“Put your money where your methodology is”

A research paper into the world of open source

St. Cloud State University

CSCI 332: Computing Ethics

By: Andrew Morris

Table of Contents

[Introduction 3](#_Toc449493599)

[Definitions and distinctions 3](#_Toc449493600)

[Motivations 4](#_Toc449493601)

[Origins and history 4](#_Toc449493602)

[Social context 4](#_Toc449493603)

[Open-Source in an ethical framework 5](#_Toc449493604)

[Professional & ethical responsibilities 5](#_Toc449493605)

[Risks 6](#_Toc449493606)

[Privacy & civil liberties 6](#_Toc449493607)

[Intellectual property 6](#_Toc449493608)

[Economic issues 7](#_Toc449493609)

[Open source alternatives to proprietary media standards and formats 7](#_Toc449493610)

[Open source news 8](#_Toc449493611)

[Summary: My two cents 8](#_Toc449493612)

[REFERENCES: 9](#_Toc449493613)

# Introduction

“Linux”, “GTK”, and “EMACS”, to some these words may not seem like much. However, to the software community, these are highly functional tools that are used every day. They stem from the phenomena known as Open Source. Unlike traditional software that is developed by a company then sold as a black box for profit. Open source software is developed and maintained through a community, and when the software is available, so is the source code. This community, free of cost, donates their time into maintaining these open source projects. But why? In this paper I will address the following matters involving Open Source Software. First, we will dig into the definitions and distinctions within the jargon of open source. Secondly, explore the motivations as to why open source projects are successful and why people donate their time to them. Thirdly, some history on Open Source projects, and some of the original projects that began this phenomenon. Then, the social context of Open Source software will be examined, and how Open Source projects parallel some social networks. Then, we will discuss how open source can facilitate and improve processes of deliberation. Following that, we will investigate professional and ethical responsibilities of the Open Source community. After that, we will discuss the benefits of multiple contributors to Open Source Projects in the context of security and privacy. Then the strategies of Open Source software, and the unique benefits it’s development process provides in fighting computer crime. After that, a discussion on intellectual property, and the several types of licenses available to the open source community. Then an extrapolation on economic issues associated with Open Source Software. A discussion on Open Source options available for media standard, compression, etc.…. Then finally, a discussion on current ‘hot button’ issues in the Open Source community currently.

# Definitions and distinctions

For a conversation to be meaningful, some background knowledge is critical. So, we ask the question, “What is Open Source?”. Eric S. Raymond in his foreword in “*The Cathedral and The Bazaar”* defines Open Source as, “the process of systematically harnessing open development and decentralized peer review to lower costs and improve software quality. “(p. xi). Raymond, being an avid contributor to Open Source obviously has a romantic view on the open source community. This perhaps is a great example of the dedication of contributors in the community, and the passion that Open Source contributors have for Open Source Software. Perhaps a more neutral definition then Raymond would be from Ojhi and Roa in their essay about the emergence of Open Source Software in the business world. “The open source movement has evolved from its original objective of free sharing of the source code of software products, to becoming an alternative approach to software development that is non-proprietary in nature, developed through a peer network, supported by the Internet, and with little or no monetary benefits for the software developers or reviewers who contribute to the efforts.”. (p. 127) Ojhi and Roa point out that Open Source software is not simply a community who shares code, and provides “free software”. (In fact, Stallman stated that the term ‘free’ refers to the ability to make changes, not the price you pay to use it. (Bretthauer p. 4)) It is a community which offers an alternative development approach to the traditional Software Development model. This emergence of an effective software development model is getting the attention of “researchers and commercial companies alike”. (Kishida & Ye p.1) The participants in these communities are called ‘hackers’. The term ‘hacker’ is a general term for contributors to open source projects.

# Motivations

It is easy to see the benefits of open source for those of us that have used tools such as Firefox, Notepad++, or Linux for free. But, there is a simple question to be asked of the software producing juggernaut that is Open Source. Why? Why do thousands of developers every day give their time and effort with no form of monetary payment? To answer this question, we will consult the essay “Homesteading the Noosphere” in the collection “The Cathedral and The Bazaar”. Raymond defines the culture as a gift/craftsmanship culture. (p. 110) This means that because the software one develops is ‘infinitely’ reproducible, and there is no immediate monetary gain; the value in creating the software is the reputation one receives from the community. (Raymond p. 84)

# Origins and history

The Open Source Software movement can trace it’s roots back almost fifty years. (Bretthauer p. 3) The year that MIT obtained the first PDP-1(Programmed Data Processor-1). In fact, it was the members of the MIT computing culture that first accepted the label as ‘hacker’. (Raymond p. 4) One of the forerunners of the Open Source ideology, and often referred to as a deity in the Open Source community is Richard Stallman. (Bretthaur p. 4) Stallman worked as a programmer in the Artificial Intelligence lab at MIT from 1971-1984. (Stallman) It was in these years that UNIX was invented by Ken Thompson which sparked Stallman to work on GNU in 1984. (Raymond) It was this work that lead to the community feedback approach to open source when Stallman released the transcendent editor called GNU Emacs. Stallman began selling copies of Emacs on tape, and as his sales became more common, so did the feedback he received from the users. (Bretthaur p.4) Stallman’s model was a few tight knit developers that made changes based on feedback to release to the users (though the customer’s were free to modify the software). These original ‘hackers’ did not make up the Linux type development model that we see today. Essentially, development from a large group on the internet and improved through testing by the culture. It took a Helsinki student named Linus Torvalds to provide a free kernel for Unix on 386 machines to spark this model of development. (Raymond) Now, we will explore the societal aspect of this new model that stemmed from Torvalds’ work.

# Social context

In ‘Homesteading the Noosphere’, Raymond argues that the hackers live for reputation from their peers. This social dependency implies that the social context of the Open Source community is, in fact, it’s lifeblood. The reason that the Open Source model is effective is because the hacker’s reputation in affected positively, or negatively as a result of their work. (Raymond p.92) Because, as I stated, the hacker’s value is found in the reputation from their peers. What value would the hacker find without any reputation? Or better yet, consider the benefits of the Open Source culture as it relates to feedback from the community to better the software. If we remove the societal context of Open Source software, we completely cut out these benefits.

# Open-Source in an ethical framework

**“**Utilitarianism is one of the most common approaches to making ethical decisions, especially decisions with consequences that concern large groups of people”. (A Framework for Making Ethical Decisions) The Utilitarian view in itself defines ethics based on what makes people feel happy; not happiness stemmed from personal wealth. The utilitarian view coincides well with the open source model. The hackers do not necessarily expect monetary gain from their work, but they do find happiness in contributing through the repute of their peers. Utilitarianism places value in the greater good, the exact perceived intentions of the hacker. Another context we will examine the hacker culture, will be the egotistical Duty-Based approach. (A Framework for Making Ethical Decisions)

The Duty-Based approach to ethics is defined by not what an action does, but the intent of the actor when the action is performed. This approach to ethics does not so closely coincide with the hacker culture, but perhaps in one regard. The duty-based approach is not flexible whatsoever. Ethical behavior is based on hard set rules that are to be followed at all times. The article “*A Framework for Making Ethical Decisions”* from Brown University gives the example of the “One should always speak truthfully”. This ethical framework does not allow for any flexibility, if you lie, you are not acting ethically. As I stated, perhaps there is one aspect of the hacker culture that resounds this particular value. The requirement that the source code be available to the user. Since Stallman first struggled with the closed source code of his Xerox printer, this rule is the very bedrock of Open Source Software. (Bretthaur p. 4)

# Professional & ethical responsibilities

The open source culture does not have a centralized code of ethics, rather, these ethics have evolved over time. I believe this coincides quite nicely with the lobster fishermen Ridley cites in Chapter twelve. Where if you are treading in another’s territory when you should not be, your line can be cut. In other words, if you participate in taboo behavior within a project, it is not only the owner that with shun or flame you, it is the community. (Ridley p. 229) Unlike formal codes of ethics, the ethical code for the open source culture is not explicitly stated. Rather, many norms are taught by example. (Raymond p. 104) This is seen in the process of becoming a contributor to a project. Raymond states, “Each project is a live social context of hackers that the would-be contributor has to investigate and understand socially as well as technically in order to function”. (Raymond p. 106) Perhaps this suggests that the code of ethics for an open source project, in fact, is a uniquely evolved ethical code derived from basic Open Source customs.

# Risks

The open source model is not without it’s risks. In particular, there have been cases where Open Source software has been incorporated into a sold product. These cases generally stem from the company’s inability to comply with the GPL. (Erik Andersen and Rob Landley v. Verizon Communications Inc.) So, companies that distribute proprietary software with Open Source modules take on the responsibility to comply with the GPL, or the license that is placed on the Open Source software. Which can bear legal and economic risk for a company. In a study performed by Sojer & Henkel, they found many software developers do not properly follow GPL requirements. (License risks from ad hoc reuse of code from the internet) As is concerns the Open Source developers, there can be significant risk for the developers of Open Source projects if they do not properly protect themselves through proper licensing. Without proper licensing the contributor could be held responsible for a failure of a module.

Privacy & civil liberties

Open source projects force the source code to be available to the end user. How can civil liberties and privacy be aided through these concepts? In the case of voting machines this bodes well for the decreased probability that there is malicious code in the programs. Or that a candidate is able to coerce a programmer to insert code to give advantage over another candidate. This assertion is based on a study performed by Dr. David Grimes stating the mathematical probability of a conspiracy with many people is infeasible. (Grimes) The case of cryptography is not so well defined. Because of the ability to see the source code, cryptography software, software meant to keep malicious parties from seeing their data, algorithms can easily be seen to anyone. This includes the malicious parties that wish to break the encryption. Which poses a threat to the users of the cryptographic software.

# Intellectual property

Intellectual property is a work or invention that results from some creativity that can be patented, copyrighted, or trademarked. In the open source community, there are several pre-packaged licenses to choose. Some the more popular open source licenses include: Apache License 2.0, BSD 3-Clause “new” or “revised” license, BSD 2-Clause “Simplified” or “FreeBSD” license, GNU GPL, LGPL, MIT, Mozilla Public License 2.0, Common Development and Distribution license, and Eclipse Public License. (opensource.org) For this analysis we will focus on the GNU GPL license, as it is one of the older licenses in the open source community. GNU GPL 1.0 was designed by Stallman in 1989. The GPL licenses focus on the sharing of code to the users which is referred to as “copyleft”. The FSF defines copyleft as, “Copyleft is a general method for making a program free software and requiring all modified and extended versions of the program to be free software as well”. (fsf.org) The purpose of this is to give users the freedom to modify the software to their needs. At the same time, the GPL requires that the ability to make changes to the program be passed to the users. Perhaps an interesting dichotomy, the freedom to modify, but the strict requirement to share. The copyleft obligation does closely resemble the “derivative work” definition in US copyright law defined as, “…a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications, which, as a whole, represent an original work of authorship, is a “derivative work””. (17 U.S.C.§ 101) But, unlike the derivative work, which is defined as a compilation. The GPL also requires that even works containing all, part, or even a translated part of a program be shared. Some argue that this introduces ambiguity into the license. Consider the example where a user dynamically links GPL licensed software to their program. Any further revisions to that software, which requires the ‘copyleft’ would modify the original EULA. For this reason, the LGPL license was developed. The LGPL license gives the developer the ability to link LGPL modules with proprietary programs without the proprietary program required to copyleft. As we can see in this single example. There are many ways to license Open Source software, but each license must be carefully inspected before making a decision.

The Free Software Foundation’s page for GPL 3.0 reads, “Our mission is to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software users.” (fsf.org)

# Economic issues

There are economic issues surrounding the emergence of open source as a real alternative (perhaps favorable?), to proprietary software. First we will define the different values that computer programs have. Computer programs have Use Value. Use Value is the economic value of a program as a tool. One can consider apt-get, or homebrew package managers. There is no price associated with installing them, but they are highly used as a tool. Besides Use Value, computers also have market value. This is the price people are willing to pay for a piece of software. Finally, programs have monopoly value. Monopoly value is the exclusivity value of a software. (The Economics of Open-Source Software) It has value because it is unavailable to competitors. There are many preconceptions on what would happen in a world where the open source model was the primary generator of computer programs. There are uncertainties about job security for programmers, compensation for programmers, and profitability of companies in a completely open source model. One ‘myth’ that Raymond claims to completely address is the issue of a ‘Tragedy of the Commons’ if all software was open source. Raymond argues that the overuse of a software actually increases its value rather than decrease it. (Raymond p. 123) This does bring an interesting light onto software, in general. Perhaps our current models of economics cannot quite explain this anomaly, and further insight is needed before defaulting to assume current economic behaviors such as Hardin’s tragedy.

# Open source alternatives to proprietary media standards and formats

There are many alternatives to the proprietary media formats and standards. Unfortunately, there are several superior formats that have been patented. Because of these patents, some of the best compression software is unusable by open source developers. There are, however, alternatives to many proprietary media formats in the open source community. For example, the FDK AAC is an open source library that can decode, and encode Advanced Audio Coding format. The AAC format is an interesting case where open source developers of AAC codecs can create software, but cannot distribute the source due to possible infringement from the patent owners. (Wikipedia) This renders the GPL license unusable in this case because of the inability of the developer to ‘copyleft’. A popular open source video codec is Xvid. It is based on the MPEG-4 video standard. It is licensed under GPL 2.0. (Wikipedia)

# Open source news

In recent news, a huge story has emerged about Microsoft implementing bash on Ubuntu on Windows. After many years of being mutually exclusive, Microsoft, and Linux, it seems like there will be universal terminal commands. According to linux.com, this is a win for open source. Now for the first time, Ubuntu tools will be available on Windows, Mac, and Ubuntu. As this story is being released, I cannot help but think there is a feeling of uncertainty. I would cite the “Embrace, Extend, Extinguish” phrase found in an internal investigation of Microsoft in the trial of Sun vs. Microsoft.

# My two cents

In conclusion, after investigating the motivations, origins, and risks of open source. I believe it is something that should be embraced. With the emergence of open source, traditional proprietary software companies embracing the open source development paradigm, and the standardization of Ubuntu on Mac and Windows. It is time that we start to modify our standard development methodologies to inherit from the open source model. It is quite possible that the market will feel some strain in the move to open source solutions, and traditional software development will need to be modified from a monopoly value driven ideology to a use value methodology. (The Economics of Open-Source Software) There are group theory implications that we must consider. In the open source model there is the danger of behavior such as groupthink. There is no guarantee that the software will get better with more people contributing, but as Sunstein stated in the closing moments of Infotopia, “it makes sense to bet on optimism”. (p. 225) And even under the crippling debt of my education, along with Sunstein, I’d happily invest my two cents.

# 

# REFERENCES:

Raymond, Eric S. "The Cathedral and the Bazaar." Eric S. Raymond's Home Page. N.p., n.d. Web. 25 Apr. 2016.

Ojha, Abhoy K., and Ravi Anand Rao. "The Emergence of an Organizational Field: The Case of Open Source Software." Vikalpa: The Journal For Decision Makers 39.2 (2014): 127-143. Business Source Premier. Web. 25 Apr. 2016.

Ye, Y. ( 1,2 ), and K. ( 2 ) Kishida. "Toward An Understanding Of The Motivation Of Open Source Software Developers." Proceedings - International Conference On Software Engineering (2003): 419-429. Scopus®. Web. 25 Apr. 2016.

Bretthauer, David. "Open Source Software: A History." Information Technology & Libraries 21.1 (2002): 3. Library, Information Science & Technology Abstracts. Web. 26 Apr. 2016.

"Richard Stallman's Personal Page." Richard Stallman's Personal Page. N.p., n.d. Web. 26 Apr. 2016.

"Brown University." A Framework for Making Ethical Decisions. N.p., n.d. Web. 26 Apr. 2016.

Ridley, Matt. The Origins of Virtue: Human Instincts and the Evolution of Cooperation. New York: Viking, 1997. Print.

Sojer, Manuel, and Joachim Henkel. "License Risks From Ad Hoc Reuse Of Code From The Internet." Communications Of The ACM 54.12 (2011): 74-81. Business Source Premier. Web. 26 Apr. 2016.

Grimes DR (2016) On the Viability of Conspiratorial Beliefs. PLoS ONE 11(1): e0147905. doi:10.1371/journal.pone.0147905

"Licenses & Standards." Open Source Initiative. Web. 27 Apr. 2016. <https://opensource.org/licenses>.

"The GNU General Public License V3.0 - GNU Project - Free Software Foundation." The GNU General Public License V3.0 - GNU Project - Free Software Foundation. Web. 27 Apr. 2016.

"GNU General Public License V1.0 - GNU Project - Free Software Foundation." GNU General Public License V1.0 - GNU Project - Free Software Foundation. Web. 27 Apr. 2016. <http://www.gnu.org/licenses/old-licenses/gpl-1.0.en.html>.

"Copyright Law of the United States of America." U.S. Copyright Office. Web. 25 Apr. 2016.

"The Economics of Open-Source Software." The Open-Source Model and Money. Web. 25 Apr. 2016. <http://bat8.inria.fr/~lang/hotlist/free/licence/raymond/open-economics.html>.

"Xvid." Wikipedia. Wikimedia Foundation. Web. 27 Apr. 2016.

"Fraunhofer FDK AAC." Wikipedia. Wikimedia Foundation. Web. 27 Apr. 2016.