Why Open Source Misses the Point of Free Software

GNU philosophy

This essay was originally published on http://gnu.org, in 2007.

Why Open Source Misses the Point of Free Software

When we call software "free," we mean that it respects the users' essential freedoms: the freedom to run it, to study and change it, and to redistribute copies with or without changes. This is a matter of freedom, not price, so think of "free speech," not "free beer."

These freedoms are vitally important. They are essential, not just for the individual users' sake, but for society as a whole because they promote social solidarity—that is, sharing and cooperation. They become even more important as our culture and life activities are increasingly digitized. In a world of digital sounds, images, and words, free software becomes increasingly essential for freedom in general.

Tens of millions of people around the world now use free software; the public schools of some regions of India and Spain now teach all students to use the free GNU/Linux operating system. Most of these users, however, have never heard of the ethical reasons for which we developed this system and built the free software community, because nowadays this system and community are more often spoken of as "open source," attributing them to a different philosophy in which these freedoms are hardly mentioned.

The free software movement has campaigned for computer users' freedom since 1983. In 1984 we launched the development of the free operating system GNU, so that we could avoid the nonfree operating systems that deny freedom to their users. During the 1980s, we developed most of the essential components of the system and designed the GNU General Public License (GNU GPL) to release them under—a license designed specifically to protect freedom for all users of a program.

Not all of the users and developers of free software agreed with the goals of the free software movement. In 1998, a part of the free software community splintered off and began campaigning in the name of "open source." The term was originally proposed to avoid a possible misunderstanding of the term "free software," but it soon became associated with philosophical views quite different from those of the free software movement.

Some of the supporters of open source considered the term a "marketing campaign for free software," which would appeal to business executives by highlighting the software's practical benefits, while not raising issues of right and wrong that they might not like to hear. Other supporters flatly rejected the free software movement's ethical and social values. Whichever their views, when campaigning for open source, they neither cited nor advocated those values. The term "open source" quickly became associated with ideas and arguments based only on practical values, such as making or having powerful, reliable software. Most of the supporters of open source have come to it since then, and they make the same association.

Nearly all open source software is free software. The two terms describe almost the same category of software, but they stand for views based on fundamentally different values. Open source is a development methodology; free software is a social movement. For the free software movement, free software is an ethical imperative, because only free software respects the users' freedom. By contrast, the philosophy of open source considers issues in terms of how to make software "better"—in a practical sense only. It says that nonfree software is an inferior solution to the practical problem at hand. For the free software movement, however, nonfree software is a social problem, and the solution is to stop using it and move to free software.

"Free software." "Open source." If it's the same software, does it matter which name you use? Yes, because different words convey different ideas. While a free program by any other name would give you the same freedom today, establishing freedom in a lasting way depends above all on teaching people to value freedom. If you want to help do this, it is essential to speak of "free software."

We in the free software movement don't think of the open source camp as an enemy; the enemy is proprietary (nonfree) software. But we want people to know we stand for freedom, so we do not accept being mislabeled as open source supporters.

Common Misunderstandings of "Free Software" and "Open Source"

The term "free software" is prone to misinterpretation: an unintended meaning, "software you can get for zero price," fits the term just as well as the intended meaning, "software which gives the user certain freedoms." We address this problem by publishing the definition of free software, and by saying, "Think of 'free speech,' not 'free beer.'" This is not a perfect solution; it cannot completely eliminate the problem. An unambiguous and correct term would be better, if it didn't present other problems.

Unfortunately, all the alternatives in English have problems of their own. We've looked at many that people have suggested, but none is so clearly "right" that switching to it would be a good idea. (For instance, in some contexts the French and Spanish word "libre" works well, but people in India do not recognize it at all.) Every proposed replacement for "free software" has some kind of semantic problem—and this includes "open source software."

The official definition of "open source software"[1] (which is published by the Open Source Initiative and is too long to include here) was derived indirectly from our criteria for free software. It is not the same; it is a little looser in some respects, so the open source people have accepted a few licenses that we consider unacceptably restrictive. Also, they judge solely by the license of the source code, whereas our criterion also considers whether a device will let you *run* your modified version of the program. Nonetheless, their definition agrees with our definition in most cases.

However, the obvious meaning for the expression "open source software"—and the one most people seem to think it means—is "You can look at the source code." That criterion is much weaker than the free software definition, much weaker also than the official definition of open source. It includes many programs that are neither free nor open source.

Since that obvious meaning for "open source" is not the meaning that its advocates intend, the result is that most people misunderstand the term. According to writer Neal Stephenson, "Linux is 'open source' software, meaning simply, anyone can get copies of its source code files."[2] I don't think he deliberately sought to reject or dispute the "official" definition. I think he simply applied the conventions of the English language to come up with a meaning for the term. The state of Kansas published a similar definition: "Make use of open-source software (OSS). OSS is software for which the source code is freely and publicly available, though the specific licensing agreements vary as to what one is allowed to do with that code."

[1] See http://opensource.org/docs/osd for the full definition.

[2] Neal Stephenson, *In the Beginning. . . Was the Command Line* (New York: HarperCollins Publishers, 1999), p. 94.

The *New York Times* has run an article that stretches the meaning of the term to refer to user beta testing[3]—letting a few users try an early version and give confidential feedback—which proprietary software developers have practiced for decades.

Open source supporters try to deal with this by pointing to their official definition, but that corrective approach is less effective for them than it is for us. The term "free software" has two natural meanings, one of which is the intended meaning, so a person who has grasped the idea of "free speech, not free beer" will not get it wrong again. But the term "open source" has only one natural meaning, which is different from the meaning its supporters intend. So there is no succinct way to explain and justify its official definition. That makes for worse confusion.

Another misunderstanding of "open source" is the idea that it means "not using the GNU GPL." This tends to accompany another misunderstanding that "free software" means "GPL-covered software." These are both mistaken, since the GNU GPL qualifies as an open source license and most of the open source licenses qualify as free software licenses.

The term "open source" has been further stretched by its application to other activities, such as government, education, and science, where there is no such thing as source code, and where criteria for software licensing are simply not pertinent. The only thing these activities have in common is that they somehow invite people to participate. They stretch the term so far that it only means "participatory."

Different Values Can Lead to Similar Conclusions. . . but Not Always

Radical groups in the 1960s had a reputation for factionalism: some organizations split because of disagreements on details of strategy, and the two daughter groups treated each other as enemies despite having similar basic goals and values. The right wing made much of this and used it to criticize the entire left.

Some try to disparage the free software movement by comparing our disagreement with open source to the disagreements of those radical groups. They have it backwards. We disagree with the open source camp on the basic goals and values, but their views and ours lead in many cases to the same practical behavior—such as developing free software.

As a result, people from the free software movement and the open source camp often work together on practical projects such as software development. It is remarkable that such different philosophical views can so often motivate different people to participate in the same projects. Nonetheless, there are situations where these fundamentally different views lead to very different actions.

The idea of open source is that allowing users to change and redistribute the software will make it more powerful and reliable. But this is not guaranteed. Developers of proprietary software are not necessarily incompetent. Sometimes they produce a program that is powerful and reliable, even though it does not respect the users' freedom. Free software activists and open source enthusiasts will react very differently to that.

[3] Mary Jane Irwin, "The Brave New World of Open-Source Game Design," *New York Times*, online ed., 7 February 2009, http://www.nytimes.com/external/gigaom/2009/02/07/07gigaom-the-brave-new-world-of-open-source-game-design-37415.html.

A pure open source enthusiast, one that is not at all influenced by the ideals of free software, will say, "I am surprised you were able to make the program work so well without using our development model, but you did. How can I get a copy?" This attitude will reward schemes that take away our freedom, leading to its loss.

The free software activist will say, "Your program is very attractive, but I value my freedom more. So I reject your program. Instead I will support a project to develop a free replacement." If we value our freedom, we can act to maintain and defend it.

Powerful, Reliable Software Can Be Bad

The idea that we want software to be powerful and reliable comes from the supposition that the software is designed to serve its users. If it is powerful and reliable, that means it serves them better.

But software can be said to serve its users only if it respects their freedom. What if the software is designed to put chains on its users? Then powerfulness means the chains are more constricting, and reliability that they are harder to remove. Malicious features, such as spying on the users, restricting the users, back doors, and imposed upgrades are common in proprietary software, and some open source supporters want to implement them in open source programs.

Under pressure from the movie and record companies, software for individuals to use is increasingly designed specifically to restrict them. This malicious feature is known as Digital Restrictions Management (DRM) (see http://defectivebydesign.org) and is the antithesis in spirit of the freedom that free software aims to provide. And not just in spirit: since the goal of DRM is to trample your freedom, DRM developers try to make it hard, impossible, or even illegal for you to change the software that implements the DRM.

Yet some open source supporters have proposed "open source DRM" software. Their idea is that, by publishing the source code of programs designed to restrict your access to encrypted media and by allowing others to change it, they will produce more powerful and reliable software for restricting users like you. The software would then be delivered to you in devices that do not allow you to change it.

This software might be open source and use the open source development model, but it won't be free software since it won't respect the freedom of the users that actually run it. If the open source development model succeeds in making this software more powerful and reliable for restricting you, that will make it even worse.

Fear of Freedom

The main initial motivation of those who split off the open source camp from the free software movement was that the ethical ideas of "free software" made some people uneasy. That's true: raising ethical issues such as freedom, talking about responsibilities as well as convenience, is asking people to think about things they might prefer to ignore, such as whether their conduct is ethical. This can trigger discomfort, and some people may simply close their minds to it. It does not follow that we ought to stop talking about these issues.

That is, however, what the leaders of open source decided to do. They figured that by keeping quiet about ethics and freedom, and talking only about the immediate practical benefits of certain free software, they might be able to "sell" the software more effectively to certain users, especially business.

This approach has proved effective, in its own terms. The rhetoric of open source has convinced many businesses and individuals to use, and even develop, free software, which has extended our community—but only at the superficial, practical level. The philosophy of open source, with its purely practical values, impedes understanding of the deeper ideas of free software; it brings many people into our community, but does not teach them to defend it. That is good, as far as it goes, but it is not enough to make freedom secure. Attracting users to free software takes them just part of the way to becoming defenders of their own freedom.

Sooner or later these users will be invited to switch back to proprietary software for some practical advantage. Countless companies seek to offer such temptation, some even offering copies gratis. Why would users decline? Only if they have learned to value the freedom free software gives them, to value freedom in and of itself rather than the technical and practical convenience of specific free software. To spread this idea, we have to talk about freedom. A certain amount of the "keep quiet" approach to business can be useful for the community, but it is dangerous if it becomes so common that the love of freedom comes to seem like an eccentricity.

That dangerous situation is exactly what we have. Most people involved with free software, especially its distributors, say little about freedom—usually because they seek to be "more acceptable to business." Nearly all GNU/Linux operating system distributions add proprietary packages to the basic free system, and they invite users to consider this an advantage rather than a flaw.

Proprietary add-on software and partially nonfree GNU/Linux distributions find fertile ground because most of our community does not insist on freedom with its software. This is no coincidence. Most GNU/Linux users were introduced to the system through "open source" discussion, which doesn't say that freedom is a goal. The practices that don't uphold freedom and the words that don't talk about freedom go hand in hand, each promoting the other. To overcome this tendency, we need more, not less, talk about freedom.

Conclusion

As the advocates of open source draw new users into our community, we free software activists must shoulder the task of bringing the issue of freedom to their attention. We have to say, "It's free software and it gives you freedom!"—more and louder than ever. Every time you say "free software" rather than "open source," you help our campaign.

Notes

- Joe Barr's article "Live and Let License" (ITworld.com, 22 May 2001, http://www.itworld.com/LWD010523vcontrol4) gives his perspective on this issue.
- Karim R. Lakhani and Robert G. Wolf's paper on the motivation of free software developers ("Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," in *Perspectives on Free and Open Source Software*, edited by J. Feller and others (Cambridge: MIT Press, 2005)) says that a considerable fraction are motivated by the view that software should be free. This is despite the fact that they surveyed the developers on SourceForge, a site that does not support the view that this is an ethical issue.