Journal of Politics and Ethics in New Technologies and AI

Vol 1, No 1 (2022)

Journal of Politics and Ethics in New Technologies and AI

Free and Open-Source Software: Freedom, Transparency and Efficiency in the Digitalization Era

Emmanouil Prokakis

doi: 10.12681/jpentai.31230

Journal of Politics and Ethics in New Technologies and AI
Volume 1, Issue 1 (2022)
e-ISSN: 2944-9243

https://doi.org/10.12681/jpentai.31230

COMMENTARY

Free and Open-Source Software: Freedom, Transparency and Efficiency in the Digitalization Era

Emmanouil Prokakis
Department of International and European Studies, University of Piraeus, Greece.

Abstract

Free and Open-Source Software (FOSS) has grown substantially over the past decades, becoming an essential part of the modern world. FOSS refers to software that preserves the user's freedom to read the program's source code, make modifications and redistribute it. Since the source code is open and accessible to everyone, the user is not only permitted, but also encouraged to make modifications to the software to fill specific needs. Corporations utilize that capability to get access to specifically tailored software for their needs, while proprietary software fails to meet that promise due to providing a standardized package of the software, adopting a one-size-fits-all approach. Therefore, FOSS allows for a more efficient fulfillment of software needs. Additionally, FOSS can play an important role in the protection of human rights in the digital realm, particularly in the case of the right to privacy. Because of its transparent nature, FOSS is more respectful towards the user's privacy, compared to proprietary software, since the latter restricts the user from knowing the program's under the hood functionality. With closed source software, arises a problem of asymmetric information between the software company and the consumer. FOSS offers a solution to this information asymmetry, by allowing users to obtain information about the inner workings of the program they rely upon. Through FOSS, the imbalance of power between the developer and the user disintegrates.

Keywords: Free and Open-Source Software, Proprietary software, Digital rights, Right to privacy, Software needs, Information asymmetry, Transparency

Introduction

Free and Open-Source Software (FOSS) has blossomed over the past decades, securing its place as an integral part of the modern world. From the Apache web server to the Mozilla Firefox internet browser and the GNU/Linux Operating System, FOSS seems to be everywhere we look. Even in space exploration, FOSS acts as a companion to the human race, seeing the International Space Station switching to Linux in 2013 and NASA utilizing an open-source operating system and flight software framework for the Ingenuity helicopter of the Perseverance rover in 2021 (NASA, 2020; Hill, 2021).

Free and Open-Source Software?

Free and Open-Source Software refers to software that preserves the freedom of the user to view, run, alter and redistribute the software as they see fit. Contrary to closed source / proprietary software, users

Prokakis, E. (2022). Free and Open-Source Software: Freedom, Transparency and Efficiency in the Digitalization Era. *Journal of Politics and Ethics in New Technologies and AI, 1*(1), e31230. https://doi.org/10.12681/jpentai.31230

have access to the source code and are able to study the inner workings of the program, modify it to better suit their needs and share their modifications with the community.

Free Software differs from Open-Source Software, although they are often presented as one. Free Software, according to the Free Software Foundation (FSF), means that the program must respect the right of the user to run the program for any purpose, the right to read and modify the program's source code, the right to redistribute copies of the program and the right to distribute modified versions of the program. When we discuss Free Software, it's crucial to understand that 'free' doesn't refer to software that's free of charge (gratis software), but to software that ensures the freedom of the user (libre software). Since unfree software is a social problem for the FSF, the role of Free Software is essential for guaranteeing the freedom of society, through the promotion of social solidarity and freedom in the modern Digitalization Era. In practice, Free Software embodies a set of social and ethical values, which differentiates it from Open-Source Software.

According to Richard Stallman (2009), father the of Free Software Movement, Open Source doesn't incorporate the social values of Free Software. Open source aims for a more practical, non-ethical and non-prescriptive approach, that focuses on the development of programs by a community, making the software more reliable by providing universal access to the source code. While Free Software focuses more on the protection of the user's freedom and rights, Open Source tends to be more interested in developing powerful and reliable applications, which is achieved through the 'openness' of the source code. In spite of the ideological differences among the two groups, Free Software and Open Source are closely related in practice.

As mentioned above, both Free Software and Open Source require the source code to be open, modifiable and redistributable. To ensure those freedoms, Stallman came up with the concept of Copyleft licenses, that license the software but also protect the user's freedom to read the code, make modifications and redistribute the software (Stallman, 1999). Contrary to traditional Copyright licenses, the Copyleft licenses, as for example the GNU General Public License, ensure the user's aforementioned freedoms and demand that any new fork or modified version of the software automatically adopts the licensing principles of its parent software (Schweik, 2007). The Copyleft licenses provide a solution for developers that look to license their software without choosing between the two extremes of all rights reserved and no copyright at all. Thus, the Copyleft licenses retain the freedoms of the user, while also keeping the source code outside of the public domain, i.e. subject to no copyright at all.

FOSS and the addressing of software needs

The development model of FOSS applications brings a lot of benefits to enterprises and households alike, allowing the combined programming knowledge, creativity and expertise of a decentralized internet community to shine (Pappas-Johnson, 2001). The development process is based around thousands of volunteer programmers co-developing applications through the internet, suggesting improvements, bug fixes and new features. Of course, this doesn't mean that the entire FOSS ecosystem is built on altruism - in the absence of economic motives. Large companies, like Microsoft and Google, invest heavily on Open-Source Projects. IBM recently acquired Red Hat - one of the biggest companies that built a successful business model around Open Source during the 1990s - for $34 billion dollars in 2019. Enterprises show interest in Open Source because it provides the means for obtaining customized software that better meets in-house needs (Bessen, 2002). Open Source, compared to proprietary software, has an advantage in satisfying the user's needs due to the customization it offers. Since the source code is open, the user is not only permitted, but also encouraged to make modifications to the software to fill specific needs and share it. Large companies utilize that capability to get access to specifically tailored software for their needs.

Proprietary software fails to meet that promise by providing a standardized package of the software, adopting a one-size-fits-all approach. As a result, many features go unused, which unfortunately makes applications feel "bloated", and at the same time missing important features that are specific to the needs of each user. Additionally, as Bessen (2002) puts it "the cost of debugging ultimately limits the ability of standardized products to meet highly disparate needs", because the debugging cost is positively correlated with more complex and feature-rich programs. The Open-Source development model makes it possible for enterprises to get the source code from an existing project and tailor it to meet their specific requirements, allowing for a more efficient fulfillment of their software needs. The efficiency of Open Source explains how an application written and maintained by volunteers, that, in practice, functions as a public good, can compete directly with software developed by large corporations.

Businesses might also prefer relying on Open-Source elements for strategic reasons, since being dependent on commercial / closed source software can be identified as a vulnerability for the interests of the company. If the software provider were to shut down or dropped support for the software, the company would have no choice but to assess damages and restore its operation through new expenses and product licenses. The bug fixing process would also be entirely resting on the software provider.

As a result, the company would be dependent on an external actor for ensuring its operational status and protection from bugs and software vulnerabilities.

Privacy and Security could by their own be a reason for enterprises to prefer relying on Open-Source programs than their proprietary counterparts. When a program is closed source, the user can't be sure about what actions the program takes when executed, since it's not uncommon for programs to carry backdoors and collect user information (Stallman, 2002). Even if a program doesn't incorporate malicious behaviour, bugs in code could leave the user/company susceptible to cyber-attacks. FOSS offers a solution to both of these problems, seeing that the code is open and modifiable, malicious elements wouldn't be accepted into the program by the community. At the same time, more eyes looking at the code, combined with user feedback, means that bugs and software vulnerabilities would be easier to locate and patch (Raymond, 1999). In the theoretical level, FOSS should be more secure than proprietary software because vulnerabilities should be easier to pinpoint and fix. However, since the source code is also accessible to malicious users, there is room for found vulnerabilities not being reported or patched. Singh et al. (2013) conclude that "*… open-source software is not automatically more or less secure than proprietary software. Both development approaches have their strengths and weaknesses, but neither automatically produces more secure code than the other*". Of course, the debate about the security of FOSS is far from over.

FOSS, Transparency and Digital Rights

In June of 2020, the president of the United Nations, Antonio Guterres, presented the Secretary-General's Roadmap for Digital Cooperation, that aims to get all people connected, respected, and protected in the digital age. Since human rights apply both online and offline, the UN recognizes the protection of human rights in the digital sphere as an important issue to be addressed, underlining the fact that the constantly evolving digital technologies can also be used for surveillance and censorship. The mass collection and processing of personal data can a be a threat to the right to privacy, therefore, arises the need for transparency when an individual's data are used by large corporations and governments. The General Data Protection Regulation (GDPR) is a privacy and security law passed by the EU, amounting to a holistic approach for ensuring the privacy and data protection of EU citizens. GDPR applies since May of 2018 and stipulates large penalties for companies that aren't GDPR – compliant. The United Nations promote the establishment of internationally agreed standards on data protection and the protection of the right to privacy. The UN approach, however, appears to be lacking in comparison to the EU, notably in the case of developing countries. The EU approach has been successful in the promotion of privacy and data protection, setting the international standard for the protection of human rights in the digital world. An approach similar to the EU's regulatory framework becomes increasingly necessary on a global scale, chiefly when it comes to private companies, that collect personal data for commercial reasons.

FOSS can play an important role in the protection of human rights in the digital realm, particularly in the case of the right to privacy. Because of its transparent nature, FOSS is more respectful of the user's privacy, compared to proprietary software. The source code is open and accessible, and even if it's not understood by everyone, a community of programmers studies the code on a daily basis, suggesting improvements and patching bugs. Since the source code is open and the developers of the software are at the same time users of it (Von Hippel, 2001), the incorporation of telemetry wouldn't be embraced neither by the community nor by the developers. Additionally, supposing that someone contributed code that introduced malicious elements to the program, that code would not be accepted by the moderators of the project, let alone the community. Even if a decision was made to encompass user data collection and potentially unwanted behaviour, the creation of a fork of the original project, one that lacks those characteristics, would have been quick to emerge, owing to the openness of the source code. This type of transparency allows users to place their confidence in FOSS programs, in contrast to closed source commercial software, where the user is restricted from knowing the program's under the hood functionality. FOSS endorses the privacy by design concept, where nothing is a secret to the user.

Regarding closed source software, arises a problem of asymmetric information between the software company (developer) and the consumer (average user). The former has better information on the software product, thus creating an imbalance of power between the two, with the latter being at a disadvantage. The user is uncertain about the behaviour of the software he relies upon, since he has no way of knowing the actions that the program takes. FOSS offers a solution to this information asymmetry, by permitting users to obtain information about the inner workings of the program. Through FOSS, the imbalance of power between the developer and the user disintegrates due to oversight from the users and the broad development base.

Conclusions

This paper attempted to demonstrate how FOSS acts as the way forward for freedom, transparency and efficiency in the modern Digitalization Era. With the rise of big data and the data collection practices of big corporations maturing, FOSS performs as an isle of freedom and transparency in the modern digital technologies' ocean, providing the user with essential freedoms to obtain, modify and distribute software, while ensuring that the right to privacy remains respected. Both the average users and companies can - and do - entrust FOSS applications with critical tasks, due to their transparent nature and efficient fulfillment of software needs.

References

Bessen, J. (2002). What Good Is Free Software. In Hahn, R. (ed) (2002). *Government Policy towards Open Source Software.* Washington, DC: AEI Brookings Joint Center for Regulatory Studies

Colford, S. (2009). Explaining Free And Open Source Software. *Bulletin of the American Society for Information Science and Technology* – December/January 2009 – Volume 35, Number 2

EU (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

IBM (2019, July 9). IBM Closes Landmark Acquisition of Red Hat for $34 Billion; Defines Open, Hybrid Cloud Future. Accessed on August 5, 2021 from https://newsroom.ibm.com/2019-07-09-IBM-Closes-Landmark-Acquisition-of-Red-Hat-for-34-Billion-Defines-Open-Hybrid-Cloud-Future

EPAM Systems (2021). Open Source Contributor Index. Accessed on August 4, 2021 from https://opensourceindex.io/

Hill, T. (2021). *Meet the Open-Source Software Powering NASA's Ingenuity Mars Helicopter*. NASA Science Mars Exploration Program. Accessed on August 20, 2021 fromhttps://mars.nasa.gov/news/8982/meet-the-open-source-software-powering-nasas-ingenuity-mars-helicopter/

Linux Foundation (2013). Linux Foundation Training Prepares the International Space Station for Linux Migration, https://training.linuxfoundation.org/solutions/corporate-solutions/success-stories/linux-foundation-training-prepares-the-international-space-station-for-linux-migration/

NASA, (2020). F' Flight Software & Embedded Systems Framework. Accessed on August 20, 2021 from https://nasa.github.io/fprime/

Pappas-Johnson, J. (2001). Economics of Open Source Software. Accessed on August 4, 2021 from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.588.1436&rep=rep1&type=pdf

Raymond, E. (1999). *The Cathedral and the Bazaar - Musings on Linux and Open Source by an Accidental Revolutionary.* CA, USA: O’Reilly & Associates

Schweik, C. (2007). Free/Open-Source Software as a Framework for Establishing Commons in Science. In Hess C., & Ostrom E. (2007), *Understanding Knowledge as a Commons From Theory to Practice.* Cambridge, Massachusetts: Massachusetts Institute of Technology Press

Singh, A., Mittal, P. & Jha, N. (2013). FOSS: A Challenge to Proprietary Software. *International Journal of Computer Science and Technology*, Vol. 4 Issue 3, July-Sept 2013

Stallman, R. (2002). Free Software, Freedom and Cooperation. In Gay, J. (ed) (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman.* CreateSpace Independent Publishing Platform

Stallman, R. (2009). Why "Open Source" Misses the Point Of Free Software. *Communications of the ACM* Volume 52 Issue 6 pp 31–33. https://doi.org/10.1145/1516046.1516058

Stephen, B. (2019, February 19). Linux has made it to Mars. Accessed on August 3, 2021 from https://www.theverge.com/2021/2/19/22291324/linux-perseverance-mars-curiosity-ingenuity

UN. (2020). Report of the Secretary-General Roadmap for Digital Cooperation.

Von Hippel, E. (2001). Innovation by User Communities: Learning from Open Source Software. *MIT Sloan Management Review*, 42(4), 82-86.