Open Source and Piracy, a Media Analysis

William Jagels, Jonathan Terner, Nikolas Vanderhoof, and Jia Yang

Binghamton University

Open Source and Piracy, a Media Analysis

Introduction

Summmary and Analysis

In a talk given to university students in Paris, Richard Stallman warns of being a prisoner of the digital world. It does not seem likely that Stallman, a computer scientist, would preach against a digital lifestyle. However, Stallman takes issue not with computers and software on their own, but rather with software and services that disrespect their users' freedom. He believes proprietary software allows companies to insert malicious code that can spy on users. Companies that make this kind of software, he says, not only deny users freedom of privacy, but also establish control over those users by keeping their source code secret. To combat this loss of control, Stallman argues "we have to extract people from digital society if it doesn't respect their freedom or we have to make it respect their freedom" (2011, para. 2).

Early in the talk, Stallman clarifies what he means by free software: software that respects users' freedom. Non-free software, he says, has potential to control its users. As an example, he points to computers running Microsoft Windows. Those machines do not respect users' freedom, he says, because they track "data about the use of the computer" (2011, para. 4). Stallman adds that malicious software need not be on users' hardware; an online service can also spy on its users and control users' data . He points out a privacy double standard—companies do not protect users' privacy, but in the case of DRM, take great lengths to protect their own.

Stallman chiefly supports his argument for "digital extraction" with an analysis of what he believes to be the main threats to digital freedom: surveillance, censorship, restricted data formats, proprietary software, Internet services, and computer voting. While Stallman alludes to

the more political topics of surveillance, censorship by governments and computer voting, he

devotes much of his talk to the regulation and distribution of digital media and software. Stallman

envisions a world where users distribute content easily and freely; accordingly, he views any copy

protection measures, such as the DMCA, as a form of censorship (2011, para. 96). Stallman

encourages his audience to fight the "digital handcuffs" imposed by copy protection that seek to

control how they use content (2011, para. 98). Stallman similarly opposes proprietary software.

Stallman supports his claim for prohibiting proprietary software by citing its negative effects on

education. He further argues society as a whole would benefit by severing its dependence on

proprietary software (2011, para. 53). To realize his vision of a world with predominantly free

software, Stallman launched the GNU Project and the Free Software movement.

Stallman supports his arguments with examples carefully tailored to his audience. Because

he delivered his talk at a French university, Stallman illustrates threats to digital freedom using

examples from France and greater Europe. For instance, he mentions the tracking of bicycles in

Paris to demonstrate threats posed by digital surveillance (2011, para. 8), and he mentions secret

file formats employed by Italian public television to demonstrate threats posed by restricted data

formats (2011, para. 30). Stallman also enhances his credibility through his use of French

terms2011. Stallman's explanation of the meaning of "free"—or rather "libre"—software

compliments the subtleties of French (2011, para. 34). Stallman's extensive explanation of free

software also lays an important foundation for the rest of his talk that can be understood by

different audiences.

One should note that Stallman presents his views in a purely ethical context. In fact,

Stallman believes one should only mention free software as "an ethical issue." (2011, para. 63).

Stallman claims additionally that those who coined the term "open source" did so to avoid

discussing the ethics of free software (2011, para. 52). Stallman also refuses to entertain any

arguments on the economics of free software, choosing to focus solely on ethics (2011, para. 34).

When talking about a digital society, Stallman draws his ethical truths about a digital

society from a non-digital society. For example, in his section entitled "the war on sharing,"

Stallman, who views book lending as an "important social act," criticizes the Amazon kindle for

its inability to lend books (2011, para. 98). Stallman maintains that residents of a digital society

must fight to maintain the basic freedoms and rights enjoyed by the non-digital society of the past.

Research Analysis

Practical Advantages of Open Source Software - Jonathan Terner

Stallman maintains that a free society can only come about through the adoption of free

software, stating that he launched the GNU operating system for the express purpose of freedom

and not for technical innovation (2011, para. 48). Stallman criticizes those who associate the free

software movement with the ilk of Linus Torvalds and others who would seek to shift the focus of

the free software movement away from ethics and towards practicality. However, if Stallman truly

wishes to see widespread adoption of free software, then it would behoove him to also encourage

others to capitalize on the practical benefits that made Linux and other open source software

solutions so successful.

While the transparency of source code serves to inhibit the existence of unethical and

malicious code, it also serves to enhance the collaborative nature of software development. A key

feature of open source is that it creates an environment in which users mutually borrow and

contribute back to one another's projects. The community that formed around the development of

Linux embodies this idea very closely. Jim Zemlin, the executive director of the Linux

Foundation, provides an illustration in which changes made to Linux to conserve power on a

cellphone can be used to benefit supercomputers that also run Linux (2013, 11:34). Similarly, the

increased availability of code allows developers to adapt software to new challenges. By

inheriting httpd's Common Gateway Interface standard, the development of the Apache web

server was expedited, as it's creators no longer had the need to build these components from

scratch (Bisson, 2007, p. 7).

The overall restrictiveness of proprietary software presents a large disadvantage to its users.

Because software is built from source code, the support for any piece of software is ultimately

tied to the ability to edit and view that source code. Having access to a product's source code

decreases dependency on that product, which is critical should that product become discontinued

or should its vendor declare bankruptcy. Thus, vendor lock-in threatens software users with

eventual obsolescence, even in cases where it was possible to retroactively obtain the rights to

proprietary code . In one such case involving the migration to new electronic voting machines in

New York, the code obtained to make these machines function was licensed for testing and not for

deployment (Colannino, 2012, p. 916). Moreover, this issue could have been avoided had the

code been released with a free software license, such as the GNU public license. Open source

licenses protect the continuous improvements made on software that prevent vendor lock-in

(Colannino, 2012, p. 919).

Open source software also has a unique quality assurance mechanism in the form of its

peer-review process. The communities that develop around open source software are comprised

of passionate individuals who are devoted to problem solving (Bisson, 2007, p. 19). These users

endlessly patch vulnerabilities and provide new features. In 2013, it was determined that 6,782

lines of code are added and subtracted to Linux on a daily basis (Zemlin, 2013, 12:03).

Consequently, the constant revision of Linux ensures its stability and relevance. Yet, the process of open source peer-review is not as straightforward as some would believe. In a study analyzing the bug reports for Mozilla Firefox, Wang et. al. reevaluates "Linus's Law," or the idea that "given enough eyeballs, all bugs are shallow"(Wang, Shih, & Carroll, 2015, p. 52). They determine that Mozilla's peer-review process lacked a central focus due to the differing levels skill levels of bug reporters, as disputes commonly arose over the relative importance of certain bugs (2015, p. 52). However, Wang et. al. claim that careful labeling of duplicate bug reports would solve this problem (2015, p. 52).

Due to the ease and availability of open source software, one can easily match a software need with an open source solution. When developing Facebook, Mark Zuckerberg used Linux and other free software to build the word's most successful social media platform (Zemlin, 2013, 6:18). One can even argue that a software need—not an ethical need—provided Stallman with the impetus to launch the free software movement. Stallman's animosity towards proprietary software originated from a printer's software's lack of extensibility. When denied permission to add features to the software, Stallman vowed that he would never let a programmer share in his frustration (2015, para. 19).

Economics of Open Source - William Jagels

Stallman (2011) explains the need for free software in his lecture, but does not explain the economic power of free software. Although free software is by definition free of charge, that does not mean that free software has no place in a capitalistic society. Major companies contribute to open source software, and some are even core contributors and maintainers. Using open source enables these companies to leverage the knowledge of the community, which is not possible when

creating closed source software. Open source software is also useful to companies in its finished

form, for example, Apache web server ran two-thirds of major web sites (Powell, 2012, p. 696)

Open source software can also become the basis for entrepreneurs starting new companies,

as liberal software licenses do not restrict any kind of usage, including commercial. One such

example is OpenSimulator, which is a BSD licensed 3D software system. Because the BSD

license is commercially friendly, a startup can use OpenSimulator in their own software, and

redistribute it (Yetis-Larsson, Teigland, & Dovbysh, 2014, p. 477-8).

OpenSimulator's community was the subject of a study on open entrepreneurship,

examining how networking in an open source community can help an entrepreneur fulfill business

goals. The study ultimately found that entrepreneurs rely on the networks they form by actively

participating in the open source community when starting new firms. The study also found that

entrepreneurs make open source communities better with their political skills while still achieving

business goals (Yetis-Larsson et al., 2014); sharing environments do not exclude business

opportunities, in fact, these communities greatly benefit capitalism. An economy where costs to

start a business are low can foster innovative technologies and increase the standard of living for

everyone, since in-house software can be costly, open source can lower costs for business

ventures.

Open source businesses have proved sustainable, Red Hat being an excellent example. Red

Hat makes money from subscriptions, training, and services instead of by licensing their software

(Red Hat Inc., 2015, p. 31). Red Hat's software is licensed under the GNU GPL, and also

promises to not enforce patent rights on their code (Red Hat Inc., 2015, p. 65), making their

software freely usable and distributable by anyone. CentOS, for example, is a free clone of Red

Hat Enterprise Linux, making the power of Red Hat software available to anyone who can

download an installer. Despite the existence of a free clone, Red Hat still makes a large amount of

money, $524 Million of revenue in Q3 FY 2016, up from $456 Million in Q3 FY 2015 (Red Hat

Inc., 2015, p. 24). Red Hat is living proof that developing open source software can still generate

revenue, which contributes to the economy by providing valuable services and advancing global

technology.

When companies can make open source software profitable, the nature of open source

hugely benefits the overall economy. Open source technology enables a more efficient economy,

because individuals and firms do not have to go through the laborious process of building

everything from scratch, or bear the expense of paying large licensing fees to get anywhere with

software development. However, open sourcing a piece of software is not always the best solution

as it could mean sacrificing profits. One paper studied the economic viability of open sourcing a

piece of software at a certain point. The paper concludes that it makes sense to open source

software that has high costs to maintain quality, because open source contributors can reduce the

cost of maintaining that software in-house. However, the costs to switch to open source must also

be considered before making the switch (Caulkins et al., 2013). One example cited in the paper is

the Doom-engine, which was very high quality software when it was released, but competitors

eventually caused the cost of keeping the engine up to date to be too much for id Software. Id

Software open sourced the engine in 1997, but still made money by selling content that runs on

the engine (Caulkins et al., 2013, p. 1188).

Although Stallman's lecture can give an almost Marxist view of software, open source still

has a big place in a capitalistic society. Much like global trade, sharing knowledge can benefit all

parties involved and improve the overall efficiency of the economy. Wasted resources, namely

human resources, decrease the efficiency of the economy, so reducing the amount of duplicated

work in software development is good for the overall economy.

Security Differences between Open-Source and Closed-Source - Nikolas Vanderhoof

If a closed source developer adds malicious code into their product, then according to Stallman, its users are left powerless to fix it (2011, para. 36). One can reasonably assume that some closed-source developers do not try to introduce malicious features. But any developer, closed-source or open source,can still accidentally introduce security vulnerabilities. Disregarding Stallman's position on surveillance present in proprietary software, when one examines both open source and closed-source software, neither has a clear security advantage over the other.

With open source software, users can more easily find security holes because they have access to source code, and users can propose fixes immediately. With closed-source software, when users notice a security flaw, they can report it to the developers, but the users have to rely on the developers to implement a fix (Schryen, 2009, para. 17).

Because closed-source software goes through a tremendous amount of testing by the group maintaining the project, the speed of patches and fixes is limited by the size of the development team. This becomes more of an issue as projects get larger; the developers have much more code to go through in order to fix the bug. With open source however, as the project grows, so does the number of people working on the code review process, as some users actively try to help fix the software (Boulanger, 2005, p. 245).

Kadura and Schryen compare security vulnerabilities in Microsoft Office and Apache OpenOffice. They only count vulnerabilities listed as "Common Vulnerabilities and Exposures (CVE) entries by the CVE editorial board, which was itself created by the MITRE corporation" (2009). They found that Microsoft Office had a total of 108 vulnerabilities, while Apache

OpenOffice had only 16. Both Microsoft Office and Apache OpenOffice had comparable numbers

of vulnerabilities listed as low severity, three and two respectively, but Microsoft Office had

roughly 7 times as many medium and high severity vulnerabilities as Apache OpenOffice. The

authors recognize that the number of medium and high severity vulnerabilities for Apache

OpenOffice may be substantially lower because patches for vulnerabilities were applied before

the CVE editorial board tallied the numbers (Schryen, 2009).

Many security vulnerabilities do not depend on the viewing of source code. In an attempt to

gain access to user data, black-hat hackers can use a program that tries millions of passwords until

one works. Many programs have ways of detecting this, so hackers often employ cleverer

approaches. With methods such as SQL injection and buffer overflow, one can find vulnerabilities

by providing strange input to a program. If the program cannot handle the input, the program may

have unexpected behavior that may give the hacker access he or she should not have.(Clarke,

Dorwin, & Nash, n.d., p. 8-9).

Even if a program can check for brute force attacks, or is designed to handle any format of

input without failing, its developers can never account for a corrupt developer or administrator

that works from the inside to take advantage of users' data. Humans are often the weakest link in

software security. Otherwise, phishing, pharming, and other similar scams would not be effective.

Conclusion

Richard Stallman's ideas, although radical, have merit. Open source has many practical

advantages, including the ability to build off one another's work without worrying about

intellectual property lawsuits. Proprietary software is generally hard to decode to learn how it

works, and even if one was successful, publishing information about the internal workings could

be considered a breach of license. With open source, users gain the power to fix problems and

create their own flavors to suit their specific needs, without having to contact the original

publisher. The open source community is a powerful entity, with brilliant problem solvers

collaborating in an open setting, and this helps progress technology.

The open source community also is a place where entrepreneurs can network and eventually

form new technology companies based on open source technologies. By lowering the entry costs,

innovation is bound to appear, contributing to the economy and raising the standard of living.

Existing open source businesses provide great value to other firms and individuals, since their well

maintained software can be used by anyone, and company revenue improves the quality of the

software. Companies with legacy software can also benefit from open source, by releasing their

code they can gain the help of the open source community, lowering the cost to maintain quality.

Along with quality, the open source community is also great for finding security

vulnerabilities in software, as there's a large collection of people and firms who are willing to find

and fix security flaws. With access to source code, the users are actually able to have confidence

that their software is operating in a certain way, because they can simply pull up the source code.

Lowering the amount of security holes in software helps fight cybercrime, and lead to a safer

computing experience for all parties involved.

Stallman's ideas for a free and open source software ecosystem can be applied to an extent,

bringing collaboration into software development and promoting idea sharing. With free and open

source software, software developers can easily build off each other and avoid repeating work and

wasting time building things from scratch.

References

Andrés, A. R., & Asongu, S. A. (2013). Fighting software piracy: Which governance tools matter

in africa? *Journal of Business Ethics*, *118*(3), 667-682. doi: 10.1007/s10551-013-1620-7

Bisson, C. (2007). What makes open source work? *Library Technology Reports*, *43*(3).

Boulanger, A. (2005). Open-source versus proprietary software: Is one more reliable and secure

than the other? *IBM Systems Journal*, *44*(2), 239-248.

Caulkins, J. P., Feichtinger, G., Grass, D., Hartl, R. F., Kort, P. M., & Seidl, A. (2013, Jun). When

to make proprietary software open source. *Journal of Economic Dynamics and Control*,

37(6), 1182-1194. doi: 10.1016/j.jedc.2013.02.009

Clarke, R., Dorwin, D., & Nash, R. (n.d.). Is open source software more secure?

Colannino, J. C. (2012). Free and open source software in municipal procurement: The

challenges and benefits of cooperation. *Fordham Urban Law Journal*, *39*(4), 903 - 929.

Cosovanu, C. (2006, Jan). Open source software in eastern europe and other emerging markets:

The moral alternative to piracy? *Journal of Internet Law*, *9*.

Mark, O. (2008, Mar 31). Open sourcing can tackle software piracy. *Africa News Service*.

Pahlka, J. (2012, Feb). *Coding a better government* [Video file]. Retrieved from

https://www.ted.com/talks/jennifer_pahlka_coding_a_better_government

Pankaja, N., & Raj, M. (2013). Proprietary software versus open source software for education.

American Journal of Engineering Research, *2*(7), 124-130.

Powell, A. (2012, Sep). Democratizing production through open source knowledge: from open

software to open hardware. *Media, Culture & Society*, *34*(6), 691-708. doi:

10.1177/0163443712449497

Pykäläinen, T., Yang, D., & Fang, T. (2009). Alleviating piracy through open source strategy: An exploratory study of business software firms in china. *The Journal of Strategic Information Systems*, *18*(4), 165-177. doi: 10.1016/j.jsis.2009.10.001

Red Hat Inc. (2015, Nov). *Form 10-Q.* Retrieved from https://www.nyse.com/quote/XNYS:RHT/sec

Resondry, S., Boudaoud, K., Kamel, M., Bertrand, Y., & Riveill, M. (2014, Aug). An alternative version of https to provide non-repudiation security property. In *Wireless communications and mobile computing conference (IWCMC), 2014 international* (p. 536-541). doi: 10.1109/IWCMC.2014.6906413

Schryen, K. R., G. (2009). Open source vs. closed source software: Towards measuring security. In *Sac '09 proceedings of the 2009 acm symposiumon applied computing* (p. 2016-2023). doi: 10.1145/1529282.1529731

Stallman, R. M. (2011, October 19). *A free digital society - what makes digital inclusion good or bad?* [Lecture transcript]. Retrieved from http://www.gnu.org/philosophy/free-digital-society.en.html

Stallman, R. M. (2015). *The gnu project.* Free Software Foundation. Retrieved from http://www.gnu.org/gnu/thegnuproject.en.html

Wang, J., Shih, P. C., & Carroll, J. M. (2015). Revisiting Linus's law: Benefits and challenges of open source software peer review. *International Journal of Human-Computer Studies*, *77*, 52-65. doi: 10.1016/j.ijhcs.2015.01.005

Yetis-Larsson, Z., Teigland, R., & Dovbysh, O. (2014). Networked entrepreneurs: How entrepreneurs leverage open source software communities. *American Behavioral Scientist*, *59*(4), 475-491. doi: 10.1177/0002764214556809

Zemlin, J. (2013). *What the tech industry has learned from Linus Torvalds* [Video file]. Retrieved

   from https://www.youtube.com/watch?v=7XTHdcmjenI