

Taking matters into our own hands: influencing factors and concerning factors for libraries that developed Open Source library software

Kathryn Greenhill
Special Services Librarian
Cottesloe-Mosman Park-Peppermint Grove Library
Kathryn.Greenhill@gmail.com
<http://KathrynGreenhill.com>

Abstract:

Libraries that develop Open Source library software are influenced by a number of factors. The insights from this group are useful for library staff who are considering adopting Open Source library software. The author surveyed and interviewed staff who developed Scriblio, SOPAC2, VUFind, Blacklight, Koha and Evergreen. She also interviewed librarians who specified the Open Library Environment. The interviews and questionnaires revealed a common attitude to the cost of Open Source Software; similar concerns about speed and process for enhancements to proprietary library software; and similar satisfaction with the control and community that Open Source software has brought to their library operations.

Introduction

Moving to Open Source software to run core functions in your library can seem like a daunting decision. There are so many influential factors that make it appear to be a good idea, and so many concerning factors that make it seem risky.

Developers and managers in libraries that developed their own core library software and then released it as Open Source were also influenced and concerned by a large number of factors, such as cost, staffing, flexibility, integration with other systems, data conversions and on-going support.

This paper looks at three broad questions:

- Why did libraries develop Open Source software?
- What risks were there for libraries when they developed Open Source Software?
- What can libraries thinking of adopting Open Source library software learn from this?

Asking developers and managers “why did you do it, and did you worry about anything?” would produce very broad answers that would be hard to compare between participants. To answer the first two questions, I surveyed developers and managers to discover whether the advantages and risks cited in formal literature matched their experience. Advantages and benefits are referred to as “influencing factors” in this paper. Disadvantages, risks and costs are referred to as “concerning factors”. Interviews with participants revealed useful points for consideration for libraries thinking of adopting Open Source library software.

Before undertaking this study, I had three presumptions about the development of Open Source library software. These proved incorrect. My presumptions were:

- That the libraries would have a rather regular organisational culture, but that they employed some outstandingly clever programmers who were allowed to develop their own pet projects.
- That one of the most influential factors was a reaction against the proprietary software licensing model and vendors who did not provide adequate support or maintenance.
- That developing Open Source software was a cheaper option and that this was a major influence.

The six Open Source library applications in this study are:

- Scriblio – an OPAC replacement and Content Management System based on Wordpress
- SOPAC2 – a “social OPAC” that improves the OPAC search and allows social interactions like reviewing, rating and tagging.
- VuFind – an OPAC replacement written in PHP
- Blacklight – an OPAC replacement using Ruby on Rails

- Koha – an integrated library management system replacement
- Evergreen – an integrated library management system replacement

Also included in the study is the *Open Library Environment* (OLE) Project, which created the specifications for a software solution for large research libraries. OLE is planned to be released under an Open Source license (OLE Project 2008).

Background

In the last ten years, approximately fifteen libraries worldwide have embarked on major projects to write their own software for core library functions and to release this code under an Open Source License (Bisson 2007; Breeding 2008). These projects have generated much professional interest. According to Breeding (2009), “hundreds if not thousands of libraries have adopted an Open Source ILM [Integrated Library Management System]”.

The table below shows many of the major Open Source Library software applications:

Open source library applications	
ILSs	OPACs
Evergreen	Blacklight
Koha	Fac-Back-OPAC
Repositories	MARC Module for Drupal
Digital Asset Factory	Scriblio
DSpace	SOPAC
Fedora	VuFind
Metasearch Resolvers	
CUFTS	
LibraryFind	

Figure 1. Open Source Library Applications (Colford 2008)

Proprietary software licensing model

There is nothing new about libraries developing their own software. The first library automation projects involved straight-out stock control using punch card systems (Melin 1964). As libraries expanded their automation from stock control to resource discovery, there were again few off-the-shelf systems that met this need.

The first proprietary software products, for example Auto-Graphics and NOTIS, were not marketed until the late 1960s/early 1970s (Breeding 2009). In Australia, Kim Jelbart’s company, ADAPS Data Centre, provided software solutions to several libraries from the late 1960’s (Groenewegen 2004). By the mid 1990s, however, most of the software used in libraries were generic solutions created outside the individual library. Libraries paid an ongoing “proprietary” license to use these.

Typical features of the proprietary licensing model are:

- Licenses are restricted by number of users or ports.
- The library has no access to the product source code.
- The library is not allowed to amend the product in any way.
- Modifications by the software vendor are based on priorities not determined by individual customer. Modifications are often submitted via a user group, but resolutions by the user group often compete with the vendor's other plans for the product.
- Some systems were initially licensed as "turnkey" systems, where proprietary software was bundled together with proprietary hardware.

Open Source Licensing

As more libraries adopt Open Source software, there is a wider understanding in the library community that Open Source does not mean "no-cost software with which you can do whatever you want". This may be the case with some Open Source software, but there are over 60 different Open Source licenses approved by the Open Source Initiative, all with different requirements, rights and responsibilities (Licenses By Name | Open Source Initiative n.d.).

The first software developed by a library and issued under Open Source was probably in 1999: MyLibrary from Erik Lease Morgan at North Carolina State University, issued under an GNU General Public License (Lease Morgan 1999). Koha, Evergreen, VuFind, SOPAC2 and Scriblio are all released under some version of GNU General Public License. Blacklight is released under an Apache 2.0 license. OLE aims ultimately to release under "an open source license". (OLE Project 2008)

To be defined as Open Source, the license under which the software is released must fulfil all ten criteria laid down by the Open Source Initiative (Open Source Definition| Open Source Initiative n.d.). Many library staff are familiar with the first four of these criteria, but may not have considered effect of the last six criteria on the systems architecture or social utility of Open Source software.

Partially paraphrased, these criteria are:

1. **Free redistribution**

The license shall not restrict re-distribution by sale or give-away.

2. **Source code**

Everyone must be able to access the source code, not just the compiled form.

3. **Derived works**

The license must allow modifications, derived work and allow distribution with the same sort of license.

4. Integrity of the Author's Source Code

Modified versions of the source code may be restricted from distribution, only if the license allows distribution of "patch files" to modify the source code.

5. No discrimination against persons or groups

The license must not discriminate against any person or groups of persons.

6. Not discrimination against fields of endeavour

The license cannot restrict use for particular purposes.

7. Distribution of license

The rights conferred by the initial license continue without the need for an additional license by redistributors.

8. License must not be specific to a product

The software cannot only be made available in a bundle with other products.

9. License must not restrict other software

The license must not specify that it can only be bundled with other software that meets specified criteria (eg. Only distributed with other Open Source products).

10. License must be technologically neutral

The software must not be restricted to any individual hardware or interface style.

Influencing and concerning factors in the development of Open Source Software

Current research into how organisations outside libraries develop and accept Open Source software suggests a number of influential factors and concerning factors. Coders in Open Source projects are more often motivated by user need, intellectual stimulation derived from writing code, and improving their programming skills - rather than job advancement or money (Lakhani & Wolf 2003). Firms that contribute to Open Source projects are more likely to be motivated by a chance to harness the talents of a group of developers not yet known to the general market (Prüfer 2004).

More has been written about the motivations for adoption of Open Source Software than the development of it. Firms tend to adopt for technological and economic reasons, rather than community-motivated reasons (Bonaccorsi & Rossi 2006). Morgan and Finnegan (2007) identified many factors from the literature that are

viewed as benefits, including reliability, security, performance, flexibility of use, escaping vendor lock-in and encouraging innovation.

Levesque (2004) suggests that concerns about Open Source software adoption include bad documentation and a difficult user interface. Other possible concerns include security risks, installation problems, lack of expertise, lack of ownership and a higher training investment (Morgan & Finnegan 2007). In Australia, managers who reject Open Source software are concerned about a lack of relevance of the software, a lack of formal support, a lack of resources to assess and implement Open Source and their firm's pre-commitment to Microsoft (Goode 2005).

Library specific literature identifies similar concerns and benefits to the general literature about adoption of Open Source software. There are also some library-specific factors. Of particular note is what John Blyberg calls a "flawed partnership system" (Blyberg 2007, p.32) between libraries and vendors of proprietary library software. Libraries are forced into a state of learned helplessness because they are unable to control changes to the system (Schneider 2009), receiving support that "rarely matches the value – monetary or quality – attributed to it by the vendor or non-technical library staff" (Colford 2008, p.13). Recent changes of ownership of companies have left "libraries skeptical about the business environment of the incumbent vendors" (Breeding 2009a, p.24).

Other library-specific factors that influence the development of Open Source library software are a "growing cohort of programming savvy librarians in our midst" (Bisson 2007, p.36), professional conferences that discuss Open Source library applications (Breeding 2008), a rise in companies that support Open Source library software (Breeding 2008) and an existing professional philosophy of our collections being open, available and visible for all to see (Schneider 2009).

Methodology

Two people associated with the development of each software application were interviewed: a technical developer and a manager who supported the decision to develop the library's own software and release it under Open Source. These participants are referred to below as "Developers" and "Managers".

The paper also looks at the influencing and concerning factors for another six librarians who worked on the OLE Project. These participants are referred to as "Specifiers".

Participants were asked to complete an online questionnaire. All participants were then interviewed face-to-face, except for one who was interviewed via Skype.

Questionnaire

The questionnaire asked participants to indicate on a Likert Scale whether specified factors were influential or concerning in relation to their development of the Open Source library software. They could also indicate that they did not agree that the factor listed was true.

The factors were identified from formal literature about libraries. There was an imbalance of influencing factors (46) to concerning factors (13). This is partly because fewer concerning factors are mentioned in the literature, but also because where a factor may have been either concerning or influencing it was framed positively and listed in the influencing factors (eg. “***There are more library staff with coding/development skills than in 1999***” could have been listed as the concerning factor “***There are not as many library staff with coding/development skills as in 1999***”).

Questionnaire analysis

Weighted means were calculated for each factor.

Each response on the Likert scale (of 1 to 4) was given a numerical weighting according to how influential it was (for one set of questions), or how concerning it was (for another set of questions). An answer of “do not believe this statement is true” was assigned a weighting of zero.

For the questions about influential factors, for example, an answer of “very influential” was given a value of 4, and an answer of “not very influential” was given a 1. Means were then calculated for every question. A factor with a mean of 3.4 would thus be considered by participants as slightly more influential than one with 3.2 – but both would be somewhere between “rather influential” and “very influential”.

Appendix One contains a print version of the questionnaire that was administered via the survey collection software, Survey Monkey.

Interviews

The interview aimed to determine additional influential factors or concerns that were not covered in the questionnaire, and to allow participants to expand on the significance of identified factors.

Questionnaire results: all participants

Most influential factors

The seven most influential factors all had a mean of over 3.2, so were all considered somewhere between “rather influential” and “very influential”.

Factor	Weighted Mean 4= Very influential 1= Not influential at all
1. More control of features	3.78
2. Existing software product used by the library does not meet the library needs	3.78
3. Easier to modify	3.67
4. You can make the system interoperable with other systems	3.67
5. You can work with recent software development tools and methods	3.39
7. If others see the source code, then they can develop it further	3.39
8. You do not have to wait until lots of libraries want the same modification before it happens	3.39

Table 1. Most influential factors ranked from most to least influential. All participants

Least influential factors

The seven least influential factors all had a mean of 2.3 or under, with just one with a value of lower than 1 (not very influential at all).

Factor	Weighted Mean 4= Very influential 1= Not influential at all
1. Open Source support companies only provide support, so will not cut back on support for other revenue streams	1.39
2. Many companies now exist to support Open Source Library Software	2
3. Compelled by the license of the development tools to release the product under Open Source	2
4. Self-developed software costs less.	2.06
5. There are more library staff with coding/development skills than in 1999	2.11
6. The Open Source community is large, mature and sustainable.	2.17
7. Vendors have not previously delivered on maintenance agreements	2.17

Table 2. Least influential factors ranked from least to most influential. All participants.

Concerning factors.

Participants felt less strongly about the concerning factors.

In contrast to the influencing factors –where 44% of the factors had a mean greater than “rather influential” – no concerning factors had a mean greater than “rather concerning”. Eight of the thirteen concerning factors (62%) had a weighted mean lower than 2 (not very concerning).

Factor	Weighted Mean 4= Very concerning 1= Not concerning at all
<i>Open Source Library Software may make the organisation too dependent on the staff member who knows most about it.</i>	2.89
<i>It may be hard to hire someone with the right skills to support Open Source Library Software</i>	2.61
<i>Our library may not be able to afford staffing to support Open Source</i>	2.33
<i>Total Cost of Ownership may be more or the same as proprietary software</i>	2.28
<i>Without a vendor with contractual obligations, then there is nobody legally obliged to make sure the Open Source Library Software works</i>	2.17
<i>You need to create higher quality code if you release it under Open Source</i>	1.83
<i>It is hard to convince other people that Open Source software is not chaotic, anarchic and depends on people doing only what they feel like</i>	1.83
<i>Other software used in our organisation is proprietary and incompatible with Open Source</i>	1.78
<i>There is a much lower level of Corporate Risk when you rely on a vendor that is audited and run on a business model</i>	1.61
<i>The IT department is not used to supporting Open Source</i>	1.39
<i>It is easy to find malicious exploits if you can see all the code</i>	1.28
<i>Your organisation will not earn profits from your development</i>	1.11
<i>Open Source cannot match the sophistication and functionality offered by proprietary software</i>	0.94

Table 3. All concerning factors ranked from most to least influential. All participants.

Factors that the participants did not agree were true

There were 25 factors where at least one respondent indicated that they thought the statement was untrue. Interestingly, five out of six developers did not agree that the first factor below was true.

Factor	Number who did not agree factor was true Out of 18 participants
<i>Open Source cannot match the sophistication and functionality offered by proprietary software.</i>	9 (50%)
<i>It is easy to find malicious exploits if you can see all the code</i>	5 (28%)
<i>The IT department is not used to supporting Open Source</i>	5 (28%)
<i>Self-developed software costs less</i>	4 (22%)
<i>The Open Source community is large, mature and sustainable</i>	4 (22%)
<i>There is a much lower level of Corporate Risk when you rely on a vendor that is audited and run on a business model</i>	4 (22%)
<i>Library staff are more likely to be familiar with tools like MySQL, Apache and PHP</i>	3 (17%)
<i>Open Source support companies only provide support, so will not cut back on support for other revenue streams</i>	3 (17%)
<i>Your organisation will not earn profits from your development</i>	3 (17%)

Table 4. Factors that 3 or more participants did not believe were true. All participants.

Factors in the tables above are discussed in greater detail in the following sections.

Questionnaire results: by groups

Interoperability with other systems was the most important factor for the Specifiers. This is not surprising as one of the key aims in the design of the OLE involves some components being provided by other data systems in the organisation. This factor ranked fourth for the Managers and ninth for the Developers.

The Developers were most influenced by the fact that the **existing product used by the library does not meet library needs**. This was second most influential for the Managers, but only fifth most influencing for the Specifiers. It may be a reflection of the Developers' greater knowledge of exactly what software is able to do.

Managers were most influenced by having **more control of features**. This was second most influential for the other two groups.

Other factors also showed differences between groups. The eighth most influential factor for Specifiers (who were all potential users of OLE) was that **potential users can test the product during development**. This was only ranked 22nd and 21st by Developers and Managers respectively.

Of note is that three factors were ranked much higher by the Developers than the other two groups. The factors with the Developers' ranking are listed below, with the second and third numbers indicating the ranking by Managers and Specifiers:

- ***Creating good Open Source software enhances the organisation's reputation*** (D-5th, M-27th, S-30th)
- ***Open Source lets you give back to your community*** (D-7th, M-29th, S-21st)
- ***Proprietary library software vendors have not previously delivered on software requirements*** (D-4th, M-18th, S-12th)

The first two factors – reputation and giving back - relate to working in a community. A possible explanation is that one of the practical daily steps for developers in writing their code is to connect and communicate with the community that has grown around the product.

The Specifiers, who are still using proprietary software, ranked two factors related to vendors as much more influential than the other two groups.

- ***The library will not be locked into a support of maintenance contract*** (S-4th, M-14th, D-19th)
- ***Vendor business decisions (like business mergers and acquisitions) impact negatively*** (S-23rd, M-41st, D-44th)

Concerning Factors

Although there were small differences between the groups for most factors, two factors were ranked quite differently.

The fourth most concerning factor for the Developers, ***It is hard to convince other people that Open Source software is not chaotic, anarchic and depends on people doing only what they feel like***, was ranked sixth most concerning for Managers and eleventh most concerning for the Specifiers.

The eighth most concerning factor for the Specifiers, ***It is easy to find malicious exploits if you can see all the code***, was ranked eleventh by the Developers and twelfth by the Managers. During interviews, those who did not rank this as very concerning mentioned that this can be positive, as problems are fixed quickly when others can see the code.

Interviews

Developers and Managers were specifically asked about the influencing and concerning factors at the time of development, but it is likely that some of their answers to the questionnaire were informed by their later experience. They all described in interviews how they dealt with praise, criticism and questions about their projects. Some of their insights seem to have been informed by these later discussions. They suggested some factors, which were not included in the questionnaire, that may concern or influence an organisation currently adopting Open Source software.

Overriding influencing factors for developing Open Source library software

The interviews revealed that applications studied could be divided into two major groups. For one group, the overriding influencing factor was *necessity* – the managers felt that there was simply no other option than to develop their own software to fulfil a current need. The other group had a more complex set of influencing facts, but the overriding influencing factor for these was *improvement* - a desire to design a new product with a better architecture or set of functions than anything currently available.

Koha and Evergreen, the integrated library management systems, both fall into the “necessity” category. Scriblio, SOPAC2, VuFind, Blacklight and OLE fall into the “improvement” category.

Necessity

Koha was developed in 15 weeks for the three-branch Horowhenua Library Trust in New Zealand, when the networking platform supporting their current library management system was due to stop working on 1 January 2000. No other cost-effective system that could cope with their unique technical requirements could be found.

Work on Evergreen for the Georgia PINES, with its 270 libraries, began the year before the contract for their incumbent ILMS was due to be renewed. Interviews with major and minor vendors could not find any solution that met the requirement for a single library card across the entire state. In the words of one PINES interviewee:

People oftentimes asked us “Are you radical or just plain crazy?”. Well, actually the truth of the matter is that we were just plain desperate.

Improvement

The nucleus of the reason to develop software locally was due to lack of quality software in the market place along with a major lack in customization with current products that the library was heavily invested in. (Developer)

This quote is typical of the libraries that wanted to develop their software to improve on what was offered. Interviewees stressed that they considered these to be “mission critical” functions. They were already budgeting for and assessing products on the market before they decided to write their own software. They considered that the need for the product aligned with the organisational goals and priorities. For all of them, the library’s need for a better product came before the desire to write one.

Organisational Culture

The developing libraries had a common culture of self-developed solutions and innovation. It was remarkable that almost every interviewee, including the Specifiers, described home institutions that had a track record of other innovative projects that were models for other organisations. Many of the developing libraries already were using Open Source for other projects. During the interviews, the cultures of the organisations were described variously as:

- *“a culture where pockets of innovation can sprout out”*,
- *We have a history of “we can do it better, so we’ll have to do it ourselves”*,
- *“we have a history of the institution doing its own stuff”*
- *“experiment and try new things”*
- *A “let’s try it” attitude*
- *A “let me take that and fiddle” attitude*
- *“a culture of permission”*

The size of the institution was mentioned in several interviews as an important influencing factor. A small library serving 15 thousand people, a large library system serving 270 libraries and a library that described itself as a “medium sized institution” all gave reasons why their size was perfectly suited to developing their project. Despite the claims of the interviewees – it seems that the structure and flexibility of the organisation was more important than the size.

Staff

The interviews revealed – as I had expected – that the Developers were extraordinarily hardworking, creative people who were personally dedicated to the project. Access to people with these personal skills and dedication was seen by the Managers as one of the influencing factors for taking on the project, but not the principal one.

Developers mentioned personal motivations that influenced them to contribute to the project. One said that there was a secondary “feelgood” factor. Another mentioned being motivated by ethics to write software that can be used by citizens of poorer countries.

Given the reliance on the Developers, it is not surprising that the major concern factor was what would happen if the key person left. There were several suggestions about ways to ameliorate this. “Good, realistic, current documentation” was suggested as critical insurance. Building a strong community around the Open Source software was another suggestion. Insisting on metrics and milestones to ensure that the project was not “all in the person’s head” was also suggested. Also critical was ensuring that there was buy-in from other library staff, so that they supported the project and kept it running

There were two conflicting views about the effect of having clever, dedicated developers on staff. One Manager believed it made the organisation more vulnerable

to good staff “being poached”. The other believed it made it more likely that other good staff would want to work there.

Librarians’ understanding of software development and Open Source

Developers and Managers discussed the challenges of developing software in libraries where staff were not familiar with software development or Open Source.

A Manager pointed out that:

When you buy a system like [name of vendor software used by the library] you get used to just making do...workarounds...telling people “Well, you just can’t do that” and that becomes a culture I think...and I saw this as an opportunity to look at it differently.

One of the Developers, when disagreeing with the statement that there are more librarians with coding ability, commented that librarians:

...let that skill atrophy as a profession and that’s unfortunate. What we’ve done is that we’ve offloaded that particular aspect of librarianship and over time we have sort of empowered our vendors as the gatekeepers to our technology and our data

One Developer described how staff were not forthcoming in the focus groups set up to specify the system.

They were trying to give us incremental changes to what they were used to. They wanted us to reimplement [vendor software they were using] with some improvements.

Things changed when they used what the Developer called the “Magic Box” approach. The developers asked the staff to “pretend it was magic and it could work any way that you wanted it to”. This made things drastically different and produced some good brainstorming.

One of the Developers suggested that librarians would understand the need for constant change and development if they compared Software Development to Collection Development:

The software will never be “done”, just like the library collection budget. You don’t say “You’ve “done” the collection, so no more budget for this”. You do not expect your collection to fossilize, so you shouldn’t expect your interface to fossilize either.

When library staff stopped showing “learned helplessness” and started demanding changes to the software, there were positive and negative outcomes. One Manager described a “tension between the personal and the corporate” when developers were part of the library:

They [library staff] have their specific pet interests that they want to see addressed immediately and the idea is that “Oh, if it’s vendor software then you just have to wait for the vendor to get to it” ...[it] cultivates a certain

amount of forced compliance. Whereas in an Open Source environment the idea is “Oh, you’re not getting to it because I’m not important enough” or “it’s something you don’t care about”. It’s much more personal.

Three of the project Managers or Developers also described the delight of their library staff when they wanted something changed and it was completed almost immediately. One library wanted to re-organise their collection, so they needed another field in the OPAC. This was coded by the next day. In another project, to move the location of a button on the screen took just a “phone call and half a day”.

“Narcissism of Small Differences”

Several Developers expressed frustration with the way that software is required to accommodate the procedures and processes that vary so widely between libraries. One Developer used a term from the writing of Sigmund Freud, “the narcissism of small differences”, to describe this. The “narcissism of small differences” strictly refers to the compulsion of one ethnic group to become more united by differentiating itself, sometimes with aggression, from another very similar ethnic group (Freud, 1962, p.61).

One Developer described his initial analysis of the development task as “it’s just a large database, how hard can it be?” After working with libraries, he concluded:

It’s the insane rules that go with the big database that make it hard...I’m yet to meet two libraries that do the same thing...everyone does their own special way of doing everything so it’s really hard to make a generalised solution.

One Manager suggested that this “narcissism of small differences” is an existing long-term cultural practice in libraries:

The single greatest fault of what libraries have done in the last 50 years is, I believe, that each one of them has believed that by a significant intensive effort they could create a catalogue which was better than anyone else’s... how is it that each one of them, that made a slightly different judgment, was right? That’s just egos at work.

One Specifier pointed out the way that the OLE project has acknowledged this challenge in the “Assumptions that frame the design of OLE” section of their draft final report: (OLE 2009)

Business workflows at peer libraries are more similar than they are different.

- *Shared workflows are essential for collaborative efforts and cost-effective for both design and implementation*

There were also comments about how libraries unnecessarily differentiate their software systems from those of other businesses. The design premise of the OLE project is not to replicate systems that are already used by their parent institutions (eg. user addresses, financial systems), but have the data flow between OLE and the external system. Scriblio was predicated on the assumption that it is most useful to find what is common rather than different between library needs and general enterprise needs, so that it could be built on a large, constantly developed and tested platform with a huge worldwide community, Wordpress.

One of the Developers mentioned that it is not only libraries that do not standardise how they do things, pointing out that there is “wilful non-compliance to Open Standards between vendors [for commercial advantage]”.

Vendor relationship

Generally, libraries were not creating Open Source because they were unhappy with vendor maintenance, support arrangements or performance.

One Manager described their motivation for creating their own software as “not driven by failure of the vendors, but new opportunities in new directions”. One Specifier said that the decision to work on the OLE project was not “anti vendor driven”. Her library’s “frustration is more around closed systems and closed data”. A Developer explained that a key motivation was that there was a “lack of customization and development efforts coming from the vendors”.

The growth of companies supporting Open Source was not seen as necessarily a better alternative to vendor contracts. One Specifier was cautious about relying on Open Source support companies, claiming that traditional vendors are much better managed and resourced than many Open Source library companies. A Manager was concerned that much of the “real work” in Open Source projects is “grassroots” and “is from passion behind it”. He was sceptical that companies supporting Open Source library software could “commodify that passion”.

Another Developer commented on the vendor community’s relationships to Open Source projects:

There have been changes for the better in the vendor community because of the prevalence of Open Source and a lot of the vendors actually do use Open Source under the hood. They use things like Perl and Apache...but they were also putting out a lot of FUD, so behind the scenes they’re trying to seed a lot of discontent.

Costs and Benefits

Most interviewees said that Open Source did not cost less, but that they received different value for their outlay. Many of the organisations had already budgeted for a product to fulfil a similar function, so finding extra money, or saving money was not a motivation. As one Developer said, “It is not about costs it is about what it does”. One of the Specifiers said that it “is not about free, it is more about control”.

Another Developer suggested that one should look at “costs and benefits” instead of just costs, and look beyond financial costs:

having unhappy librarians costs, not working for the user costs... vendor costs are inflexibility costs.

The costs of Open Source were described as different to a Proprietary license. In the words of a Manager –

It costs differently. You have to shift your investments from purchasing a commodity, software, to acquiring talent that can support the development of applications.... I think that over time the direct costs may diminish but I've always thought of Open Source as redirection of existing investments rather than saving money.... redirecting investments to get precisely what you need, rather than ...something that has a fuzzy match to what you want but is not exactly it

Three interviewees did believe that Open Source would cost less in the long term. Interestingly, the only person who claimed that Open Source software did cost less – with a definite emphasis on the word “did” when she said it – was the Manager who had been using Open Source library software for almost ten years so far.

Community and Open Source

One of the attractions of Open Source for many of the interviewees was the ability for other developers outside the organisation to expand the program. Every Developer mentioned this as an essential part of their development strategy.

One Developer said that the project was “written with the express intent that it could be built on”. The two ILMS replacements were both deliberately released under Open Source, because the developing libraries could not afford to single-handedly continue to develop the software.

Five of the interviewees mentioned experiences where their organisation had developed different software, but decided not to release it as Open Source. They were concerned about being obliged to become a “leader” of the project, rather than just a participant in the project, and did not want to take on that responsibility.

The corporate relationship to the Open Source projects involved support for the code to be freely available, rather than trying to make a financial profit. One Manager told the board that if they sold the software then they would be obliged to support it. Another Manager said that they did not want to be seen to be earning profits or to patent it as this would make others lose the ability to contribute.

The board of one library did not want to employ the Developers to provide support to others working on the project who were outside the organisation. The library and the developers felt that the only choice in this situation was for the developers to resign from the organisation and form their own company so that the software could grow in a community.

A Manager from one of the projects was disappointed that a larger community had not grown up around the project, and told the cautionary tale of creating reliance on a developer not only leading the project locally, but also being lead for the project

worldwide. Another developer pointed out that “if you have a product that people want then it will attract a community.... but you can’t just release it and expect community to grow around it”.

Implications for practice

The experiences shared by these Developers, Managers and Specifiers have several implications for library staff thinking of adopting library-specific Open Source software.

1. Do not expect to save money

Open Source will only shift costs to different areas, mainly from licensing costs to hiring staff with different skills or developing different skills for existing staff.

2. Open up organisational culture

Ensure that your culture allows staff to experiment and supports appropriate Open Source solutions in other areas.

3. Ensure adequate staffing

If an Open Source library software support company is not used then the organisation will need to employ a developer.

4. Check community

A healthy, diverse and harmonious community around the product is essential for Open Source software to continue to thrive.

5. Check documentation

Without any responsible vendor, the documentation becomes a crucial part of support.

6. Consider using a “Magic Box” approach

Try asking, “what if the product was magic?” when working out exactly what you want it to do.

7. Give proactive suggestions to developers, not reactive criticisms

8. Reassess inhouse procedures

Before adopting any Open Source solution, it is important to identify which things the library is doing due to workarounds for the existing software.

9. Look for the “Narcissism of Small Differences”

It is important to identify whether procedures that are different from how most libraries do things are providing essential services to users.

10. Ensure staff understand what developers do and the software development cycle

This is essential so that staff contribute their expertise to the project and do not frame imperfections and glitches in the software as failures. It allows them to see the product as an ongoing work in progress to which they can contribute, rather than an imperfect object.

Conclusion

When asked about their initial decision to develop their own Open Source library software, library developers and managers revealed much about the ways their organisations changed to support this decision. Librarians who were part of a project to specify Open Source software also described many of the reasons why other libraries may want to use Open Source library software. The challenges described, and the solutions proposed, by the interviewees are helpful for any librarian considering adopting Open Source library software.

Acknowledgements

Research for this paper was undertaken in Australia and the United States of America during March and April 2009. It was sponsored by VALA as their Travel Scholar while I was employed by Murdoch University. It forms part of my research for the degree of Master of Information Management at Curtin University of Technology in Perth, Western Australia.

I am extremely grateful for the professional development opportunity from the Travel Scholarship, which allowed me to meet many of my library heroes and learn about much more than just Open Source library software.

References

- Bisson, C., 2007. Open Source Software for Libraries. *Library Technology Reports*, 43(3).
- Blyberg, J., 2007. Mouse bites cat: taking back the 21st-century ILS. In R. S. Gordon, ed. *Information Tomorrow: reflections on technology and the future of public and academic libraries*. Information Today, pp. 31–40.
- Bonaccorsi, A. & Rossi, C., 2006. Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology, and Policy*, 18(4), 40–64.
- Breeding, M., 2008. Open Source Integrated Library Systems. *Library Technology Reports*, 44(8).
- Breeding, M., 2009 History of Library Automation [Library Technology Guides]. Available at: <http://www.librarytechnology.org/automationhistory.pl> [Accessed 22 September, 2009]
- Breeding, M., 2009a. Opening up library automation software. *Computers in Libraries*. 29(2), 25-28.
- Colford, S., 2008. Explaining free and Open Source software. *Bulletin of the American Society for Information Science and Technology*, 10–14.
- Freud, S., 1962. *Civilization and its discontents*, New York: Norton. Available at: http://ia340930.us.archive.org/0/items/CivilizationAndItsDiscontents/freud_civilization_and_its_discontents.pdf.
- Goode, S., 2005. Something for nothing: management rejection of open source software in Australia's top firms. *Information & Management*, 42(5), 669–681.
- Groenewegen, H., 2004. Four decades of library automation: recollections and reflections. *Australian Library Journal*, 53(1), 39-53.
- Lakhani, K.R. & Wolf, R.G., 2003. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *Social Science Research Network Working Paper Series*. Available at: <http://ssrn.com/abstract=443040>. [Accessed 9 July, 2009]
- Lease Morgan, E., 1999. MyLibrary@NCState. In *SIGIR '99 conference*. Available at: <http://infomotions.com/musings/sigir-99/> [Accessed 22 September, 2009].
- Licenses by name|Open Source Initiative n.d, <http://www.opensource.org/licenses/alphabetical> [Accessed 23 October, 2009]

Levesque, M, 2004. Fundamental issues with open source software development *First Monday Special Issue #2 Open Source*. Available at: <http://firstmonday.org/article/view/1484/1399> [Accessed 9 July, 2009]

Melin, J, 1964. *Libraries and Data Processing --Where do we stand?*, University of Illinois Graduate School of Library Science. *Occasional Papers 72*

Morgan, L. & Finnegan, P., 2007. How Perceptions of Open Source Software Influence Adoption: An Exploratory Study. In *Proceedings of the 15th European Conference on Information Systems*. Available at: <http://is2.lse.ac.uk/asp/aspecis/20070050.pdf>. [Accessed 9 July9, 2009]

OLE Project, 2008. *The Open Library Environment Project: A project to reconceptualize technology for modern library workflows* Available at: http://oleproject.org/wp-content/uploads/2008/06/ole_projectdescrip_web1.pdf [Accessed 9 July, 2009]

OLE Project, 2009. *The Open Library Environment Project Final Report (Draft) July 26, 2009* Available at: oleproject.org/final-ole-project-report/ [Accessed 23 October, 2009]

Open Source Definition|Open Source Initiative n.d. Available at: <http://www.opensource.org/docs/osd> [Accessed 23 October, 2009]

Prüfer, J., 2004. Why Do Developers and Firms Contribute to the Production of Open Source Software? *Social Science Research Network Working Paper Series*. Available at: <http://ssrn.com/abstract=590581>. [Accessed 9 July, 2009]

Schneider, K.G., 2009. The Thick of the Fray: Open Source Software in Libraries in the First Decade of this Century. *Bulletin of the American Society for Information Science and Technology*, 35(2), 15–19.

Factors influencing libraries to write their own Open Source

PART TWO - ABOUT YOU

*** 1. I acknowledge that my responses here will not be used until I have read the Information Sheet and signed the Consent Form during the personal interview.**

Yes

*** 2. Please enter your name here:**

*** 3. What has been your primary role in the creation of Open Source Library Software (OSLS) ?**

Initial technical development or coding for OSLS currently being used.

Manager who supported the initial development of OSLS currently being used.

Helping to specify possible future OSLS

Factors influencing libraries to write their own Open Source

PART THREE - WHY DEVELOP YOUR OWN SOFTWARE?

This question asks your opinion about why your library is involved in developing or specifying a new software product - rather than using software that already exists.

Please indicate how much the following factors influenced you:

4. Why develop your own software? - Organisational Factors

	4. Very influential	3. Rather influential	2. Not very influential	1. Not influential at all	0. Do not agree that this statement is true
4.1 There are more library staff with coding/development skills than in 1999	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.2 Non-coding staff can be involved as well eg. specifying, testing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Why develop your own software? - Software Factors

	4. Very influential	3. Rather influential	2. Not very influential	1. Not influential at all	0. Do not agree that this statement is true
5.1 Self-developed software costs less	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.2 More control of features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.3 Easier to modify	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.4 No need for technical or workflow workarounds	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.5 You can work with recent software development tools and methods.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.6 You can make the system interoperable with other systems	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.7 You can build in a mechanism to share data with third parties	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.8 The library will not be locked into a support or maintenance contract	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Factors influencing libraries to write their own Open Source

6. Why develop your own software? - Market Factors

	4. Very influential	3. Rather influential	2. Not very influential	1. Not influential at all	0. Do not agree that this statement is true
6.1 Existing software product used by the library does not meet library needs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.2 No product on the market meets library needs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.3 Software product with similar functionality is too expensive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.4 Proprietary library software vendors have not previously delivered on software requirements	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.5 Vendors have not previously delivered on maintenance agreements	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.6 Vendors have not previously delivered on support agreements	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.7 Vendor software is not developed solely to solve problems but also to attract new customers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.8 Vendor business decisions (like business mergers and acquisitions) impact negatively	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.9 Proprietary products are not developing quickly enough	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.10 Proprietary products are developed using outdated technology	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. If you would like to elaborate on any of the answers above, please do so here.

8. Please describe here any additional influential factors about the decision to develop new software.

Factors influencing libraries to write their own Open Source

PART FOUR - WHY OPEN SOURCE ?

This question asks your opinion about why your library is involved in developing or specifying a product released under an Open Source License.

Please indicate how much the following factors influenced you:

9. Why Open Source? - Organisational Factors

	4. Very influential	3. Rather influential	2. Not very influential	1. Not influential at all	0. Do not agree that this statement is true
9.1 Library staff are more likely to be familiar with tools like MySQL, Apache and PHP	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.2 Library staff already accept Open Source products like Firefox and Linux	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.3 The continued existence of the software does not depend on a single company	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.4 You can choose how the system is supported and maintained	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.5 Library organisational culture now accepts that usable software can be a work-in-progress and not perfect.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.6 Creating good Open Source software enhances the organisation's reputation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Why Open Source? - Technical Factors

	4. Very influential	3. Rather influential	2. Not very influential	1. Not influential at all	0. Do not agree that this statement is true
10.1 Compelled by the license of the development tools to release the product under Open Source	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.2 If others can see the source code, then they can develop it further	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.3 Library staff can "try before they buy" and download and test the product	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.4 Library staff can see all modules and what they do	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.5 You can check that the system is standards-compliant if you can see the code	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.6 Tools like Apache, MySQL, and PHP make it easier to build OSLS than in the past	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.7 The Open Source Licensing model has established success with products like Firefox, Linux, Wordpress	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Factors influencing libraries to write their own Open Source

11. Why Open Source? - Community Factors

	4. Very influential	3. Rather influential	2. Not very influential	1. Not influential at all	0. Do not agree that this statement is true
11.1 Bugs are easily noticed and fixed when many people work on the same product	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.2 There is no charge to contribute to an Open Source project	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.3 The Open Source community is large, mature and sustainable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.4 It will cost less individually if many libraries can pool their efforts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.5 Potential users can make suggestions during development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.6 Potential users can test the product during development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.7 Social software like blogs, wikis and forums make it quick and easy to share progress	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.8 Solutions to problems with Open Source can be found easily online with tools like blogs, forums and online documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.9 Many companies now exist to support Open Source Library Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.10 Open Source support companies only provide support, so will not cut back on support for other revenue streams	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.11 Open Source lets you give back to your community	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.12 You do not have to wait until lots of libraries want the same modification before it happens	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.13 Releasing software under Open Source license fits in with existing library philosophies that our collections are open, available and visible for all to see.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. If you would like to elaborate on any of the answers above, please do so here.

13. Please describe here any additional influential factors about the decision to use an Open Source license.

Factors influencing libraries to write their own Open Source

PART FIVE - CONCERNING FACTORS

This question asks your opinion about factors of concern for your library when developing or specifying Open Source Library Software.

Please indicate how much the following factors concerned you:

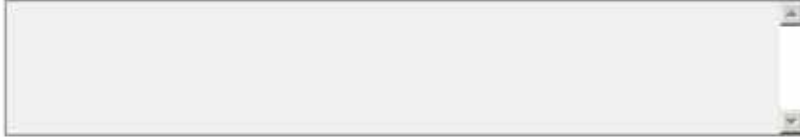
14. Concerning Factors

	4. Very concerning	3. Rather concerning	2. Not very concerning	1. Not concerning at all	0. Do not agree that this statement is true
14.1 Total Cost of Ownership may be more or the same as proprietary software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.2 Open Source Library Software may make the organisation too dependent on the staff member who knows most about it.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.3 It may be hard to hire someone with the right skills to support Open Source Library Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.4 Our library may not be able to afford staffing to support Open Source	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.5 Without a vendor with contractual obligations, then there is nobody legally obliged to make sure the Open Source Library Software works	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.6 There is a much lower level of Corporate Risk when you rely on a vendor that is audited and run on a business model	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.7 Open Source cannot match the sophistication and functionality offered by proprietary software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.8 It is easy to find malicious exploits if you can see all the code	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.9 Your organisation will not earn profits from your development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.10 You need to create higher quality code if you release it under Open Source	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.11 The IT department is not used to supporting Open Source	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.12 Other software used in our organisation is proprietary and incompatible with Open Source	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.13 It is hard to convince other people that Open Source software is not chaotic, anarchic and depends on people doing only what they feel like	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

15. If you would like to elaborate on any of the answers above, please do so here.

Factors influencing libraries to write their own Open Source

16. Please describe here any additional concerning factors about the decision to create new Open Source Library Software.



Factors influencing libraries to write their own Open Source

THANK YOU

Thank you for completing the questionnaire.

I will be following this with a short interview to allow you to elaborate on any of your answers or to identify other concerning or influencing factors not listed here.

If you would like to find out more about my research project, you will find it at:
<http://kathryngreenhill.com/study/>

I expect to submit my thesis by June 2010.

17. Please check here if you would like a copy of the results when they are available.

Yes