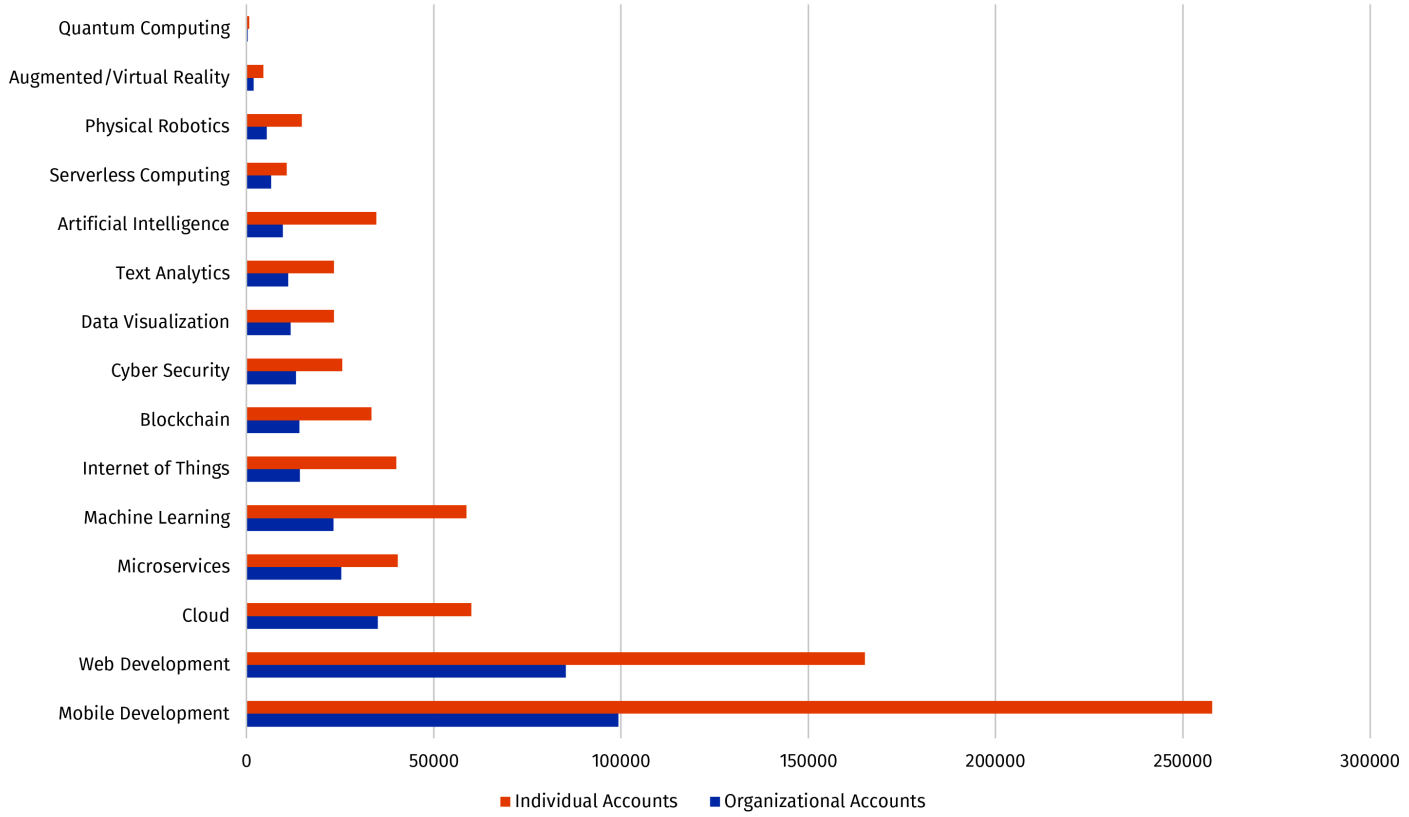

Figure 2: Technological mix of GitHub contributions, 1Q19. Source: Open Source Compass (https://opensourcecompass.io/domains/#which-domains-have-the-most-contributors), which uses data from the GH Torrent project, a research initiative led by Georgios Gousios of Delft University of Technology. GH Torrent monitors the GitHub public event timeline and retrieves and stores the contents and dependencies of each event.

Conclusion

Each year, the Augean task of patching OSS vulnerabilities falls to small groups of solitary maintainers who generally rise to the occasion but who also have to balance competing commitments. This developer dynamic has unfortunate security ramifications for widely used software like bash, OpenSSL, and Apache Struts, the latter of which played a significant role in the Equifax breach. In parallel, bitsquatting and typosquatting (e.g., the `python3-dateutil` library masquerading as the popular `dateutil` tool) as well as developer infrastructure exploits (such as the `event-stream` hack) are opening up new attack vectors that undermine trust in OSS. In addition, with "rage-quit" takedowns (like the npm `left-pad` deletion [13], which briefly impacted React and Babel) and with maintainer withdrawal on libraries like `core-js` and `jsrsasign`, enterprise risk managers are increasingly attuned to the risk of broken builds. Given these challenges, federated package registries, cryptographically signed software packages, and reproducible builds are all steps in the right direction.

In the long run, however, establishing a *modus vivendi* between IT risk managers and open source developers will be critical to open source innovation, security, and competitiveness. Such an outcome will be as much a function of cultural adjustment as of technological advancement. Organizations paying the open source piper need to remain attuned to developer trust and transparency issues, and while there are few easy answers for how to sustain and secure OSS, paying it forward on maintenance is likely to generate outsized benefits, not only for end users, but also for society at large.

References

[1] P. Ford, "What Is Code?" *Bloomberg Businessweek*, June 11, 2015 (describing this secular shift in detail from the perspective of non-technical company managers): https://www.bloomberg.com/graphics/2015-paul-ford-what-is-code/.

[2] In early 2020, the list includes media companies (Disney, CBS), insurers (State Farm, Liberty Mutual, and Northwestern Mutual), asset managers (JP Morgan, Goldman Sachs, BNY Mellon, BlackRock), industrial firms (3M, GE, and Emerson Electric), energy giants (Halliburton, DCP Midstream, and NRG), retailers (Walmart, Nordstrom, and Home Depot), airlines (Alaska Air and American Airlines), tractor OEMs (John Deere and AGCO), and automakers (Tesla and Ford), among others.

Another ≈20% of the Fortune 500 appears to have OSS placeholder pages for brand integrity and/or developer recruiting purposes.

[3] See generally T. Wang, "Knowledge Spillovers in the Open Source Community," Toulouse Digital Seminar, 2017; see also J. Meinwald, "Why Two Sigma Contributes to Open Source" (January 29, 2018): https://www.youtube.com/watch?v=5lk7LJU_zZM.

[4] F. Nagle, J. Wilkerson, J. Dana, and J. L. Hoffman, "Vulnerabilities in the Core Preliminary Report and Census II of Open Source Software," The Linux Foundation & The Laboratory for Innovation Science at Harvard, February 18, 2020: https://www.coreinfrastructure.org/wp-content/uploads/sites/6/2020/02/census_ii_vulnerabilities_in_the_core.pdf.

[5] G. Haff, *How Open Source Ate Software* (Apress, 2018), p. 172.

[6] C. Aniszczyk, "Open Source Gerrymandering," Oct. 8, 2019: https://www.aniszczyk.org/2019/10/08/open-source-gerrymandering/; B. Scott, "The Hacker Hacked," *Aeon*, August 10, 2015: https://aeon.co/essays/how-yuppies-hacked-the-original-hacker-ethos.

[7] See generally Y. Benkler, "Coase's Penguin, or, Linux and *The Nature of the Firm*," *Yale Law Journal*, vol. 112, no. 3 (December 2002), pp. 369–446 : https://www.yalelawjournal.org/article/coases-penguin-or-linux-and-the-nature-of-the-firm.

[8] N. Eghbal, *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure* (Ford Foundation, 2016): https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/.

[9] "Our Stewardship of GitLab": https://about.gitlab.com/company/stewardship/.

[10] A. Vance, "Open Source Code Will Survive the Apocalypse in an Arctic Cave," *Bloomberg Businessweek*, November 13, 2019: https://www.bloomberg.com/news/features/2019-11-13/microsoft-apocalypse-proofs-open-source-code-in-an-arctic-cave.

[11] See [7], citing Y. Benkler, 2002. As Benkler notes, "the geek culture that easily recognizes 'Coase' doesn't [always] recognize the 'Penguin,' and vice versa. 'Coase' refers to Ronald Coase, who originated the transactions costs theory of the firm that provides the methodological template for the positive analysis of peer production...The penguin refers to the fact that the Linux kernel development community has adopted the image of a paunchy penguin as its mascot/trademark. One result of this cross-cultural conversation is that [discussions of open source require one to] explain in some detail concepts that are well known in one community but not in the other."

[12] See [4], citing, Nagle et al., 2020.

[13] D. Haney, "NPM & left-pad: Have We Forgotten How to Program?" (March 23, 2016): https://www.davidhaney.io/npm-left-pad-have-we-forgotten-how-to-program/.