**The Open Source Models**

**Affect on The Computer Science Industry**

Aidan Hopper

hoppera@cwu.edu

02/28/2022

Table of Contents

[Abstract 3](#1%19Abstract%19C)

[Introduction 4](#1%19Introduction%19C)

[Narrative 4](#1%19Narrative%19C)

[Motivations to contribute 4](#1%19Motivations_to_contribute%19C)

[Collective innovation 5](#1%19Collective_innovation%19C)

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

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

[References 7](#1%19References%19C)

**A****bstract**

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 needs to produce innovation. Open source software is also more aligned with their users because development isn’t motivated by profit. Because of this software developed in an open source environment lacks the promotion of toxic additions to software that closed source development incentivizes.

**Introduction**

The open source software development model is defined as the ability for anyone inspect, use, and contribute to the codebase. This stands in contrast to how software would otherwise be produced, defined as closed source. Software is close sourced when the people who can view, use and contribute to the codebase is 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 the people—because of the free to use nature of the model. Big name examples of open source software include the GNU/Linux operating system, Blender, Firefox, and Python. Many use cases of open source software are under the hood of other software. The best example of this is Python, the programming language. Python is used to build a multitude of projects open and close source; and while most users other than programmers will never interact with Python directly, users will still interact with it indirectly through their software usage.

**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 hugely successful open source projects—Python and Linux to name two—means that it has to have some merit. The open source development model works because people actively decide to contribute to projects. The motivations behind their contributions vary widely, from just enjoying the thrill of contributing to a project, to repuatation building, to making improvements to the software that they need for their own use cases (von Krogh, 2006). The fundamental lack of a profit motive in open source projects aligns the incentives of the developers more with the users. Because developers are often users, and that the code is freely viewable, the model promotes a less toxic development than that of their close sourced profit driven counterparts. This includes a general lack of embedded advertisements, paywalls, data farming, and other potentially toxic additions that closed source profit driven development incentivizes.

*Collective innovation*: Open source software is considered another form of collective innovation, which has been seen in the distant past inside other industries before computer science even existed. An example of an innovation that was subject to collective development was the steam engine after 1800 (Osterloh, 2007). The open source model challenges the traditional innovation model of private innovation.

*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—guarantees that all software derived from the original remains free to use (The GNU General Public License v3.0, 2007). This makes the GPL the ultimate public-good open source license. Because of the GPLs fine print the license tends to disincentivize any commerical use cases.

**C****onclusion**

The open source model has had many impacts on Computer Science. The most notable of these impacts being the public-good nature of open source software. The ability for anyone to view, contribute, and use the code of open source projects is invaluable. Another notable impact is the ability to develop software without a profit motive. This aligns the incentives of the developers with the users of the software.

**References**

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>

*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

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

‌