

It's Time to Consider Open Source Software

Jay Pfaffman

University of Tennessee, Knoxville

1122 Volunteer Blvd,

Claxton 443

Knoxville, TN 37996

<pfaffman@utk.edu>

865-974-0497

Fax: (865) 974-8103

Abstract

There now now Open Source Software (OSS) alternatives for many of the proprietary applications typically used in schools. Though much OSS is written for the Open Source GNU/Linux operating system many of the most useful applications are also available for Windows and Mac OS X. This paper points out that many of the inconveniences of using proprietary software are unnecessary and often forgotten. Next, it proposes responses to many reasons that people may have for avoiding OSS. OSS is often developed by amateurs who develop software out of the same expertise and dedication that allows amateur astronomers to make useful contributions to that field. There are many examples of low-quality OSS just as there are examples of proprietary software. Using OSS requires neither extensive retraining nor badly inadequately preparing students to work with “standard” applications in the future. Finally the paper provides examples of alternatives for the most commonly-used applications in schools.

Keywords: Ethics, training, software

Open Source Software Background

In 1985 Richard Stallman released the GNU Manifesto (Stallman, 1985), which described his dream of a free UNIX-like operating system and associated software needed to accomplish one's work. Few people paid attention. Few people had computers that could run anything like UNIX anyway. In 1991 the Linux kernel was publicly released, and soon thereafter Stallman's dream started looking like a possibility. For the past five years or so, using Linux, as an Internet server has been fairly common. It is a good bet that somewhere in your district a computer is running Linux as a mail, DNS, or web server, probably tucked in a closet somewhere. Or perhaps you have used the Google search engine, which runs on Linux servers. One pundit has claimed "At this point, anyone proposing to run Windows on servers should be prepared to explain what they know about servers that Google, Yahoo, and Amazon don't" (Graham, 2005). Open Source Apache is the most widely used web server. BIND, the software that almost everyone uses to convert domain names to IP addresses, is also Open Source. If you have ever typed a domain name instead of an IP number, you have used this software. Open Source Software (OSS) isn't limited to the GNU/Linux operating system or servers. Now there are enough OSS applications designed for normal people using mainstream operating systems that it's time to think about using OSS outside of the server closet.

If you use Microsoft Windows, Mac OS X or Linux OpenOffice.org (OOo) is a good replacement for Microsoft Office. Abiword, is an excellent word processor that works fine even on older hardware. The GNU Image Manipulation program (The GIMP) is a good replacement for Photoshop. PDFCreator will enable any Windows application that can

print to create PDF files (though OOo has this built in). Firefox not only has several very useful features that Internet Explorer lacks, but also is impervious to most viruses and worms that attack Internet Explorer. Nvu and Bluefish are also surprisingly good alternatives to Frontpage and Dreamweaver. That's right. Taking advantage of Open Source Software no longer means giving up your favorite operating system or re-learning how to use your computer.

There are many advantages—technical, economic, and political—to using OSS rather than the proprietary software that most people use. Proprietary software often provides significant advantages over their OSS “competitors,” but these advantages should be carefully weighed against the disadvantages that many of us have come to accept as unavoidable. This article is to help you better understand—or explain—the advantages of OSS and when it may be cheaper and better than the familiar commercial alternatives.

Convenience

Proprietary software is inconvenient. It is inconvenient to purchase additional licenses when you buy new machines. It is inconvenient to negotiate a new license agreement each year. It is inconvenient to support multiple versions of a package on campus because money is not available to upgrade software on all computers when a new version comes out. It is inconvenient for a technology coordinator to be forced with an ethical dilemma when a teacher buys a home computer and wants the school's software to be installed. It is inconvenient that we cannot provide students with the same software that they use at school for their home computers. It is inconvenient when a vendor who previously provided a package free to schools changes the license agreement after a school has invested in training

people to use it. It is inconvenient to be forced to perform a software audit to document that every piece of software has a valid license. Thanks to the work of Stallman and thousands of others, these inconveniences are becoming increasingly unnecessary.

Concerns about Open Source Software

Many people think that “free software” is too good to be true. “There has to be some catch.” “You can’t get something for nothing.” But there is no catch. Economists now are starting to recognize the contributions that ProAms—people who do high quality work in a field other than the one that earns them their money—have on the economy and society (Leadbeater & Miller, 2004). Amateur astronomers, for example, now regularly make new discoveries. Amateur musicians and songwriters generate money for coffee houses and bars. Several of Einstein’s most famous papers were written while he worked in a patent office. So it is with software. Many people develop software because they enjoy writing software. Some do it for the respect that they gain in online communities that form around these Open Source applications. This section addresses some common concerns—or myths—that people have about OSS.

Myth: You Get What You Pay For

Many people remember low-quality and niche shareware from the 1980s and think that today’s Open Source Software is the same. Maybe you fondly remember some cool little utility like 4DOS, but wasn’t something that you were going to stake even a day’s instruction on and besides, it was hard to track down the applications that really were useful even if you spent hours on bulletin board systems downloading the stuff. The Internet has enabled

people to work together in spite of geographic disparities, so instead of a program being developed by one person in his garage, it can now be the work of dozens of people, many of whom make some contributions as part of their paying job. Also, the Internet makes it possible to find and get reliable reviews software, proprietary or Open Source. Perhaps the biggest advantage that the Internet provides is that one can now garner considerable fame by managing or contributing to a successful project. Companies make profits distributing and supporting OSS. Sun and Netscape turned their proprietary products into Open Source projects (StarOffice and Netscape Navigator, respectively). If those commercial examples are not enough, remember that the Apple computer was designed by a couple of amateurs in a garage—and that the source code to the ROMs was included in the manuals that came with the computer. People who work for fun are more likely to approach that work with pride and dedication than are those whose do it to put food on their table.

Another way that pay for what you get with “free” software is by giving away information that you meant to keep private. Wired magazine reported that many users consider spyware (software that tracks your web surfing habits or records your passwords) the cost for using free software (Delio, 2004). This suggests that many people do not understand the distinction between “free” software and Open Source Software (OSS). Because the source code to OSS is available for programmers to inspect there are no secret “features” like the spyware that comes as a payload in many “free” programs. This has partly to do with the economic model of these profoundly different types of free software. Many OSS projects are supported in part by vendors who make their money selling support. Others work on OSS as a way to make a name for themselves or because they just like working on computer programs. Those who distribute proprietary software for free are doing it for some kind of

compensation. Sometimes it is to encourage you to buy a full-featured version of the software or to provide you with a viewer (e.g. Acrobat Reader) so that others will be encouraged to buy a program to produce things to view (e.g. Acrobat Distiller). Others make their money by collecting marketing information on you, graciously popping up ads depending on what web sites you have gone to, or stealing private information about you. For software that is free but not Open Source, one should be very mindful about what the motivation for providing the software is. If it seems too good to be true, it probably is.

Myth: Open Source Software is Low Quality

There is a lot of really, really bad Open Source Software. Some projects do not get fully developed. Some are missing essential features. Some are difficult to install. Some just don't work. Don't use that software. Use only the best OSS, that works reliably and meets your needs. There is plenty of bad software that you can pay for, but that does not stop people from paying for good software. So it should be for OSS.

Myth: Using Commercial Software Helps Avoid Viruses

Gone are the days that you could tell people that if they did not run an application that was infected with a virus they would be safe. Then came Microsoft Word macro viruses. Now your computer can be infected by a worm or virus simply by accessing a web page with Internet Explorer or opening an email with Outlook, a problem that may be mitigated by diligently updating your operating system. Both of these programs gain some power and ease of use by being tightly coupled with the operating system, but that tight coupling comes at the price of security. The same mechanisms that make for this ease of use are the very

ones that make it easy to write crippling viruses. One good way to avoid a large percentage of viruses and worms is to avoid using Internet Explorer, Microsoft Office, and Microsoft Outlook. The Open Source alternatives are safer not only because they lack hooks deep into the operating system allowing them to bypass its security mechanisms, but also because they are less attractive targets for those writing viruses and worms. This latter benefit, of course, may be mitigated as time goes on, but because the source code is publicly available, the safety of OSS can be audited publicly in ways that proprietary software cannot.

Myth: Moving to OSS Will Require Expensive Retraining

This argument is quite compelling. Until one remembers that people move to new versions of applications all the time, and not all new features are met immediately with open arms (“How *do* I make that dancing paper clip go away forever!?”). For most people, moving to an open source alternative to a familiar piece of software is no more disruptive than any other upgrade. Having the freedom to upgrade every computer in your district to the latest version at the same time, without writing a purchase order or even counting the number of licenses that you need is a significant training advantage. When you train people to use OSS you know that they will have the same version back in their classrooms and homes. When everyone is using the same version, people can provide better support for each other.

Training may not be the issue. People, and perhaps especially teachers are often reticent to try new computer programs. Those who are most likely to be reluctant to try new things are often those who are most likely not to notice subtle differences between one program and another. A very practical joke sometimes played by computer coordinators is to upgrade a user’s proprietary office suite with OpenOffice.org without notification and

leaving the Microsoft Word icon on the desktop. Users often fail to notice the difference.

This retraining issue was also once very real. In the past, different programs had very different user interfaces. Moving from WordStar to WordPerfect, or WordPerfect to Microsoft Word was traumatic. Since then, however, the computer's interface has become much more consistent. The menu structure is similar even across different types of applications. Now except for the most esoteric of functions, a word processor is a word processor, and a spreadsheet, a spreadsheet.

Myth: Students Need to Learn the Standard Applications

By the time our students get into the job market the applications that are available today will be hopelessly antiquated. It *is* important that students know how to use word processors to be better writers and spreadsheets to work with numbers, but if we are teaching our students in such a way that they will not be able to fluidly deal with small differences between one application and another version then we are doing these students a disservice.

For most tasks that students and teachers do an OSS application and its popular proprietary alternative are indistinguishable. The menus are similar. The commonly-used keyboard shortcuts are the same. Of course a few things are different; one review of OpenOffice.org's Writer points out that ooWriter encourages people to use styles rather than individual formatting commands, which results in documents that are formatted more consistently and are easier to adjust when the writing is done (Byfield, 2002). Of course Microsoft Word has styles too, but according to one author they are more cumbersome to use.

Myth: OSS is not as Well Supported as Proprietary Software

This concern might be troublesome, except that proprietary software is also unsupported. Microsoft explicitly makes vendors responsible for support. Microsoft's End User License Agreement explicitly claims no warranty after ninety days (Zymaris, 2003). The first place that most people turn when they encounter a problem with any piece of software is their favorite Internet search engine (which likely runs on Open Source Software). Sometimes a vendor's web site will be the one that provides the answer, but very often, the solution to whatever the problem is will have been posted on a user forum somewhere. The likelihood of getting a vendor to provide individualized help via phone or email is rather remote. By using OSS you might be able to spend the money that would have purchased licenses on people who will be accountable to provide you with support. (It might also free up some time for the person who would otherwise be engaged in dealing with software audits.) If your district spends say, US\$50,000/year in licensing fees for MS Office, you could instead use OSS and hire someone to provide support and training to your teachers. That will probably be more effective than waiting on hold for someone a continent away to help solve your problem.

Educators Pay for Software—Twice

Software is more valuable the more people who use it. This is the primary reason that Microsoft Office is so popular. Everyone uses it because everyone uses it. And when enough of your friends have upgraded to a new version that produces files that you can't read reliably, you upgrade.

When we train our teachers and students to use a piece of software we are making that software more valuable. Vendors know this. Business sense, not altruism, is why education

gets deep discounts on software. I once spoke to vendor of some online grade book software who, upon learning that I train teachers, was very interested in my using it in my classes.

“What does it cost?” I asked.

“You can use it for free for as long as you like.”

“And once I addict my students to your software,” I asked pointedly, afraid that I was being rude, “what will it cost them?”

The vendor became excited. “That’s what we were talking about in our last sales meeting!” It is good business sense.

When we train our teachers and students to use proprietary software, we effectively spend their money as well as our own. A \$500 Dell computer costs another \$150 (%30) with Microsoft Office Basic; the version for students and teachers is available for about \$100 (%20). These costs do not come directly out of the school’s budget, but when multiplied by the number of students and teachers who buy Microsoft Office because it is what is used at school, it becomes a significant expense. If a medium-sized school district has 17,000 households with children and only 10 percent buy Office because it will help their students in school, the district has indirectly spent almost US\$200,000 in parents’ money. Whether schools should be effectively taxing its constituents in this way is an issue that deserves more attention.

Examples of High Quality OSS

There are some examples of OSS that all technology educators should know about. They replace commonly-used applications, are mature, and have obvious uses in classrooms. For many people, the convenience of the having installers on an easy-to-use CD makes up for

the fact that often more recent versions of these programs are available for download from the respective software's web site. Unless indicated otherwise all of the programs mentioned in this section are available for Windows and Mac OS X.

OpenOffice.org, already mentioned here, is a complete office suite that includes a word processor, spreadsheet, presentation tool, database and drawing tool. Its look and feel is very similar to that of Microsoft Office and it will read and write several versions of MS office files, sometimes better than Microsoft Office itself. In addition to the features one would expect from an office suite, OpenOffice.org also can save any document as a PDF. NeoOffice is a version of OpenOffice.org for Mac OS X.

AbiWord is a small word processor. It is not as fully-featured as OpenOffice's Writer, but takes little disk space and will run on antiquated software. It will also read and write Microsoft Word documents, making it a good solution for places where older computers are in use.

PDFCreator is a handy Windows-only utility that installs as a printer driver. It allows any program that can print to create a PDF.

The GNU Image Manipulation Program (The GIMP) is an unfortunately-named photo editor. It is similar to Photoshop in that it has layers and filters and can be difficult for a novice to figure out. Like Photoshop, it offers many features that most people will never need.

NVU is an HTML editor that is based on code from Mozilla. One of its best features is that it has a very simple tool for managing a remote web site. For the kinds of web pages that most K-12 teachers and students need to create, NVU is more than adequate. Like other HTML editors, it has multiple views, allowing users to edit HTML by hand or use

word processor-like commands to affect the same results.

A great server-based OSS tool is Moodle. An often-used technology to support teaching and learning is course management systems like WebCT and Blackboard. Moodle provides an alternative that requires less training for the teacher, the student and the technology support department (Pfaffman, 2005). Though Moodle is designed as a way to organize a course where everyone has access to the web, it is also a great way for K-12 teachers to create web sites for their courses, letting students, parents, and other teachers know what is going on in their classes. Because it is not only free but also requires little support, my university is able to provide any school in the state their own Moodle site. Because Moodle is easy to run and will work on Windows, Mac OS X and GNU/Linux servers, some districts have opted to run it on their own servers.

References

- Byfield, B. (2002). Opening up to openoffice.org: Finding an alternative to microsoft word. *TECHWR-L*. ([Online.] Available: <http://www.raycomm.com/techwhirl/magazine/technical/openofficewriter.html>. Accessed 2004-08-04.)
- Delio, M. (2004). *Spyware on my machine? so what?* Retrieved 2/10/2006 from <http://www.wired.com/news/technology/0,65906-0.html>.
- Graham, P. (2005). *What business can learn from open source*. [Online] Available: <http://www.paulgraham.com/opensource.html>. Accessed 2/10/2006.
- Leadbeater, C., & Miller, P. (2004). *The Pro-Am revolution: How enthusiasts are changing our economy and society*. London: Demos.

Pfaffman, J. A. (2005). Open source solutions: Moodle. *Learning and Leading with Technology*, 33(2), 42-45.

Stallman, R. (1985, March). The GNU manifesto. *Dr. Dobb's Journal of Software Tools*, 10(3), 30-?? [Online.] Available: <http://www.gnu.org/gnu/manifesto.html>.

Zymaris, C. (2003, October 21). *A comparison of the GPL and the Microsoft EULA*. Retrieved November 21, 2004 from http://voidmain.is-a-geek.net/docs/comparing_the_gpl_to_eula.html.