Running Head: HOW OPEN-SOURCE EFFECTS CS

How The Open-Source Model Has Effected the Computer Science Industry

Aidan Hopper

hoppera@cwu.edu

Central Washington University

**Abstract**

The open-source development model differs greatly from the traditional closed-source model. Anyone can contribute, view, and use the code. This is why open-source software is considered a public good, because of its free-use nature. This style of innovation benefits the public without the profit incentive that a firm typically uses as leverage to produce innovation. Many licenses enable project leaders to give the rights to contribute, view and use the code. These rights are the foundation of open source development, without them open-source software would have no legal protection. Popular open-source software has also been found to have very quick rates of development.

Keywords: open-source; closed-source; proprietary; software; development models.

**Table of Contents**

[Abstract 2](#1%19Abstract%19C)

[Introduction 5](#1%19Introduction%19C)

[Narrative 6](#1%19Narrative%19C)

[Motivations to contribute 6](#1%19Motivations_to_contribute%19C)

[Collective innovation 7](#1%19Collective_innovation%19C)

[Licenses 8](#1%19Licenses%19C)

[Development speed 8](#1%19Development_speed%19C)

[Open-source versus closed-source adoption 10](#1%19Open-source_versus_closed-source_adoption%19C)

[Conclusion 10](#1%19Conclusion%19C)

[References 11](#1%19References%19C)

Table of Figures

[Figure 1: Compares closed-source and open-source development models. 6](#Figure!0|sequence)

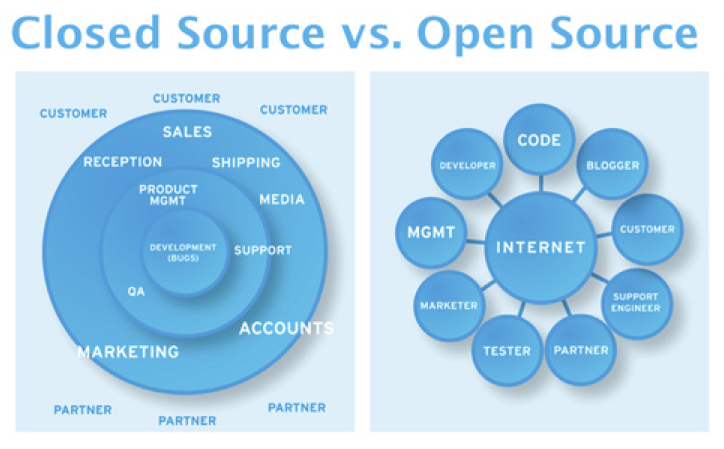
[Figure 2: Intervals between major Linux releases. 10](#Figure!1|sequence)

How The Open-Source Model Has Affected the Computer Science Industry

**Introduction**

The open-source software development model is defined as the ability for anyone to inspect, use, and contribute to the codebase. This stands in contrast to how the software would otherwise be produced, defined as closed-source. Software is closed-source when the people who can view, use, and contribute to the codebase are limited by the owner of the software. Examples of software developed under the open-source and close source models are the Linux kernel and Microsoft Windows respectively.

Open-source software is widely described as a public good—a good that benefits the general welfare of people—because of the free-to-use nature of the model. Big-name examples of open source software include Linux, Blender, Firefox, and Python. Many use cases of open source software is being used as a building block for higher-level software. A good example of this is Python, the programming language. Python is used to build software open and closed-source but most who utilize these projects—other than programmers—will never interact with Python directly. The users will interact with it indirectly through their software usage.

Figure 1: Compares closed-source and open-source development models.

Source: What Is a Decentralized Application? - Decentralized Applications [Book]. (n.d.).

**Narrative**

**Motivations to contribute**

open source innovation seems almost impossible at first glance. The lack of a profit incentive makes it hard to believe that any type of innovation could happen under an open-source model at all. But the fact that there are multiple hugely successful open source projects—Python and Linux to name two—shows the development model must have some ground to it. The open-source development model works because people actively decide to freely contribute to projects.

The motivations behind contributions vary widely, from just enjoying the thrill of contributing to a project, to reputation building, to making improvements to the software that's needed for personal use cases (von Krogh & von Hippel, 2006). The fundamental lack of a profit motive and the fact that developers are also often users in open source projects aligns the incentives of the developers with the users more than closed-source development.

Some tasks must be performed that do not fit these motivations well, described as mundane but necessary. One such task is providing support to the less sophisticated user—also known as technical support—which is not covered well in motivations to contribute. This raises the question, what motivates people to answer questions about open source projects on forums and other mediums? According to a study on the Apache web server help forums, most provide support because they were provided support in the past (Israeli & Feitelson, 2010). But many questions still remained unanswered for months if answered at all. Robust technical support can be considered one of the weaknesses of the open-source development model.

**Collective innovation**

open-source software is considered a derivative form of collective innovation. An example of an innovation that was subject to collective development was when the iron-making industry in Britain's Cleveland district shared their bast furnace designs to direct competitors to innovate on their designs in the second half of the 20th century (Osterloh & Rota, 2007). The open-source model challenges the traditional innovation model of private innovation. The collective innovation of free software has created a robust library of tools and platforms. This allows for unrestricted access to sophisticated tools like operating systems, programming languages, video editors, web browsers, and much more.

**Licenses**

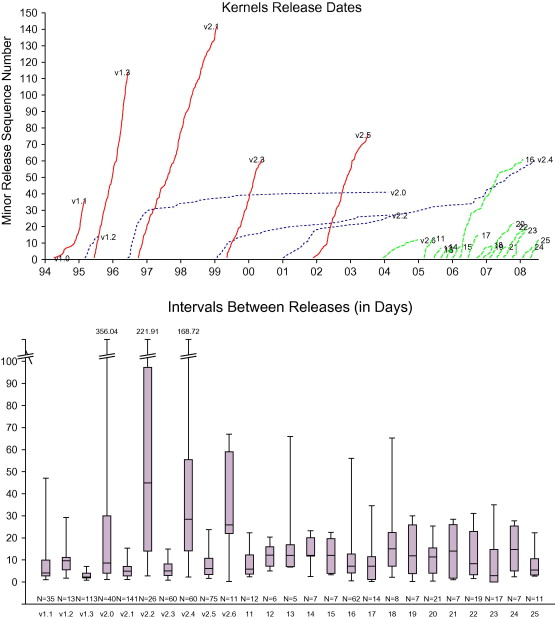
Open source software uses open-source licenses to give users the rights to view, develop, and use the software. Some licenses include the GNU General Public License (GPL), described as copyleft—in contrast to copyright where intellectual property laws are used to generate revenue—guarantees that all software derived from the original remains free to use and open source (The GNU General Public License v3.0, 2007). Because of the GPLs fine print, the license tends to disincentivize any commercial use cases.

Another widely used open source license is the GNU Lesser General Public License (LGPL) which lacks the GPLs copyleft characteristic. An example of software that uses the LGPL is the Linux kernel, which is why Linux can be commercially used. The *Lesser* portion of the LGPL means that software built on top of the original software is not guaranteed to be free or open-source, opening up the software to commercial use cases. Licenses are the foundation of open source development, without them, there would not be a way for project leaders to release their code with legal protections (The GNU Lesser General Public License, 2007).

**Development speed**

one of the benefits of the open-source development model is development speed. The speed comes from the extremely low overhead costs to development (Dalle & Jullien, 2003). One framework to measure development speedis Lehman's laws of software evolution. There are three distinguished types of software, S-type, P-type, and E-type, with an S program written to a specification, a P-type being written to implement a procedure, and E-type which is used to perform a real-world activity. E programs are described as infrastructure. E-type programs are meant to evolve with their surroundings so it is the best type to analyze for development speed.

Lehman outlined eight laws to analyze software evolution: continuing change, increasing complexity, self-regulation, conservation of organizational stability, conservation of familiarity, continuing growth, declining quality, and feedback systems (Lehman, 1980).

Figure 2: Intervals between major Linux releases.

Source: Israeli & Feitelson, 2010.

Figure 2 is best analyzed through the lens of continual growth, showing that at the time in 2010 Linux had been steadily growing for multiple years. The growth rate shows that popular open-source software develops at rapid rates due to the volume of contributors and small overhead.

**Open-source versus closed-source adoption**

*T*here are different aspects of the software that a user can value which dictates the type of software should be adopted: open or closed-source. Between industries, the common factor that the open-source model provides is security, due to the number of contributors viewing the code every day, and the cost-efficiency. For closed-source models, the common factor is compatibility with other commercial software (Sacks, 2015).

**Conclusion**

The open-source development model provides an alternative to the typical closed-source development model used by most major institutions. Whether the software is adopted depends on the user's use-case, but it does create more competition in the software world. The model is built on top of the idea of licensing, which allows project leaders to release their code with legal protections. The open-source development model allows for high amounts of transparency because everything is viewable, and creates an environment where developer interests are intrinsically aligned with user interests.

**References**

Dalle, J.-M., & Jullien, N. (2003). “Libre” software: turning fads into institutions?. *Research Policy*, *32*(1), 1–11. https://doi.org/10.1016/s0048-7333(02)00003-3

Israeli, A., & Feitelson, D. G. (2010). The Linux kernel as a case study in software evolution. *Journal of Systems and Software*, *83*(3), 485–501. https://doi.org/10.1016/j.jss.2009.09.042

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: “free” user-to-user assistance. *Research Policy*, *32*(6), 923–943. https://doi.org/10.1016/s0048-7333(02)00095-1

Lehman, M. M. (1980). Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*, *68*(9), 1060–1076. https://doi.org/10.1109/proc.1980.11805

Osterloh, M., & Rota, S. (2007). Open source software development—Just another case of collective invention? *Research Policy*, *36*(2), 157–171. <https://doi.org/10.1016/j.respol.2006.10.004>

*Sacks, M. (2015). Competition Between Open Source and Proprietary Software: Strategies for Survival. Journal of Management Information Systems, 32(3), 268–295. https://doi.org/10.1080/07421222.2015.1099391*

*The GNU General Public License v3.0 - GNU Project - Free Software Foundation*. (2007). Gnu.org. https://www.gnu.org/licenses/gpl-3.0.en.html

*The GNU Lesser General Public License - GNU Project - Free Software Foundation*. (2007). Gnu.org. https://www.gnu.org/licenses/lgpl-3.0.en.html

von Krogh, G., & von Hippel, E. (2006). *Management Science*, *52*(7), 975–983. http://www.jstor.org/stable/20110574

*What Is a Decentralized Application? - Decentralized Applications [Book]*. (n.d.). www.oreilly.com. https://www.oreilly.com/library/view/decentralized- applications/9781491924532/ch01.html

‌

‌