Volume 6 – Spring 2010
djim.management.dal.ca

The Myth of Free: The Hidden Costs of Open Source Software

Abstract: Open source software (OSS) has become increasingly visible in recent years as a viable option for library automation. Many libraries rely on an integrated library system (ILS) to perform automation and library functions. As this is expensive software, many libraries are interested in open source ILS options, which provide the source code to anyone who requests it free of charge. This report assesses the costs of choosing an open source integrated library system over a proprietary integrated library system. It examines the human, time and monetary costs of choosing to work with an OSS ILS, in the face of claims that OSS is "free". OSS ILS's provide their users and contributors with the freedom to make the software into anything and everything a library desires. But open source ILS options certainly aren't completely costless.

About the Author(s): Amanda Foote is an MLIS candidate at Dalhousie's School of Information Management. She is originally from St. John's, NL and hopes to return home to relevant work upon completion of her degree. While at Dalhousie, Amanda has become extremely interested in the technological side of librarianship, especially regarding information systems and software use in the library. This paper was originally developed for INFO 5590: Information Management Systems at Dalhousie University. Amanda currently works as a teaching assistant for INFO 6620: Electronic Text Design

Introduction

Open source software (OSS) is a relatively new and attractive option for any institution that focuses on patron or client service. Making the change to open source can bring any number of benefits to a library, but the cost often attracts the most attention. Software, in its most basic form, is source code. The source code is what determines the features and functions of a software program. The source code, in an open source context, is provided free of charge upon request. The library community is particularly receptive to the idea of free, especially in the face of shrinking budgets, rising material costs, and inflation. Over the course of this report, I will examine the open source form of one particular piece of software that is often heavily relied upon in libraries: the integrated library system, or ILS. The integrated library system is an interesting example, as it can be put to effective use in any number of records management situations, from law offices to small, specialty libraries. The ILS is also rapidly becoming a necessity as institutions grow their collections and records. The traditional ILS is proprietary, provided by a for-profit vendor, and is an extremely costly piece of software in terms of both initial purchase price and maintenance. On the surface, choosing an open source integrated library system over a proprietary integrated library system eliminates these costs (Schultz-Jones, 2004). This suggests that an open source ILS demands little to no financial commitment. In actuality, there are any number of "hidden" costs associated with the creation or adoption of an open source integrated library system, which will be assessed throughout the course of this report. Despite the lack of a purchase price, a library's choice to migrate to, or create a new, open source integrated library system comes with human, time and monetary costs. This report assesses these costs in regard to migration and maintenance and illustrates how an open source integrated library system may actually be more financially demanding, depending on an institution's individual circumstances and abilities.

It is safe to say that most modern libraries would be hard pressed to operate efficiently and achieve growth without an integrated library system, and these ILS require a significant staffing commitment. Tracy Primich and Caroline Richardson (2006) detail the history of the ILS in *Integrating Print and Digital Resources in Library Collections*. While their history includes an examination of the money spent by libraries on their integrated library system, they also note that "keeping up with software upgrades, equipment maintenance, and general system maintenance, such as regular backups and regular monitoring of the system, are ongoing staffing requirements" (Primich & Richardson, 2006, 122). To ensure patron satisfaction, a library's ILS must run efficiently, and be easily fixed when problems arise. Although a proprietary system is maintained and corrected by a vendor, this is not always an ideal situation. This reliance can lead to long wait times on repairs (Breeding, 2009). Some of the benefits of OSS are directly related to time and efficiency. With access to the software code, librarians can customize the ILS to their library's specific needs, as well as fix problems as soon as they arise. With this level of access, open source allows for much more freedom in regard to how the ILS functions in the library. The tricky problem of "free" is perhaps no better illustrated than with the above point. Open source is certainly free in the sense of the library's freedom to choose what features are needed, but it is not financially free (Weber, 2004).

Methodology

To support the arguments made in this report, a thorough literature review of both trade publications and academic articles was conducted. Trade publications examined focus on the library environment, or more specifically the example of the integrated library system within an open source context. Academic articles also focus on the role of ILS and the library, as the ILS is used as the main example in this report. In an attempt to examine the issue outside of the library environment, as well as to show that the hidden costs of open source software are not limited to the ILS example, a number of books that focus generally on open source were consulted. Although these sources may or may not mention the role of an ILS in the library, their comments are applicable in the sense that the open source community operates in much the same manner, despite the nature and name of individual projects. Independent examples of particular open source systems are not thoroughly examined in this report for two reasons: firstly, as the open source ILS technology is relatively new and has not been widely adopted in the library environment, analysis or studies of the success or failures of a particular library with these systems are generally not available. Secondly, the intent of this report is to show that the ILS example can be representative of costs of open source software projects as a whole. The information given is intended to provide readers with the background to make an informed choice for their own particular situations. As each situation is different, I felt it more appropriate to leave out specific library examples.

Literature Review

Various open source software options have been previously discussed in the related literature, and the cost of OSS projects is often a point of difference. On one side, librarians and authors argue that open source, generally speaking, is a dismissal of traditional, expensive proprietary systems. In his book, *Here Comes Everybody,* Clay Shirky (2008) argues that "the open source movement...doesn't have employees, it doesn't make investments, it doesn't even make decisions. It is not an organization; it is an ecosystem, and one that is remarkably tolerant of failure" (p. 246). He also argues that this "tolerance of failure" lowers costs in the long run, as significant money is rarely spent on OSS projects (Shirky, 2008). More specifically, some authors argue that the time and human costs of choosing an OSS product are also lower. Consider the ILS example. As the open source integrated library system code would rarely be kept completely unique to one particular library, users are often eager to share code and help others fix mistakes. Open source software also often erases the need for "duplication of effort," meaning once one person has committed to fixing a mistake or problem with the software, it is shared amongst the community (Deek & McHugh, 2008). Moreno Muffatto (2006), in his book: *Open Source: A Multidisciplinary Approach*, acknowledges that the initial costs of open source software can be very high in terms of money, time and effort. He counters that those copies of the created system are virtually costless.

While these authors raise valid points, there is some contention around the implementation and use of open source software in libraries. Some argue that the very nature of open source software allows an integrated library system to be changed to fit the specifics of any library, at little to no initial cost. These authors also state that support and help exist throughout the OSS community as a whole, and hence there is no need to pay a vendor or company for a product that cannot be modified. However, as we will see, other authors argue that open source creates costs outside of purchase price by requiring its users to solve their own problems and make their own changes. While both sides are technically correct, all of these issues must be examined to determine what the true costs of running an open source integrated library system are, in regard not only to purchase and upkeep prices, but also in regard to human and time costs.

OSS and Proprietary Systems: Initial Costs

To place the following discussion in context, it is important to examine the upfront costs of proprietary and open source integrated library systems. As mentioned above, the proprietary ILS is often one of the most expensive parts of running a library. Linda Riewe (2009) provides hard data about the costs of open source ILS, using the example of Koha, one of the major pre-existing OSS options available: “A Koha consulting firm in the United States estimated the cost for a library of 15,000 items at $10,700 including hardware and the first year of support. Thereafter, the annual support cost would be $2,500” (p. 53). These numbers are very attractive, especially considering a comparable proprietary system may be three or four times the cost. For an average library consortium using the Unicorn ILS, made by vendor SirsiDynix, and running on a Sun Microsystems server, the server would cost one million dollars initially, and the annual cost would probably fall around $558,000 (USD) (Riewe, 2009). On the surface, this appears to prove that an open source choice is the more affordable option. The next three sections examine how OSS can end up costing quite a bit more, even without a high initial purchase price.

Human Costs

Any integrated library system, whether it is proprietary or open source, requires a significant human commitment. Consider the library that chooses to create its own unique open source ILS. In the beginning, the source code for the software or program must be thought through, evaluated, re-evaluated and written. This alone could represent a huge commitment of staff, who would each fill certain specific roles; however, this is the rare case. Many libraries will choose to work with a pre-existing open source ILS because project creation would simply require too large a staffing commitment. OSS projects are almost always group-oriented with many people sharing the work and burden of creation. As noted by Riewe (2009), “the availability of OSS in libraries today is due to the contributions of librarians, programmers, volunteers, and sponsoring libraries, many of which are funded by the public" (p. 15). Although OSS projects can be undertaken for a variety of reasons, it is safe to assume that those who choose to participate have some degree of specific technical knowledge or skill, which they wish to use to its best advantage. Their time is valuable to a library, and may be involved in a number of projects outside of the ILS. Even when working with an existing option, significant energy must be spent for it to be improved and maintained. There are human costs of the project outside of the library as well. OSS relies heavily on a community of contributors to improve functionality; so many people outside of the library may be engaged in the project. Without this pool of people, the project may never amount to anything significant (Sowe et al, 2008). A library may not have previously considered these commitment costs, and the need to involve parties outside of the library staff. As a proprietary system provides any help and solutions needed, an institution may not have previously needed to commit any resources at all to problems. The same is obviously not true in an open source context. An OSS project relies heavily on a system of people to keep it working correctly. But there is no guarantee that a library may or may not become "stuck" on a problem, with no solution readily available through the online community.

This should not encourage a library to disregard open source communities. Outside contributors may choose to participate for a variety of reasons. These skilled contributors may choose to participate in OSS projects for the "reputation capital" that accompanies their successful implementation (Muffatto, 2006). Work on an open source project can be a chance for programmers to add to their resume, or show their unique skills. Library staff may work on outside projects for the reputation capital, or they may do so simply because it is part of their job. Outside organizations also recognize the value of human commitment to OSS. An OSS community provides a reservoir of experienced and productive workers, from which a company may hire (Muffatto, 2006). The fear that outside companies will poach staff should not be considered too great for a library to take though. Instead, it should highlight the value of participation in communities. All of this effort can add up to a significant human investment, which may only grow as contributors take more and more pride in their work.

Time Costs

It is fair to say that human and time costs are closely related when it comes to OSS projects. When choosing a pre-existing option, time must initially be spent deciding on an open source ILS, which may mean extended staff meetings and surveys. Once a project is decided upon, more time must be spent in planning how the project will take shape. Library staff, for example, would need to devote some of their paid working time to the integrated library system project in terms of creating new features and working on existing functions. This may detract from the amount of time spent on other projects or workflows. Frank Cervonne (2007) comments on ILS migration in his article: "ILS migration in the 20th Century: Some New Things to Think About

This Time Around." He notes that any software change has a time cost, as new staff members have to be trained, and the new system has to be prepared and tested for entry into the library. Migration time requirements alone may stretch the capabilities of an IT department beyond its abilities if the project is not carefully planned or considered. Planning the scope of an open source project can also be very time consuming, as it may require the library to plan years into the future. For example, there is a danger of duplication of effort if the team is not in constant communication, whether at work, or through an online community (Breeding, 2008).

Although, as discussed in the next section, libraries sometimes have the option to pay for additional support of their OSS, most often the burden of creating and maintaining specialized features will fall solely on the shoulders of the team responsible for development. Deek (2008) argues that it "has been estimated that 75% of code is written for specific organizational tasks and not shared or publicly distributed for reuse" (p. 8). So while the library can depend on an online community for support, they cannot rely on the community to create unique functions. Time must also be spent in planning who will be responsible for fixing the system once problems arise. And finally, time must be spent on revamping policies in regard to OSS development (Primich & Richardson, 2003). Is the library willing to share its specific innovations and code with the OSS community? Do they need to make this decision on a case by case basis? Open source communities grow through participant's willingness to share features and functions, and can only continue to grow through migration of programmers between various projects (Sowe, et al, 2008). Participating in this community is also a time cost for the library, as it should not be expected that a library can take from the community and give nothing back.

The true freedom of an OSS project is that it allows for control of the functionality and features of the software. It would be a waste of opportunity for a library to undertake a project without having a working plan of what form the system will take in the future. Human and time costs, as illustrated in the previous two sections, can seem greatly expensive even when not considering any further monetary costs, but these costs must be considered. In the next section, I will examine how these human and time costs can translate in actual monetary concerns.

Monetary Costs

The underlying truth of both human and time costs, when considered in this context, are that they translate to financial costs somewhere along the way. Time and energy translates into a financial cost for the library simply by committing paid staff to the project in the first place. An OSS project also relies on "vigour" to keep it viable. This "vigour" is defined as the specific capital contributions attributed to the project (Sowe et al, 2008). It would be naive to think that a library would be able to run an efficient open source ILS without at least some of the library's budget allotted to prepare for mistakes, problems or maintenance. Questions are also raised: is the library willing to allot overtime pay for creating new features? Is the library willing to commit further financial resources if the project grows larger than a single library? Is the library willing to pay for outside consultation to solve difficult problems? All of these questions must be considered in the planning phase but also in the budget.

Another cost may arise from server issues within the library. A library must make sure that they have adequate server size and capabilities to support an open source system that may require constant fine tuning and change. In the case where a library does not have these capabilities, they may have to then rely on outside servers from a business or vendor to provide the space and functionality required. This space can be either rented or bought, depending on the company and amount needed. In case of renting server space, this adds another monthly bill to a library that was hoping to cut down on its bills in the first place (Breeding, 2008). Buying server space could cause a significant dent in a library's budget.

Many of the pre-existing open source integrated library systems, such as Koha, offer an option for a library to pay for additional specialized support for the software, when a solution is not readily available. Ruth Carlock (2008) points out that it is important to recognize that while the source code might come free of charge, support may come at a price. While these costs are often significantly less than reliance on support from a proprietary company, they can still amount to a significant total if they are combined with other bills, such as one for monthly server space. Finally, it is also important to recognize that an open source ILS may not have been a viable choice, even after all that effort and commitment (Pace, 2003). There is always the danger of the new ILS being rejected by the library as being inefficient or as not being sufficient to meet the library's demands. It is entirely possible that a library may migrate only to have to change back to a proprietary system in a few years, which can create huge costs all over again. In this case, the library has lost all of the financial support it has thrown into the project through staffing and server requirements. It is important that a library examine its budget and ask hard questions about whether or not they can afford to take a risk.

Conclusions

All of the above considerations should not be taken to mean that an OSS project is not a worthwhile undertaking. The most important part of any OSS project is that a careful plan be implemented along with the project. Libraries that choose the open source ILS option must be prepared to commit significant staff, time and money in order to take full advantage of the benefits OSS offers. Libraries stand on the edge of being innovators in the library field, depending on how they choose to implement, use and grow open source software. The possibilities for ILS transformation are virtually limitless, given enough time, energy and creativity. This creativity and energy can be bolstered by commitment to a project and its online community of similar users. But this should not lead a library to believe that this community is a limitless resource that can be taken advantage of indefinitely. Libraries must be prepared to absorb the costs of paid staff time, the possibility of needing to pay for outside help, or the possibility that specialized features may require additional server space. Then they must be prepared to share innovations and help other users solve problems, in order for the software to achieve its true potential. Open source software is freedom – the freedom to transform a library’s ILS into anything staff feels is helpful and appropriate, but it costs much more than its small purchase price.

References

Breeding, M. (2008). Making a business case for open source ILS. *Computers in Libraries*, *28* (3).

Breeding, M. (2008). The viability of open source ILS. *Bulletin of the American Society for Information Science & Technology*, *35* (2).

Carlock, R. (2008). Open source Integrated Library Systems. *Nebraska Library Association Quarterly*, *39* (4).

Cervone, F. (2007). ILS migration in the 21st Century: Some New Things to Think About This Time Around. *Computers in Libraries, 27* (7).

Deek, F. M. & McHugh, J. A. M. (2008). *Open Source: Technology and Policy.* Cambridge: Cambridge University Press.

Muffatto, M. (2006). *Open Source: A Multidisciplinary Approach*. London: Imperial College Press.

Pace, A. K. (2003). *The Ultimate Digital Library: Where the New Information Players Meet*. Chicago: American Library Association.

Primach, T. & Richardson, C. (2003). The Integrated Library System: From Innovation to Relegation to Innovation Again. *Integrating Print and Digital Resources in Library Collections*. Binghamton: The Haworth Information press.

Riewe, L. (2009). Survey of open source integrated library systems. M.L.I.S. dissertation, San Jose State University, United States – California, Retrieved November 1, 2009, from Dissertations & Theses: Full Text (Publication No. AAT 1459712).

Schultz-Jones, B. (2006). *An Automation Primer for School Library Media Centers and Small Libraries.* Worthington: Linworth Publishing, Inc.

Shirky, C. (2008). *Here Comes Everybody: The Power of Organizing without Organizations*. New York: The Penguin Press.

Sowe, S. K., Stamelos, I. G. & Samoladas, I. (2008). *Emerging Free and Open Source Software Practice.* Hershey: IGI Publishing.

Weber, S. (2004). *The Success of Open Source.* Cambridge: Harvard University Press.