

Benefits of open-source for Education

Table of Contents

Learning Concepts vs. Training for Products.....	1
Lower Total Cost of Ownership (TCO).....	1
Lower Costs for Students.....	3
Freedom from Vendor-dictated Changes	3
Freedom from Lock-in.....	3
Customizations	4
Community.....	4
Extending the Life of Older Hardware.....	5

The benefits of open-source software to teachers, students and places of education are many. However, using open-source software in schools is not a silver bullet - it will not solve all problems, technological or otherwise. Decision-makers must examine how open-source can fit their specific needs and the culture of their specific organization. That said, open-source has an abundant potential to impact the strategies of educational institutions. It can provide solutions to many pressing issues in school technology and beyond. With open-source software, your school can take control of its computer resources and manage its IT future. Some of the biggest benefits are described as follows:

Learning Concepts vs. Training for Products

Instead of teaching students concepts and techniques, instructors frequently teach the use of a specific product. For example, an instructor may focus on Photoshop CS6's Curves tool rather than a more conceptual understanding of color channels and non-linear luminosity adjustments (which a "curves tool" adjusts in many image editors). Instead of teaching students the basics of good writing structure for essays, teachers often end up teaching students how to layout their essays using a particular word processor. These lessons may be useful, but they fail to empower the student with foundational knowledge that transcends software, and they create dependencies on specific software to validate the teaching and learning.

In his 2005 essay on Why FOSS in Education Makes Sense, P. Tellis states that "Popularity begets obsolescence". He notes that teaching specific tools, popular or not, leads to obsolescence when those tools cease to be in use. No one uses Word-star, Lotus 123 or D Base today, yet these were the tools once taught in schools. "What should be the purpose of computer education?"

- Teach students to *learn* any tool
- Let students learn through hands on experience
- Throw responsibility into the hands of students"

The idea should be to teach students concepts, expose them to a variety of tools, and the choice of specific tools should be theirs.

Unfortunately, propriety software has become the default standard in many cases. People are unwilling to learn alternative products because it's not what they believe they'll need to know in business. Businesses are unwilling to change to alternative products because they'll face retraining costs or are unfamiliar with the options. In schools, this reinforces the tendency to teach specific products rather than teaching students the basic concepts behind the use of those products.

Since open source software doesn't have the tendency to change file formats or become incompatible with itself, teaching concepts is easy. A popular argument against using open-source is that students

will be entering a Microsoft-dominated workforce and they need to learn Microsoft's programs. In reality, Microsoft Word has changed so many times since its inception that teachers constantly need to change the way they teach it. All software evolves, students will have to learn new user interfaces and features; whether transitioning to proprietary software upgrades (Photoshop CS5 to CS6) or between two different applications (LibreOffice Writer to Microsoft Word). With conceptual knowledge, the transition is easy in both cases. Access to a wide variety of open-source software products allows teachers more choices to illustrate their lessons and provides students greater exposure to how other applications may perform similar tasks.

Lower Total Cost of Ownership (TCO)

One of the most obvious ways open-source can help education is by saving money. Total cost of ownership, or TCO, refers to the complete cost of any solution. All software carries costs; whether they come in the form of licensing, administrative effort, training or support provision. With any operating system and any set of applications, someone must be responsible for maintaining systems and applying security and product updates. Someone will have to be responsible for fixing things when they go wrong, regardless of the reason. These costs are common to both proprietary and open-source software. Licensing fees, however, generally apply only to proprietary software, and this can represent a significant cost savings.

Let's use a common example to illustrate. One of the most widely used applications on school computers is the office suite. The most popular is Microsoft Office (MSO) and many schools have licensed this program at a significant cost. As of July 2013, Microsoft offers a license for student or home use at \$139.99. Schools get educational discounts, but pricing is not readily and publicly available. As an example, University of Hawaii charges \$48.00 to departments for a standard MSO license. For a school with 50 staff and 100 lab computers, the license cost would be \$7,200 at \$48 per license. If that school serves 500 students, their families would incur \$70,000 to license one copy for home use – which is pretty much expected if you're teaching with that software at school. This example is for a small school! If a larger school with similar staff/student ratio serves 5,000 students, the cost to schools and families becomes \$72,000 and \$700,000 respectively!

There are several good open-source stand-ins for MSO. The most popular is LibreOffice – sponsored by a nonprofit called the Document Foundation. It is a full-featured office suite including software to word process, make spreadsheets and presentations, create drawings, and even provides support for databases. LibreOffice supports dozens of file types, including Microsoft's. There are other stand-ins for many, if not most, proprietary titles commonly used in schools.

Now consider all the other proprietary software uses in schools; licensing represents an enormous cost burden. Note also that these costs recur every time a new version is released, the expense only covers a few years. Also, these licenses prohibit commercial and other uses – commercial usage rights are even more expensive. If students and their families don't understand this, they could be using the software illegally for unlicensed purposes.

Elimination of license tracking represents a second savings area in TCO. License tracking represents a burdensome investment of time, and potentially financial penalties. This has brought many headaches to technology coordinators world-wide. Being prepared for an audit requires record-keeping and careful diligence. It is generally easy to install a copy of software onto many computers regardless if it is licensed only to one, and it is very tempting for staff and faculty to do this on their own. This is illegal and can cause serious problems for schools. In 2001, under threat of lawsuit, Microsoft audited schools in 35 states to crack down on unlicensed software. In Philadelphia alone, over 264 schools were forced to conduct an audit to prove that every piece of installed Microsoft software had a valid license. Schools could either suffer the audit (an expensive and time-consuming task) or pay \$40 for every computer in the school. With open-source software, you

are free to copy and distribute freely; no time and effort is wasted on audits, no record-keeping is required, and no legal penalties await.

A third means of lowering TCO is by maximizing existing resources. Open-source software typically has lower hardware requirements than proprietary alternatives. Because of lower hardware requirements, Linux-based operating systems can run well on older hardware. This extends the life and usefulness of computers already in schools. Even older donated computers can be put to productive use. As an example, an updated base installation of Windows 7 uses over 25 Gigabytes of hard disk space, and this is before installing antivirus or office software. An updated base installation of Ubuntu Linux, including LibreOffice, requires only 4.5 Gigabytes. Software with lower hardware requirements will run faster on computers running similar proprietary software. Lower hardware requirements also strengthen investments in new hardware - you get more for your money if a computer you buy today can stay operational and useful for longer.

Replacing a proprietary application (or all of them) with open-source can reduce the total cost of ownership (TCO) by enormous figures. Excellent resources to compare options are schoolforge.net, focused on educational use of open-source; or sourceforge.com which provides information and reviews for the full range of open-source software.

Lower Costs for Students

Unfortunately, not every student has access to a high-end computer running the latest software at home. Schools often get discounts on software but these savings are lost to home users. As schools increasingly include computer skills in their curriculum and request parents to provide computers in the home environment, students from lower income families become increasingly disadvantaged. With open-source, the school can send software home with students or provide instructions for home download and installation. As we saw in the TCO example, a school's decision to use open-source software can save families a lot of money.

Fortunately, many open-source software products run on lower end machines. Since files are saved in open formats, it is possible to move files between different versions of many products (and even between different products in some cases). For instance, most open-source word processing products can also read and save files in the Microsoft Word format, the GNU Image Manipulation Program (GIMP) can read Photoshop's format.

Parents often make decisions about what computer to buy from the type their child uses in school. Choosing to run open-source software, where possible, on school computers allows families to save money by allowing interoperability with open-source software on home computers. Choosing open-source software for school computers allows families to choose less expensive hardware and removes software costs for home computing. In short, choosing open-source software for school computers levels the playing field for all students.

Freedom from Vendor-dictated Changes

In many cases, proprietary software vendors deliberately construct upgrade cycles aimed at maximizing their profits. Sometimes an upgrade introduces desirable new features, but sometimes new features amount to bloat. Worse, unnecessary user interface changes cause confusion and require additional training and support efforts.

Worse yet, vendors sometimes introduce file format incompatibilities that pressure users to upgrade in order to inter-operate with others. Microsoft has been infamous for changing the file formats of its Office software every couple of years to force buying the new version of its software. With open formats, even if LibreOffice becomes obsolete (not likely to happen), you could still use these files and not worry about them becoming lost to changing file formats.

Regardless of the nature of upgrades, using open-source software leaves the individual or institution in control of the upgrade decision. There is no licensing cost associated with an upgrade cycle, so staying current is cost-free. There is no incentive in the open-source development model for introduction of bloat, desirable features are vetted through the community before implementation. Open-source software uses open formats (frequently also supporting proprietary formats where feasible), which maintains interoperability with older versions and other users.

Freedom from Lock-in

Vendor lock-in is created when the cost of moving to different software is prohibitively expensive or even impossible. Consider two scenarios. In the first, a school chooses a proprietary learning management system (LMS). The faculty invest hundreds or thousands of hours in learning the system, developing course materials, and gaining proficiency teaching with the software. Support staff similarly invest time and energy in using this system. This investment puts the school into a committed position. The vendor can (and frequently does) begin increasing the license price because they now have power over the school – the increasing license prices are less expensive and less politically unpopular than moving the faculty to an alternate LMS.

In a second scenario, the use of proprietary data formats can make it nearly impossible to use other software or move to alternate systems. For example; committing to a proprietary database, and therefore a proprietary data format, may make it impossible to tie future applications to that data. Lock-in leaves both individuals and institutions unable to make related decisions that best meet their needs. This is especially important to institutions, because one technology-related decision can negatively impact the range of future options.

With open-source, vendors are generally selling service and support – not licenses. Support and service can frequently be obtained from more than one organization, creating competitive pricing and further avoiding vendor reliance.

In fairness, it is common for proprietary software to have newer or more advanced features than open-source alternatives. On the other hand, open-source can also be the front-runner – much depends on the size of the community behind an open-source project. The trade off for choosing proprietary software for the latest features, however, is frequently vendor lock-in.

Customizations

Software is usually sold (licensed) on an as-is basis. If it doesn't quite do everything you need it to, then either you bought the wrong software or you have to change your requirements – the software dictates how you operate. With proprietary software, if you're unhappy with software for any reason there's typically nothing you can do to make it better.

Open-source software changes this. "Open-source" means you have access to the source code of the software. You have permission to change the software, add new features, or take features away. And if you, like most users, don't have the skills to change the code yourself, you are free to hire others to do it. You can install your the customizations wherever you need them. In fact, so long as you comply to some fairly simple license requirements, you even have permission to spread your customized version far and wide. In some cases, changing the software may not be practical, but in many cases whole systems can be put together in ways that match the specific requirements of your school just by gluing together existing open-source packages.

As an example, [Open Admin](#) is an open-source option for school administration. You can choose to run the software free of charge, or you could hire the developer for support, or even contract him to write customizations that meet your needs. See Support and Development options at

<http://casestudy.seul.org/openadmin/commercial.html>

Flexibility is another advantage of open-source. Since anyone can modify the source code, programmers can port a program from one platform to another. Programs like LibreOffice, FireFox and GIMP run on Linux, Windows, and Mac OSX. This allows use of the same software on all types of computers. Often, schools have a mix several operating systems, making this type of versatility very important. With proprietary software, the power to bring a piece of software to a new platform relies solely on the company - users have no say in the matter.

Community

Open-source use opens up the possibility of collaborations with other departments, schools, or districts. For instance, if there are customizations desired – the costs could be split among multiple sponsors. Because the customizations are freely distributable, every sponsor enjoys the results of every customization. This is a primary reason that even competing businesses collaborate on open-source projects – it drives everyone's costs down.

Additionally, worldwide communities of users and developers are enabled around open-source projects. If you have questions or problems, it is highly likely that you can find help from fellow users in on-line forums, chat channels, wikis and other user-generated documentation. Two great examples from the worldwide community of Ubuntu Linux users are askubuntu.com and ubuntuforums.org – both are extremely active places where everyday people ask and answer questions to help each other enjoy Ubuntu Linux productively. Such community-based resources are valuable assets to any technology coordinator.

Extending the Life of Older Hardware

In discussing TCO, we noted that many open-source packages can run ably on older machines. Schools rarely have the money to buy the latest hardware, and software upgrades frequently require hardware upgrades in order to perform reasonably. Open-source Linux-based operating systems are “lightweight” run well on systems that would be obsolete for use with Windows or OSX. Not only does this make good use of older hardware, it slows the procurement cycle – which reduces electronic waste.

The Linux Terminal Server Project (LTSP) is an interesting project that side-steps hardware obsolescence by turning aging computers into disk-less workstations which boot from a network server. By using LTSP, a school can quickly increase the number of available workstations for students without a large expenditure on hardware or software licensing. One fast machines can be configured as a server for an entire computer lab. LTSP lowers the cost of administering the machines too. Instead of maintaining multiple computers, only the server needs administrative updates and monitoring.

Edited and authored in part by Paul McKimmy, 6/2013

This work is a partial derivation of Timothy Hart's (timothy.hart@gmail.com) open-source in Education, version 1.0 dated February, 2004. It incorporates content that is updated and adapted from Benefits of open-source in Education, <http://www.osv.org.au/index.cgi?tid=5>

References added in 6/2013 revision

Cave, D. (2001) Microsoft to schools: Give us your lunch money!, Retrieved 6/30/13 from http://www.salon.com/2001/07/10/microsoft_school/

Tellis, P. (2005). Why Foss in Education makes sense. Retrieved 7/1/2013 from <http://tech.bluesmoon.info/2005/11/why-foss-in-education-makes-sense.html>

In compliance with Timorthy Hart's license, this work is licensed under Creative Commons Attribution-Noncommercial-ShareAlike version 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0>