

Reducing Resistance to the Adoption of Open Source Systems

Abstract: This paper examines the background and psychology of the open source movement and suggests ways to decrease resistance to the adoption of open source systems in libraries, which have typically favoured proprietary systems. Resistance is frequently based on deep-rooted institutional and psychological factors, which can be overcome with sensitive and purposive training and implementation strategies. While the paper is directed at libraries specifically, many of the underlying causes of resistance to open source apply to institutions generally, and the suggestions outlined could be applicable in many industries beyond libraries.

About the Author(s): Katherine Johnson is in the third of Dalhousie University's combined Master of Library and Information Studies and Bachelor of Laws program. She also holds a BA in History from Saint Mary's University. This paper was originally written for an upper year Information Management course, INFO 5590, Information Management Systems.

Introduction

While open source has become increasingly popular in the wider world, libraries have been resistant to implementing open source resources. There are many reasons for this, most of which can be overcome with a careful implementation plan that addresses the underlying cultural factors and fears that undermine open source systems projects.

In order to understand the resistance to open source, one must understand what the movement means - its principles, benefits, and drawbacks. There are obvious drawbacks, despite the enthusiasm many outside of the library context seem to have for open source.

Despite the obstacles, open source has much to offer libraries, particularly in terms of maintaining relevance with crucial user groups. Users may feel that without the increased usability and customization that open source offers, libraries are falling behind and failing to meet their needs. Open source can also improve libraries' freedom by reducing their reliance on vendors, and allowing them to make their systems what they actually need them to be, rather than what the vendor will provide them.

Background & Basics

The term "open source" may be used to describe a number of different things, from the Open Source Initiative of 1998, to an array of allied movements, such as free software, open access, open content, free culture, and, of course, open source software. The general principles behind all of these movements are similar: freedom of access, furthering the common good through the shared creation of resources, free distribution, and collaborative development. The latter two points contribute to one of the most attractive features of open source resources: customizability. As Eric Raymond explained, "Good programmers know what to write. Great ones know what to rewrite (and reuse)" (Bisson, 2007c, p. 16). Adopters and users can alter and enhance resources to better suit their own personal or institutional needs, and in so doing evolve into developers and contribute to the broader community (Careaga & Jaffe, 2007, p. 2).

Users may initially be attracted to open source resources because they are "no cost" options but it is important to note that while open source resources are often free in the economic sense they do not necessarily have to be, and, conversely, all freely downloadable software is not necessarily open source (Bisson, 2007b, p. 13). The real benefits of open source resources are improved quality control and better security through cooperation and peer review, and the flexibility to tailor products to meet specific needs, not the lack of up-front costs (Careaga & Jaffe, 2007, p. 3).

Psychology of the Movement

Users may initially turn to open source for a number of reasons. Bisson (2007c) described three basic kinds of users and their motivations: the need-driven consumer, the user-developer, and the core developer.

The "need-driven consumer" makes up the largest part of the open source community. These users participate because the software either solves specific problems or fulfills their needs in a way that proprietary software may not. They may or may not take on an active programming role, depending on their level of interest and ability, but they often report bugs to the community which are subsequently fixed by someone else.

The "user-developer" may contribute code and participate in discussion and advocacy. Often they want to learn, have fun, and make a contribution as well as fulfill their own needs. The smallest number of contributors falls into the category of "core-developer." These individuals take on the bulk of the work in terms of fixing bugs and advancing the project (Bisson, 2007c, pp. 17-18).

Bisson's categories of users correspond generally with Mikkonen, Vaden, and Vainio's motivational classification scheme. Bisson's "need-driven consumer" would align most closely with the self-enriching motivation, which seeks to satisfy individual needs and desires. The "user-developers" likely have more group-enriching motives, striving to meet the needs of the larger community. Finally, the "core-developers" are apt to be motivated by knowledge-enriching concepts, particularly a desire to learn collectively and individually (Mikkonen, Vaden & Vainio, 2007).

Hesitant potential adopters in libraries should feel more confident knowing that their role in the larger community can be virtually as small as they like. Depending on their level of interest, skill, time, and institutional support, librarians could potentially take on any one of the roles noted above. They could also shift between them over time, perhaps increasing their level of participation as their skills and confidence increase, or reducing participation when it is impractical.

Reasons for Adopting Open Source

Different groups are attracted to open source for different reasons. Some emphasize the economic benefits, while others are drawn in by the rights and freedoms that open source can offer (Bisson, 2007b, p. 14). The actual economic benefits of open source in the context of a library may not be as great as they might be in a personal, rather than institutional setting. While the real economic costs of an open source implementation may not be as low as some

might imagine, open source nevertheless has many features which should be of interest to libraries. These include:

- local control
- customizability
- interoperability
- vendor independence
- reliance on open standards
- collaborative development
- flexible support options (Careaga & Jaffe, 2007, p. 5)

Libraries can also benefit from the same positive features which attract most users: access to the software's source code and the power to change and enhance it to meet specific needs (Bisson, 2007b, p. 14). While open source software is rarely the ideal solution immediately upon adoption, it can improve and evolve exponentially through the efforts of users, both within the organization, and in the wider development community. Ultimately, open source projects have far greater potential for growth than the proprietary models commonly used in libraries (Bisson, 2007c, p. 18).

For example, in the case of a proprietary ILS system, changes can only be made by the vendor. Requests for alterations must be made to the vendor and have to be paid for if approved. The potential to create add-on components without vendor participation is virtually nonexistent because the source code is not available. As a result, library systems may lag years behind the broader internet in terms of usability while libraries wait for vendors to create critical new features. Useful features may also be bundled in with less desirable ones which cannot be removed. Finally, migrating to a different system in the hopes of getting the features that libraries need and want is expensive and time-consuming (Careaga & Jaffe, 2007, p. 7).

A great deal of trouble could be avoided, or at least reduced, if libraries adopted more open-source technologies. The right to experiment with open source software enhances the possibility of innovations that patrons have come to expect, and indeed demand, from electronic services (Bisson, 2007e, p. 36). As Matthew Thomas warned, in the long run, "the usability of any non-Free software approaches zero" (Bisson, 2007a, p. 11). If libraries want to stay on the curve and remain relevant to their patrons, they may have to consider

implementing open source technologies just to come close to keeping pace with developments in the wider world.

Problems with Adopting Open Source

Libraries lag behind the broader web market significantly in terms of open source adoption. Only forty-one per cent of library websites use open source, while forty per cent are run on products offered by Microsoft. This compares to seventy-three percent open source implementations versus twenty per cent use of Microsoft products in the broader market (Careaga & Jaffe, 2007, p. 8). Some of the possible reasons for this will be discussed next.

While many individual users are attracted to open source because it is economically "free," in an institutional setting such as a library, the actual start-up costs can be significant. These costs include development and working out licensing and governance practices. Open source products are rarely an ideal fit straight out of the box, so investment in development and customization can be significant, both in terms of money and time. In some cases, these costs can be so high that libraries are deterred from implementing open source products, and they choose to remain with proprietary, vendor-supplied products instead (Careaga & Jaffe, 2007, p. 9).

Careaga and Jaffe (2007) suggest that another reason libraries have been reluctant to take advantage of open source technologies is an institutional bias against "free" products, with a preference for name brands. They suggest that this bias is reinforced by a lack of experience with open source. Libraries advocate for the implementation of basic tools in order to acclimatize librarians to open source in a relatively non-threatening way. Presumably the purpose of this would be to eventually turn libraries around to a preference for open source over proprietary, or "name brand" products (Careaga and Jaffe, 2007, p. 8). If this bias is not corrected, Careaga and Jaffe (2007) contend that libraries' ability to innovate will be compromised, users will see new features being made available everywhere except the library, and libraries will become "even more marginal" as a result(p. 9).

While it is very likely true that lack of exposure to open source in the library context may prejudice some librarians against adopting non-vendor supplied products, whether that actually amounts to a "bias" is debatable. Also, as open source is highly prevalent in the wider online world, librarians are more likely to have interactions with open source products on an increasingly regular basis, although they may not realize it. Following Careaga and Jaffe (2007)'s logic, as exposure increases, fear will decline, and any "bias" that may exist will inevitably be reduced. Whether this will happen quickly enough for libraries to implement the features that users expect remains to be seen.

Whether you frame the problem as overcoming bias, or simply adjusting to a new way of thinking, change is always difficult in an institutional context. If employees believe that open source will be more challenging and less reliable out of the box as compared to vendor-supplied systems, they will almost certainly be resistant to any move to implement an open source project. Some people may also be threatened by the openness and transparency which define the whole concept of open source. They may also have concerns about accountability for the code, while it is in the developmental stage, and that sharing the code may lead to personal embarrassment over mistakes (Neus & Scherf, 2005, p. 216). All of these are cultural concerns, which run deeper than superficial worries over time and financial costs. If the root cause of these concerns is not dealt with and only the symptoms of the underlying mistrust are treated, the implementation of an open source project in a new organization will most likely be unsuccessful. While many people may be more comfortable with proprietary products because they are accustomed to them and, theoretically, these products come with more reliable support, in reality, proprietary models are not as easy to use as is thought. Vendors may supply products which have not been thoroughly tested, causing unexpected problems. Having a vendor to call does not necessarily guarantee that response times on support enquiries will be timely and helpful, particularly when a new product is found to have more bugs than anticipated (Careaga & Jaffe, 2007, p. 11). Until trust in open source is cultivated, however, many librarians and administrators may be willing to stick with "the devil they know" in the form of the vendors for fear of being left totally alone and unsupported after implementing an open source product. Suggestions for how to make the mental and cultural transition easier will be provided in the next section.

Suggestions to Make the Project Work

Before implementing an open source project, particularly a large one, or one that impacts on a heavily used resource, it is essential to address the cultural factors that could undermine the project's success. This entails moving from the traditional paradigm, known as "Brooks' Law," to "Linus' law," which is characterized by an attitude more conducive to open source thinking.

Brooks' Law operates on the assumption that only a small group of people should be involved in the creation, design, or improvement of information. Under this model, the vast majority of people are relegated to a passive role, simply receiving the information and providing little or no feedback. Linus' Law, on the other hand, spreads the tasks over a much larger group of people, decreasing the transaction cost of each activity and increasing the likelihood that problems will be solved quickly and efficiently through active participation and feedback (Neus & Scherf, 2005, p. 216).

Moving from one model to another requires convincing users that practitioners, rather than just experts, can make significant contributions to software development (Neus & Scherf, 2005, p. 217). If an organization tries to effect change without addressing these underlying beliefs, behaviour may not change and the project may be compromised. People may attempt to look like they are implementing the new model, when in fact they are still behaving as close to the "same as usual" paradigm as they can get away with. To prevent this, top leadership people must change their own behaviour which should cause a trickle-down effect, bringing about change at the lower levels as people gain experience with the positive aspects of the new system (Neus & Scherf, 2005, p. 219).

Problems may arise at the upper levels if individuals, or even departments, feel that their power and position may be compromised by introducing more openness. They may also fear that more openness will lead to more criticism when things inevitably go wrong. To overcome this challenge, management must clearly address the concerns of and benefits for each stakeholder specifically (Neus & Scherf, 2005, pp. 219-220). If people can see how the change will make their own lives easier and that the new system will not necessarily make them more vulnerable, they will be more likely to accept more openness and collaboration.

If the library does not have experience with open source, management and staff will likely be wary of making the change for the reasons noted in the previous section. To counter this, it may be helpful to start small and enhance further once people are more comfortable with the concept and have seen some benefits from the implementation. Examples of smaller projects which can be integrated into the existing are LibX, developing a website with links to open source information and resources, installing open source RSS feeds onto the library's website, or creating wikis for collaborative activities (Careaga & Jaffe, 2007, p. 12).

Wikis can be helpful in a number of different contexts, including as discussion boards or as content management systems (Raman, 2006, p. 59). The installation process may require some time and effort in terms of configuration and database management, but once the wiki is installed, it is very easy to use, making it an attractive option to convince resistant users. (Raman, 2006, p. 60) If you can demonstrate value, people will be more amenable to moving forward.

Keep the project simple, and keep it small at first. Avoid overloading people resistant to open source implementation with complex methodology and jargon. Make the process simple and easy to understand so it is as non-threatening as possible. Use passionate people to drive change and drum up excitement and enthusiasm. Move toward a culture of openness beyond the systems. If people feel that they can be open and make suggestions to improve the library in general, they will be more likely to accept the principles behind an open source system. Springing from that, make sure all stakeholders are involved and have an opportunity to share

their views and concerns so they can be addressed before, rather than after, problems arise (Neus & Scherf, 2005, p. 224). If these tips are followed, an open source implementation will go much smoother and the responses to it will be more positive than if a project is undertaken without accounting for the culture of the organization.

Case Study

An example of a successful move to open source is the government of the state of Massachusetts. In 2005, the government announced new IT standards requiring its employees to adopt open file formats. The test for openness was as follows: the file format "must be published and subject to peer review"; it "must be subject to joint stewardship"; and it "must have no or absolutely minimal legal restrictions attached to it." (Bisson, 2007d, p. 23). The government eventually settled on the Open Document Format and Adobe PDF to meet its needs. Microsoft's offerings, on the other hand, failed to meet the test for openness because its license did not allow users to build new applications that could read and write the file format (Bisson, 2007d, p. 23).

If a state government overseeing more than 80,000 employees could find a way to make open source work on a large scale, surely libraries can find a way to implement open source solutions for at least part of their systems needs.

Conclusion

Open source resources have a lot to offer libraries in terms of flexibility and customization. Implementing open source applications could help libraries maintain relevance with patrons, who have increasingly high expectations of technology and who are familiar with many open source-based tools on the broader internet. Open source resources could also help libraries avoid being hamstrung by vendors, which often do not allow for any sort of user customization.

Implementing an open source project will likely be met with resistance in environments where open source has limited or no penetration. In those cases, cultural factors must be addressed in order for the project to be of lasting benefit. Libraries should start small and build trust in open source before attempting a major overhaul of existing systems. Administrators hoping to bring in new open source tools should also remember that "free" does not necessarily mean cost-free, as a great deal of investment will likely have to be put in at the early stages to make the software work in the specific environment and to bring reluctant staff and users on board.

In the end, the balance of the cost-benefit analysis will vary with the context and the resource proposed. As open source continues to evolve, however, it will likely outstrip proprietary products in terms of usability and responsiveness to users' needs. As the Massachusetts

government case indicates, this may tip the balance in favour of open source in the long run.

References

(a) Bisson, C. (2007) Why freedom matters, *Library Technology Reports*, 43(3), p.11-12.

(b) Bisson, C. (2007). "Free", "free", and "open source", *Library Technology Reports*, 43(3), p.13-15.

(c) Bisson, C. (2007). What makes open-source work?, *Library Technology Reports*, 43(3), p.16-20.

(d) Bisson, C. (2007). Open systems, formats, and standards, *Library Technology Reports*, 43(3), p. 21-27.

(e) Bisson, C. (2007). Building open-source, *Library Technology Reports*, 43(3), p.36-39.

Careaga, G. & Jaffe, L.D. (2007). Standing up for open source. *Library philosophy and practice*, 2007, June, p.1-17.

Mikkonen, T., Vaden, T. & Vainio, N. (2007). The protestant ethic strikes back: open source developers and the ethic of capitalism, *First Monday*, 12(2). Retrieved October 19, 2007, from <http://www.firstmonday.org/issues/issue122/mikkonen/index.html>.

Neus, A. & Scherf, P. (2005). Opening minds: cultural change with the introduction of open-source collaboration methods, *IBM Systems Journal*, 44(2). Retrieved November 23, 2007, from <http://www.research.ibm.com/journal/sj/442/neus.pdf>.

Raman, M. (2006). Wiki technology as a "free" collaborative tool within an organizational setting, *Information Systems Management*, Fall 2006, p.59-66.