Running head: OSS USE AND DEVELOPMENT IN LIBRARIES

Lessons from the Bazaar: Open Source Software Use and Development in Libraries

Mary Ellen Petrich

November 5, 2009

Seminar in Information Science
Topic: The Open Movement and Libraries

LIBR 287-05 – Fall 2009

School of Library and Information Science – San Jose State University

Abstract

This paper discusses how libraries can and should participate in existing OSS development communities, and why libraries may want to start their own OSS projects. When libraries use OSS software, and fail to participate in the user community, the library risks of losing sight of the project direction, which may ultimately make the product unsuitable for that specific library. While library staff may not be able to participate as programmers, other types of participation can assist the community with providing documentation such as software evaluation and best practices for implementing the software. Participating in OSS communities can educate library staff in the practical skills needed to develop in-house software.

Lessons from the Bazaar: Open Source Software Use and Development in Libraries

Open source software has become more common in the library and in everyday life. Two of the most successful open source software projects are the GNU/Linux operating system and the Apache web server. Apache (http://www.apache.org) is used by more domains than all the other web servers combined, and has been since 1999 ("October 2009 Web Server Survey", 2009). More familiar to most people is the Firefox web browser (http://www.mozilla.com) which in 2008 had about a quarter of the browser market worldwide, but as much as half the market in some European countries (Lilly & Beltzner, 2009, min. 12-15). One that should be better known to the general public is OpenOffice.org (http://www.openoffice.org) which just announced the 100,000,000th download on October 28th, 2009 (from the OpenOffice.org home page). Each of these open source software projects has large successful communities.

For libraries, major open source integrated library system (ILS) software projects are: Koha, Evergreen, and OPALS. Major open source online public access catalog (OPAC) projects are: Scriblio, VuFind, SOPAC, and eXtensible Catalog. Libraries and archives also use open source digital repository software such as DSpace, Fedora, and EPrints. In addition there are many other types of open source software that are used by libraries.

The concepts behind open source software are wildly popular with many people in the library field because the philosophy of free and open access to information tools is one which mirrors the philosophy of libraries to provide information freely. Ameral (2008) asserts, "Libraries and the open movement mesh together logically" (p. 6). Chudnov proclaims,

> "Use, study, modify, and copy" is an immensely powerful summary of both what free software is and why free software works. In my mind, the only more powerful statement of

> freedom apart from our founding documents is the statement emblazoned on hundreds of older libraries, many built nearly 100 years ago with Carnegie funds: "Free to All." (p. 23)

Many prominent library people are strong advocates for open source software. Karen Schneider (2008) (of freerangelibrarian.com) inspires with her statement that open source software enables librarians to be "leading innovators" (p. 45) who develop their own tools. Others, like Marshall Breeding, propose caution when speaking of the largest piece of software that any library is likely to own, the integrated library system (ILS): "As the open source movement matures, these systems will need to compete on their own merits and not solely on a philosophical preference" (Breeding, 2008). While open source software can be obtained free of charge, libraries still must evaluate, purchase, deploy, and maintain the software, and those services must be paid for either by employing staff in the library or by employing the services of a commercial vendor. Most recently Steven Abram, a highly visible speaker in the library field who is employed by proprietary software maker SirsiDynix, wrote an extremely negative position paper on open source software which has garnered considerable criticism, but has also generated some discussion about the strengths and weaknesses of the open source versus the proprietary model of distributing software (Hadro, 2009).

While there are a small number of super star open source projects, some studies suggest that the majority of open source projects are written primarily by one person and have not gathered a self-sustaining community (Krishnamurthy, 2002). Even the most vigorous advocates for open source software still use some proprietary products when necessary—Howard County Library, well known for using open source software on hundreds of public access computers uses some computers with Microsoft and Apple software for staff computers (DeGroff, 2008).

It is difficult to evaluate what percentage of libraries are using library specific open source software. A brief evaluation of literature may help shed some light. A literature survey by Houser (2009a) shows that the use of Linux as an operating system in libraries has expanded substantially between 1999 and 2009, but the types of articles cited indicate that using Linux is much more of an exception than a rule. In a related article, Houser (2009b) says, “While the topic may still be open to debate, there is clearly a contingent of library professionals who feel their user experience is better with open source software” (p. 7), wording which indicates that the contingent is not large. Breeding’s (2009) review of four open source discovery interfaces shows that there are a variety of open source products available to libraries. He declares that “without a doubt, the number of libraries using open source discovery interfaces is still much smaller than that of those using proprietary products.” But he goes on to say that the high quality of these products and their broad geographical acceptance leads him to believe that their popularity will increase rapidly. Breeding (2008) outlines a study that he performed which indicated a “relatively low level of interest in open source ILS options” (p. 36). Open source software relevant to libraries can also be classed as Web 2.0 software, as the open source model is used to develop and distribute more experimental types of software. A study on the availability of Web 2.0 tools on library web sites demonstrates that it is typically the largest, most popular, and best funded libraries that are able to experiment with new types of software (Lietzau, 2009). Based on this subjective information it does not seem that there are a large number of libraries that are using primarily open-source, library-purpose software.

Using open source software

For most libraries the primary perceived disadvantage of open source software is that the institution will need additional technical expertise, and additional staff time. Because open source software development tends to focus on features and the 'fun' process of software development, open source software can be more difficult to install, configure, and maintain. Schneider suggests that this goes with the territory: "to get what you want for your library needs, you need a product that has a higher implementation and maintenance curve" (p.46). Software should be evaluated point by point to ensure that it does get you what you need.

While some of the appeal of open source software to libraries is the perception of a common mission with libraries, there are other advantages, although they tend to favor more technical users. DeGroff (2008) points out that Linux does not attract the viruses that Microsoft operating systems do. Like Chudnov (2009), many writers extol the four virtues of being free to use, study, modify, and copy because of the ease to practically, safely, and legally test, use, and evaluate the software. Finally, many people use open source software just because it's the simplest and easiest thing to do. As Bisson (2007b) says, it's a "no-brainer."

While OSS is not free, because of the expense of the hardware and support, it does allow some flexibility for deciding what the cost is. Many open source software packages can be either used 'out-of-the-box', or substantially customized for the institution. For example, the University of Vermont is using DSpace with very little customization (http://badger.uvm.edu/dspace/). The user interface is the default. The logo for the DSpace organization remains in the upper left hand corner. The link to "About DSpace" goes to the DSpace web site, and the link to "Help" opens default help files. On the other hand the University of Illinois has extensively customized their

version of DSpace (https://www.ideals.illinois.edu/). The user interface is customized. The site has been rebranded from DSpace to IDEALS. The link to “About” goes to a page with extensive links to policies and procedures for using the site. And the link to “Help” opens extensive custom help files, with screen shots of the customized interface.

Ultimately, as Morgan (2009) says, using open source software means that the user has more control since the software is open and accessible. However, in addition, the user also takes on greater risk. Because open source software is so configurable there are more unknowns about whether it will be possible for an institution to use it as needed.

Participating in open source software communities

While most libraries considering open source software are aware that the maintenance costs may be greater than proprietary software, most consider only the benefits of a related open source community, and not the quality or commitment of the community. Understanding the strengths and weaknesses of the community should be part of evaluating an open source product. As Tansley, Smith, and Walker (2005) say, “a system which still needs work but has a bustling community around it is likely to be a better long-term bet than a more technically developed system with no visibly active community”(p. 252). Understanding how to interact with the community of an open source project is key to successful development, selection, deployment and maintenance of software installation. Ignoring the community means a significant increase in risk for open source product users. Van den Berg (2005) outlines a set of criteria against which open source software can be evaluated, including the qualities of the community, such as frequency of posts, number of users, quality of posts, and friendliness of respondents.

In her article, Griggs (2009) argues that libraries have to make an effort to find and evaluate open source software since such software rarely has the marketing materials most proprietary software does. She suggests that libraries should contribute back their product evaluations, even if the format is informal, so that future customers can benefit from the effort (p. 47). Hoppenbrouwers (2007) goes further by saying that open source product selection and assessment "might be significantly more expensive" (p. 515). In the open source model, these marketing costs are shifted to the customer, but can be spread out if multiple evaluators contribute information.

Hoppenbrouwers (2007) asserts that, while a user of open source software is not a customer of a vendor, that user is the customer of a community. He says, "a customer they are, with associated real costs to pay and real benefits to enjoy" (p.511). As a community customer, an organization has a responsibility to give back to the community. Most articles focus on community participation in the form of software development, but there are many other roles that a community customer can play, the simplest of which is helping other users and organizations on the community's discussion boards. Others include performing quality assurance, writing documentation, and translating.

While it is easy to say that community customers have an obligation to give back, it is unlikely that such a motivation will be enough to drive an institution to contribute. What other advantages does contributing return to the institution? Stam and Wendel de Joode (2007) speculate on the motivations of individuals and institutions contributing to an OSS community. The focus of their study is on the participation of commercial firms in open source communities. While the motivations of a library may not be to make money, there are many similarities in the

motivations of a library and a commercial firm in reducing costs and maximizing return on investment. Commercial firms, like libraries, “will only invest their time and effort in the communities if they believe it will lead to additional benefits or reduce costs” (Stam & Wendel de Joode, 2007, p. 498). The motivations of individuals are outlined briefly as developed in many other articles: building a reputation, improving programming skills, improving software functionality, and having fun.

For institutions, a danger of not participating in the community is that patches, releases and upgrades can come and go without notice. A problem could arise with the product, and without the goodwill of continued interaction, the responses of the community could be slower and less helpful. Without active involvement, an institution could find that product development has taken a turn which makes a valued tool unusable in future releases.

The advantages to participating in the community are many. Reading about the challenges of other software users educates and informs the IT staff of an organization, improving their understanding of the software. Active community members are more aware of the sometimes frequent software releases, upgrades, and patches of open source. Active participants have greater influence on product development. Organizations may participate in the community to ensure that mundane tasks are performed which individual contributors may not enjoy, such as documentation, localization, and packaging. Participating in OSS communities can educate libraries and library staff in the practical skills needed to develop in-house software. Some institutions may choose open source just because they can influence development, even taking part in bug fixes or improvements. On a larger scale, participating in the community keeps it active and vital. Without activity and interaction, “the development of new functionalities and ideas slows down and the

community begins to fade away" (Dahlander & Magnusson, 2008, p. 646). Asking questions and answering questions, making bug reports, and requests for functional improvements, contributing software evaluations, and best practices stirs the pot of the community and keeps it active.

Bonfield (2009) has written an excellent review of the major library specific open source projects. Some of the characteristics that he evaluates are whether the software was built from scratch, what language it uses for customization, how good the documentation is, and how stable the organization is based on current leadership, funding and the size and support of the community. Only Koha and Evergreen have communities that he considers sustainable. The other five projects that he evaluated: Blacklight, Kochief, Scriblio, SOPAC, and VUFind, are primarily driven by a lead developer or a sponsoring department that provides ongoing support to the project. Without that key individual, the project could disappear.

Producing open source software

When asked why they decided start development of an open source project, the first thing most librarians will list is the technical capabilities that they had in place. It's the equivalent of saying, "because we could." Looking back it is easy to see the advantages that they had that lead to success. A technically oriented institutional culture and forward thinking administrators are also cited as important factors in success. However the driving force is often a profound dissatisfaction with existing products.

At Oregon State University (OSU) the librarians found that their patrons were unable to use the vended federated search product—even the librarians were having trouble using it. They evaluated the available products and couldn't find one that did what they wanted, so in mid-2005 they decided to develop a product that became known as LibraryFind. The institutional culture at

the OSU libraries supported the project. Since the campus has an open source lab, the administration understood the open source philosophy. The library had a staff member, Jeremy Frumkin, whose role was to pursue innovative services and technologies, who could drive the project. They were able to get a grant to improve the software and share it with other institutions. In January of 2007, it was released to the world as open source software. Frumkin explains that developing LibraryFind was not the cheap way to go, but rather it was the better solution for the library. In his opinion there are times when proprietary software will be the better solution for a library and other times when using available open source software or developing new software will be the better solution. “We felt the return on our investment versus the cost of building the software was greater than our return on investment vs. the cost of purchasing the available vended products” (Bisson, 2007a, p. 38).

While the project members had hoped to develop a community as a sustaining force for the project, the community for LibraryFind has been small. The mailing list is active as of this writing and averages around 15 messages per month, mainly from users installing or configuring the software. Even these contributions assist the project in evaluating problems and bugs in the software.

At the same time that Jeremy Frumkin was working on LibraryFind, Kim Griggs, the lead programmer for the OSU libraries, began a project to create a content management system for the libraries’ web site called Library à la Carte (Griggs, 2009). The software was intended to assist librarians in creating interactive course assignment pages and subject research guides, without knowing how to write code. It was released as open source software in December of 2007, and the

project was moved to RubyForge to improve community interaction. As of this writing, the community is still active, and averages about 10 messages per month, mainly from users.

The Simon Frasier University (SFU) library was a relatively early entrant to the library specific OSS field. The reSearcher project started in 1992, and was funded in 1997 by a consortium of Canadian academic libraries (Stranack, 2007). One project to produce a link resolving program, grew into four including dbWiz, a federated search product. The projects were developed in concert with other regional academic libraries. A community of users collected around the products. Stranack outlines the opportunities and advantages of community involvement, concluding that participation in an open source project "demonstrates to administrators and other stakeholders that their libraries are innovative and taking leadership positions in the critical area of information technology" (p. 22). Currently, the community for reSearcher seems to be winding down. While there was a lot of activity on the discussion boards in 2006, as of this writing there are just a couple of posts per month (http://lib-forums.lib.sfu.ca/viewforum.php?f=1).

In 2005, the SFU library became a partner in the Public Knowledge Project (PKP). This project was started by John Willinsky in 1998 while he was at the University of British Columbia. Like the reSearcher project, this project also started with one open source project--software for managing and publishing articles for open access journals--and grew into four related projects. As of this writing, the community boards for PKP are active, so much so that it is difficult to estimate the number of posts per month (http://pkp.sfu.ca/support/forum/).

It often seems that academic libraries are able to lead the way in implementing new technologies and processes. Academic libraries often have more resources and the academic culture is more likely to encourage staff education and experimentation. But several public

libraries have also been able to collaborate to use and produce open source software. The Georgia Public Library Service (GPLS) developed Evergreen, which is open-source consortial library software (Molyneux, 2009). Howard Library is well known for moving all their patron computers completely to open source software. In the process they even went so far as to create their own branch of Linux, which eventually proved too time-consuming to maintain (DeGroff, 2008). John Blyberg previously from the Ann Arbor District Library and currently the Darian Library in Darian Connecticut has developed SOPAC and two related open source projects (Sheehan, 2009). As reported by Bonfield (2009), although software development continues, the community for SOPAC is not very active (http://thesocialopac.net/forum). It's possible that community interaction occurs in other venues not apparent from the website.

Conclusion

This article evaluated the impact of open source software and the importance of open source communities for libraries. While library staff may not be able to participate as programmers, other types of participation can assist the community such as quality assurance, writing documentation, translating, and assisting other users. The advantages to contributing to an open source community are: improved support when needed, improved knowledge of the product, and an improved community. Failing to participate creates a risk to the library of losing sight of the project direction which may ultimately make the product unsuitable. Participating in OSS communities can educate library staff in the practical skills needed to develop in-house software. For libraries with technical resources, developing and sharing new software applications is a way to demonstrate leadership in the library community.

References

Amaral, M. (2008) Institutional Repositories, Open Source Options, and Libraries. *Open and Libraries Class Journal, 1*(1). Retrieved November 1, 2009 from http://infosherpas.com/ojs/index.php/openandlibraries/article/view/11/20

Bisson, C. (2007a May/June). Building open source. *Library Technology Reports, 43*(3), 36-39.

Bisson, C. (2007b May/June). No-Brainer. *Library Technology Reports*. Retrieved November 4, 2009 from http://maisonbisson.com/oss4lib/no-brainer/

Bonfield, B. (2009, July 22). W-E-B-S-I-T-E, find out what it means to me. *In the Library with the Lead Pipe*. Retrieved November 3, 2009 from Library, Information Science & Technology Abstracts with Full Text. Also available at http://inthelibrarywiththeleadpipe.org/2009/w-e-b-s-i-t-e-find-out-what-it-means-to-me/

Breeding, M. (2008, March). Making a business case for open source ILS. *Computers in Libraries,* Retrieved November 4, 2009 from http://www.librarytechnology.org/ltg-displaytext.pl?RC=13134.

Breeding, M. (2009, June/July). Open discovery interfaces. *American Libraries, 40*(6/7) 40.

Cana, M. (2003, July 5). Open source and Ranganathan's Five Laws of Library Science. *Public infosophy: socio-technological rendering of information*. Retrieved October 25, 2009, from http://www.kmentor.com/socio-tech-info/2003/07/open-source-and-ranganathans-f.html

Chudnov, D. (2009, February). What librarians still don't know about free software. *Computers in Libraries*, 29(2), 22-24.

Dahlander, L., & Magnusson, M. (2008). How do firms make use of open source communities. *Long Range Planning, 41*, 629-649.

DeGroff, A.B. (2008, Winter). Free and Open Source Software at Howard County Library. *Nebraska Library Association Quarterly*, 39(4), 11-15.

Griggs, K. (2009, February). Library information made to order: an open source project built for and with librarians. *Computers in Libraries*, 29(2), 13-14, 46-47.

Hadro, Josh. (2009, October 30). Marketing materials from ILS vendor touches off a flurry of reaction. *Library Journal.* Retrieved November 2, 2009 from http://www.libraryjournal.com/article/CA6704622.html

Houser, J. (2009a). Open source operating systems in libraries: research in context. *Library Technology Reports, 45*(3), 8-10.

Houser, J. (2009b). Open source public workstations and today's library world. *Library Technology Reports, 45*(3), 5-7.

Hoppenbrouwers, J. (2007). Community customers. In St.Amant, K. & Still, B. (Eds.), *Open source software: technological, economic, and social perspectives* (pp. 510-521). Hershey, Pennsylvania: Information Science Reference.

Krishnamurthy, S. (2002, June 3). Cave or community? An empirical examination of 100 mature open source projects. *First Monday, 7*(6), Retrieved November 4, 2009 from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960/881

Lilly, J. & Beltzner, M. (Speakers). (2009, September 24). Firefox, Mozilla, & open source: software design on scale [1:09:54]. *Stanford University YouTube Channel.* Video retrieved November 2, 2009 from http://www.youtube.com/watch?v=S2771Tlr_38

Lietzau, Z. (2009 October). U.S. public libraries and Web 2.0: what's really happening? *Computers in Libraries*, 29(9), 6-10.

Molyneux, R.E. (2009, January). Evergreen in context [Special section]. *Bulletin of the American Society for Information Science and Technology*, 35(2), 26-30.

Morgan, E.L. (2009, March 28). Open source software: controlling your computing [Presentation for Computers in Libraries Conference]. Retrieved November 2, 2009 from http://infomotions.com/musings/oss4cil/

October 2009 Web Server Survey. (2009, October 17). Retrieved November 5, 2009 from http://news.netcraft.com/archives/2009/10/17/october_2009_web_server_survey.html

Owen, G.W.B. & Stranack, K. (2007). The Public Knowledge Project and the Simon Fraser University Library: a partnership in open source and open access. Retrieved November 3, 2009 from http://eprints.rclis.org/15613/

Schneider, K. (2008). Free for all: open source software. *School Library Journal, 54*(8), 44-46.

Sheehan, K. (2009, February). Creating open source conversation. *Computers in Libraries*, 29(2), 8-11.

Stam, W., & Wendel de Joode, R. von (2007). Analyzing firm participation in open source communities. In St.Amant, K. & Still, B. (Eds.), *Open source software: technological, economic, and social perspectives* (pp. 495-509). Hershey, Pennsylvania: Information Science Reference.

Stranack, K. (2007, July). The reSearcher software suite: a case study of library collaboration and open source software development. Retrieved November 2, 2009 from http://eprints.rclis.org/15612/

Tansley, R., Smith, M., & Walker, J. (2005). The DSpace open source digital asset management system: challenges and opportunities. In A. Rauber et al. (Eds), *Lecture Notes on Computer*

Science, Volume 3652, 242-253. Retrieved November 2, 2009 from http://hdl.handle.net/1721.1/29462

van den Berg, K. (2005). Finding open options. Master's thesis, Tilburg University. Retrieved November 3, 2009 from http://www.karinvandenberg.nl/Thesis.pdf