OSBR.CA

The Open Source Business Resource

NOVEMBER 2007

NOVEMBER 2007

PUBLISHER:
The Open Source Business Resource is a monthly publication of the Talent First Network. Archives are available at the website: http://www.osbr.ca

EDITOR:
Dru Lavigne
dru@osbr.ca

ISSN:
1913-6102

ADVERTISING:
Rowland Few
rowland@osbr.ca

GRAPHICS:
Ryan May

What is your first thought when you encounter the term "open source support"? A programmer typing the answer to a question using a chat utility? Hours spent scouring the Internet for a working configuration sample? Contacting a support engineer at a commercial call centre? If you find it difficult to think about a support engineer, you're not alone. Actuate's recently published 2007 Open Source Survey of senior personnel from financial services, Telco, and public sector organizations across North America and Europe indicates that 46.3% of respondents cite the lack of availability of long term support as a major barrier to their company's adoption of open source technologies (http://www.actuate.com/info/os-07survey.asp).

My own observations from speaking at technical conferences bear out the survey results. Many personnel, both technical and managerial, are unaware that commercial open source support options even exist. Further, most who have heard of open source support assume it is limited to vendors of Linux distributions. In other words, you're fine as long as you stick to that vendor's packages and don't try to integrate with any other software or operating systems--hardly a realistic scenario for today's complex business needs.

Fortunately for those organizations requiring commercial support, options do exist. Three of the articles in this issue were contributed by companies which provide commercial open source support. Each article focuses on the needs of a particular sector and describes the approach that company uses to address those needs. OpenLogic taps into the open source developer community as well as their own engineers to support complex mixed environments, SourceLabs provides certified stacks to meet the regulatory requirements of financial institutions, and Freeform Solutions is creating a development commons to support niche applications within the not-for-profit sector.

Two of the articles don't deal directly with support, but serve as a reminder that support, and its associated cost, is a piece of a bigger software puzzle for organizations. Glen McInnis argues that the marketing of support, or its perceived lack, is but one of several competitive actions undertaken by business organizations. Coverity's Open Source Strategist describes how the Scan project is increasing the quality and value of open source by reducing the number of defects in the code.

This issue contains a new section in response to feedback from our readers. It is a report on the recent Open Education 2007 conference which builds upon the Open Educational Resources (OER) article published in the September issue of the OSBR.

As always, we look forward to receiving and publishing reader feedback.

Dru Lavigne,

Editor-in-Chief

dru@osbr.ca

Dru Lavigne is a technical writer and IT consultant who has been active with open source communities since the mid-1990s. She writes regularly for O'Reilly and DNSStuff.com and is author of the books BSD Hacks and The Best of FreeBSD Basics.

"Open source has revolutionized the IT industry, both from the vendor perspective and the user perspective. What that means is that a lot of the assumptions that IT has about the way to do projects and work with vendors really need to be reexamined...How do you find the things you traditionally got from vendors in what I call this "unbundled" world?"
Bernard Golden, CEO of Navica

Open Source Software (OSS) has permeated the enterprise. Some organizations still prohibit the use of OSS altogether, but they are unquestionably in the minority. For most companies, it's not a question of "should we use OSS?" but rather, "how and where can we best take advantage of open source solutions?"

As companies have shifted from prohibiting OSS use to embracing it, they must now deal with technical support issues. This article examines the various option available to support companies that use OSS.

Support Requirements

When it comes to support, OSS is no different than proprietary software as the burden of supporting any software falls squarely on the Information Technology (IT) team. The question to the IT team becomes how to best support the OSS in use. Possible options include taking advantage of mailing list support, creating an internal support capability, engaging a consultant, or using a commercial open source support provider.

At the enterprise level, the expense of proprietary software support and maintenance, typically an additional 15% to 25% of the annual licensing fee, is assumed and factored into the cost of the software.

In contrast, one of the more compelling benefits of OSS has been the user's ability to obtain support directly from the development community. In other words, free software and free support, straight from the individuals who develop the software.

Mailing list support, however, requires in-house resources to: (i) identify, define, and communicate a problem in a way that it can be answered by mailing list participants; (ii) recognize whether or not an answer is accurate and implement accurate answers received; and (iii) document the implementation. It also assumes that any answers provided from the OSS developer community will be timely. Mailing lists can be very fast and helpful, but there are no guarantees.

As OSS spreads into broader enterprise usage, the bar for support gets higher. OSS components are often used in mission critical production applications requiring a high level of support, typically 24x7 coverage with well defined Service Level Agreements (SLAs). When you take into account the number of OSS components that are typically used, including the components they depend on, it's not always practical or cost-effective for a company to seek support directly from the developer community.

Enterprises often have several critical requirements for OSS support, including:

- SLAs providing 24x7 support coverage with fast response times to address production issues

- Access to experts for the OSS in use

- System-level expertise to help troubleshoot and resolve problems in mixed source production applications comprised of multiple open source components, custom code and commercial solutions

- Support for older versions due to a reluctance to upgrade production systems to the latest release of an OSS component

- Anonymity regarding the type of software in use and any associated problems

- Necessary fixes are contributed back to the OSS project for inclusion in new releases

- Cost-effective support and the ability to shop for the best service at the best price

Support Options

Support usually means that there is someone who is responsible for resolving problems with the software. Support equates to reduced downtime and higher productivity. A traditional support agreement gives an IT department access to a team of professionals who troubleshoot problems, resolve issues, and provide software updates and bug fixes.

Four options for OSS support exist. The first option is referred to as mailing list support. This option can take the form of posting questions to user forums or browsing Frequently Asked Questions (FAQs) and knowledge bases. The advantages of this option are: (i) it contacts the developers who have contributed to the OSS project, and (ii) it is usually available at no cost. The limitations of mailing list support include:

- There are no SLA guarantees

- Once a problem starts to go beyond the boundaries of that specific OSS package and includes other OSS, custom code or commercial software, the source of the problem becomes more difficult to pinpoint

- The time-sensitive nature of an organization's IT problems makes enterprises reluctant to seek aid from sources unable to dedicate continuous resources until the problem is solved

- Mailing lists may not properly address the confidentiality concerns of large enterprises

The second option for OSS support is hiring one or two of the community's experts as employees or contract workers. Most OSS developers hold software jobs and enjoy being paid to work on their favourite project. However, this is not a scalable solution; hiring a person for every open source project you use, when you use dozens or hundreds of them, is not feasible. Also, many successful OSS projects have fewer than 10 developers, so an expert isn't always available for hire.

The third option is commercial support for individual OSS packages. One of the prevailing questions for developers of OSS is "how do we make money?" and the answer is often "consulting." Typically, developers of OSS form companies that provide training, documentation, customization, and technical support. The support is generally excellent, assuming that the developers have the skills to provide support in addition to developing software. As with proprietary software support, it can be available on an incident-based price schedule or by subscription.

The downside for IT departments is dealing with many individual support providers, as many as one vendor for each OSS project. Moreover, critical issues frequently involve the integration of multiple components and systems which the project developers may not have the expertise to resolve.

The fourth option for OSS support is consolidated commercial OSS support. This new support model has emerged in recent years to provide umbrella coverage for the multiple OSS packages that a company uses. The support is typically available by a subscription contract with the level of support and speed of turnaround determining the price.

With some suppliers, support is limited to a specific stack of software projects, such as LAMP technologies and their supporting components (http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29). Other companies offer more extensive coverage. OpenLogic offers support for any of the 330-plus software projects that can be downloaded from its certified library, and extends the support to customized versions of the software. Companies offering this type of support will typically pay OSS committers and contributors to help resolve issues.

The advantage to enterprise-level companies is clear: around-the-clock quality support is available from a single vendor, regardless of the issue. Another advantage is that support aggregators often have deep experience in many projects and can help with integration issues between OSS packages as well as home grown or commercial projects.

Typical Enterprise Support Problems

In an ideal world, every IT department would have a custom support plan tailored to its specific needs. But if an OSS support provider is to be profitable, it needs to craft a menu of solutions that can be delivered to a wide range of customers. This is not a simple proposition as the typical support problems of an enterprise level company vary widely.

OpenLogic's experience, after working with many enterprise customers, is that fewer than 10% of support issues involve software defects. Most support issues are questions, ranging from a list of simple “how-to”s to complex troubleshooting. Common customer problems revolve around configuration, integration, and performance concerns. The following examples come from OpenLogic and demonstrate the types of support issues enterprises encounter.

A production team in a Fortune 1000 organization needed to deploy a complex security process using an implementation of mod_ssl for Apache. The deployment was scheduled for off-hours and needed to be finished before the beginning of the next work day. It was close to midnight, and the deployment was not proceeding as planned. Within an hour of calling, the issue had been escalated to OpenLogic’s support team; OpenLogic also tapped its “Expert Community,” a group of dedicated contributors and committers to OSS projects. In this case, Covalent Technologies worked with OpenLogic to resolve the issue. Fifteen minutes of troubleshooting revealed the problem: the proper authentication keys were located on a key store server that was on an unavailable network layer. The problem was complicated by the fact that the company was using a customized version of Apache. Work continued and the customer was able to implement a new, reengineered process without experiencing any downtime.

Another example highlights that it’s not always convenient or practical for companies to upgrade production software in order to get critical bug fixes. One customer sought a resolution that wouldn’t force a continual upgrade cycle. The OpenLogic team settled on porting fixes back to the customer’s older version of the software and was able to provide a certified patch within 12 hours.

In the fast-paced financial services industry, IT teams can gain advantages from the flexibility and lower costs of using OSS. But most production applications in financial services incorporate OSS, custom code, and traditional proprietary software. In one case, a customer was on a deadline to upgrade a system that incorporated Hibernate, an open source persistence and query service, along with BEA WebLogic, IBM's DB2, and custom code. An upgrade of DB2 was causing significant performance problems which the customer suspected were caused by Hibernate. OpenLogic worked with the customer to define a set of test scenarios to pinpoint the source of the slowdown. OpenLogic was able to rule out Hibernate as the cause and eventually determined that DB2 logging capabilities had been left on, causing the slowdown.

Making Sense of the Choices

Open source has always been valued for its promise of freedom to: (i) choose the best software and (ii) modify the software as needed. In addition, OSS enables companies the freedom to choose the best support option and provider for their needs. This creates a competitive marketplace for OSS support. Providing a wide range of support options makes it easier for enterprises to use OSS with confidence, but sizing up the choices can be confusing. Before making a decision, an IT department should ask support providers the following questions:

- Which OSS packages are supported?

- Can all packages in use be supported?

- On which operating systems, database, and configurations are they supported?

- Who exactly will be involved in providing the support?

- Is integration with custom or proprietary applications supported?

- Will I need to upgrade to take advantage of fixes?

- Will fixes be contributed back to the OSS project or will I be on a forked version?

- How much will support cost?

If you use OSS or are contemplating its use, the answers to the questions above can help you select the best option to support OSS.

The Future

As more enterprise IT departments experiment with OSS, we can expect to see continued changes in the types of OSS support available. Currently more businesses can benefit from the use of OSS. In this competitive support marketplace, companies now have the freedom to choose the support option that provides the best, most cost-effective service.

Stormy Peters is the Director of Community and Partner Programs at OpenLogic (http://www.openlogic.com). Previously, Stormy worked at Hewlett-Packard where she founded and managed the Open Source Program Office and was responsible for HP's open source strategy, policy and business practices. Stormy is a frequent keynote speaker on business aspects of OSS at major conferences and has addressed the United Nations, European Union and various U.S. state governments on OSS. Stormy is involved in OSS because it is changing the world and the community is full of smart, passionate people. Stormy graduated from Rice University with a B.A. in Computer Science.

"...the use of FOSS does not pose risks that are fundamentally different from the risks presented by the use of proprietary or self-developed software. However, the acquisition and use of FOSS necessitates implementation of unique risk management practices."

Federal Financial Institutions
Examination Council
(http://www.fdic.gov/news/news/financial/2004/FIL11404a.html)

Infrastructure Open Source Software (OSS), including middleware, database packages, and the Linux operating system, is increasingly being deployed by financial institutions. Many OSS packages are selected and incorporated directly into custom applications by developers, thus bypassing traditional purchasing channels and their attendant legal, standards, and technical review processes. Because of this, Information Technology (IT) management is often unaware of the OSS running in their data centers, and sometimes support and maintenance measures are not in place for OSS running in production applications.

With the advent of regulatory structures such as the Basel II accords (http://www.bis.org/publ/bcbs107.htm), the reliability of computing systems is increasingly subject to regulatory scrutiny. Not having adequate support and maintenance measures in place creates a significant compliance risk for financial institutions. This article describes these risks and outlines best practices for an anti-failure program that brings systems depending on OSS packages into compliance while reducing overall operational risk.

Open Source Proliferation

Strategic, forward-thinking IT organizations within financial services typically depend on custom business systems for competitive advantage. Unlike later-adopter organizations, these companies invest beyond the traditional portfolio of off-the-shelf applications, creating custom business systems to enhance their position in the marketplace. In these organizations, time to market and innovative functionality can mean the difference between success and failure of a new product or service. Tighter budgets and outsourcing pressures compound these competitive demands.

In these types of high pressure environments, elite development teams naturally seek out innovative OSS to accelerate time to market and enhance application functionality. OSS packages often evolve more quickly and incorporate more cutting edge features than vendor-controlled, standards-bound software implementations. Some of the reasons development teams choose OSS include:

- Increased flexibility in how problems are fixed, new features are added, and other packages are integrated

- Innovative features: increasingly, OSS has moved beyond commodity implementations, such as Linux, to represent the cutting edge of innovation, such as Java/J2EE technology, where developer-driven innovations have outpaced the vendor-driven standards process

- Reduction of vendor lock-in: developers don't have to wait for vendors to add new features or release a new version

- Worldwide technical community: OSS packages often have a wide universe of users to draw upon for information, instruction and even sub-contractor labour; community hubs such as SourceLabs SWiK (http://swik.net) amplify the utility of these communities

- Investment protection: OSS provides the same or better level of openness and investment protection than proprietary software

- Ready availability from the Internet: anyone can download and use OSS packages or application components

Because of these factors OSS will continue to proliferate, particularly within organizations that create custom software critical to the operation of their business.

Under the Policy Radar

In most financial service organizations, purchasing software invokes well-documented processes intended to protect licensee organizations from operational and legal risk. Virtually every financial services institution has a policy in place that dictates that no unsupported software can be used. Traditional purchasing procedures are critical to the enforcement of that policy. Because OSS is readily available from the Internet, it bypasses these safeguards. Larger packages, such as database management systems and application servers, are difficult or impossible to deploy in the datacenter without oversight from IT operations, and thus their deployments by large financial institutions are still relatively rare.

In contrast, framework packages, component libraries, applications, and tools can easily be embedded in custom software by application development teams without oversight of central IT organizations and attendant operational risk safeguards. These policy breakdowns and compliance failures often come to light only when systems fail, are compromised, or exhibit performance anomalies.

Typical trigger events that bring unsupported OSS to light include:

- Load testing during staging: as an application is subjected to load for the first time, components degrade or fail

- Upgrade of a component: unforeseen failures due to inadequate testing procedures or accountability in the OSS community

- Internal maintenance breakdown: to maintain confidentiality and insulate themselves from legal risk, organizations often maintain their own customized internal version; as the time and personnel cost becomes untenable, systems relying on this forked component are placed increasingly at risk

Special Issues Relating to OSS

Due to the current regulatory environment, issues of systems reliability affect financial services firms more acutely. Increasingly, banks and other organizations are required to demonstrate adequate attention to the operational risk inherent in their computing systems.

The reasons for IT project and business failures include several that are typical of unsupported OSS projects. These reasons include:

- The use of technology in a way or at a scale that hasn't been attempted before

- Lack of measurement and tracking systems, leading to an inability to identify that failure looms or is occurring

While these risks are not new to most IT leaders, the legal and regulatory environment surrounding the financial services industry creates a new urgency as failure to address these risks may result in prosecution and incarceration.

The effects of system failure in financial services can be enormous. While many are hidden from public view, some failures can border on the spectacular (http://tinyurl.com/288agp). Because of this type of exposure, the Basel Committee on Banking Supervision has stipulated allocation of capital to underwrite all operational risks.

While regulatory agency-stipulated timetables for implementing the Basel II accords varies by jurisdiction and type of institution, one of the key risks that must be assessed, tracked and offset with capital is the risk related to business disruption or systems failures. Institutions without adequate software support and maintenance measures in place may face higher capital reserve requirements in order to meet Basel II recommendations.

Managing Operational Risk

No computing system is without risk, and due to its inherent transparency, OSS software has less risk for failure than most commercial software. With strong anti-failure measures in place, IT organizations can take advantage of innovations and the cost and productivity advantages of OSS while reducing operational risk.

Critical elements of an effective anti-failure program include:

- Enforce existing support policies

- Make compliance easy: use tested OSS which the organization recognizes as known, supported and maintained

- Create and foster a culture that values data-driven software testing: testing approaches such as CERT7 Certification (http://sourcelabs.com/?page=software&sub=cert7) provide tests that can be adapted to approximate production environments

- Measure support providers' effectiveness: OSS enables free market competition for support services; leverage this development by comparing service offerings and testing responsiveness and value

The program should also establish appropriate oversight for appropriate risk. Key factors to assess include whether the sytem: i) is accessed by customers or business partners; ii) is revenue bearing; iii) is manipulating critical data; and
iv) its requirements for availability .

For enterprise IT organizations considering use of OSS in production applications, SourceLabs (http://www.sourcelabs.com) provides stacks, or combinations of OSS infrastructure software that have been tested and certified to work well together and be dependable under production conditions.

Conclusion

OSS offers financial services firms substantial advantages, and the right policies allow companies to realize the advantages of OSS without increasing operational risk. Due to regulatory compliance requirements, financial institutions need to ensure that OSS usage is covered by their software risk management policies. Reliable support is critical to the successful integration of OSS. Support vendors, such as SourceLabs, who provide pre-tested stacks and production-grade support options can help mitigate the risk of using OSS by financial services institutions.

This article is based on the SourceLabs White Paper "Special Considerations For Financial Services Firms Using Open Source" (http://www.sourcelabs.com/pdfs/SpecialConsiderationsForFinancialServices.pdf).

"The primary hurdles to increasing adoption of F/LOSS software in the sector are missing applications, lack of information about support options, lack of training and familiarity of nonprofit staff on F/LOSS applications, and, finally, perception... There is still the perception that F/LOSS is harder to support, or that it's not possible to find support for F/LOSS."

NOSI Primer

In 2004, the Nonprofit Open Source Initiative (NOSI) released "Choosing and Using Free and Open Source Software: A Primer for Nonprofits" (http://www.nosi.net/projects/primer). The primer describes the potential impact that Free/Libre and Open Source Software (F/LOSS) might have on the not-for-profit (NFP) sector. In a recently released update, the primer concludes that, despite many advances in the use of F/LOSS in the sector over the past several years, a real and perceived lack of support remains a significant barrier to the increased adoption of F/LOSS by NFPs.

Support is often the last thing people think of when they think of F/LOSS and NFPs. Instead, people tend to focus on the price, which is of particular importance to cash-strapped NFPs. There is a widely held belief that free software represents a cost savings to NFPs.

Major software vendors such as Microsoft, Adobe, and Symantec have software donation programs to support charitable organizations. Some donation programs are also available to support NFPs, schools, and/or libraries. Initiatives such as TechSoup Stock (http://www.techsoup.org/stock) are often used to facilitate these and other donation programs for a small administrative fee. (**Editor's note:** reduced/free licenses for institutions to use commercial software typically do not include support for that software).

Whether the application itself is free or not, the cost of the software is often a small percentage of the Total Cost of Ownership (TCO). Even when donated at low or no cost, proprietary, closed-source software often comes with regressive licensing schemes that can seriously limit an organization's ability to deploy the software effectively. Proprietary data structures and file formats can also be very detrimental to NFPs, since continued access to their data may increase costs later when their upgrade path consists solely of expensive proprietary options.

Support factors into the TCO as well. While licensing schemes and file formats are strikes against proprietary, closed-source software, the availability of support from a software vendor can be a big plus. F/LOSS products often have no vendor, so support can be harder to come by. Support to initially install, and then use, the software is often required for both commercial software and F/LOSS.

Lifecycle Support

There is another element of support that is important to consider: the lifecycle of the software itself. With commercial software, there is a reasonable expectation on the part of purchasers that the vendor will continue to develop the software, fix bugs, and release updates. When it comes to lifecycle support, F/LOSS is sometimes said to have an advantage because of the access to the source code that is inherent to the development model, and the often imagined army of developers around the world who contribute freely to the development of the software.

In terms of lifecycle support, the advantage of F/LOSS is significant for large scale development of widely used utility type software.

Utility type software includes operating systems, common server applications, and common desktop applications. Some paragon examples are Linux, Apache, Mozilla Firefox, and OpenOffice.org. However, the vast majority of F/LOSS is not that broad in its appeal or utility, and the needs of any particular organization, NFP or otherwise, will necessarily include some niche applications.

In the case of niche applications, F/LOSS alternatives may have only a handful or even a single developer. The pool of users can also be small, negating the F/LOSS truism that “given enough eyeballs, all bugs are shallow”. If there aren't enough eyeballs, carrying out lifecycle support on F/LOSS can be exceptionally difficult for the few who persevere to do so. Or worse, the maintainers can give up and abandon the software. In that case, users are left with the source code, so they are not completely lost, but unless they also have access to programming expertise, they might as well be lost. Lifecycle support is a serious issue to consider in the adoption of F/LOSS. The strength of the specific open source community behind the software is at least as important as the strengths of the software itself.

Additional Constraints for NFPs

The issues noted above – cost, licensing, file formats, support, and lifecycle support – should be familiar to users and potential users of F/LOSS. The specific constraints of typical NFP organizations cast these issues in a particular light.

First and foremost, NFPs typically have less money than for-profit organizations with a similar number of staff or similar operational needs. NFPs usually lack a predictable revenue stream. Much of their funding comes in the form of grants, many of which are tied to specific projects. There is tremendous pressure to reduce overhead costs by focusing only on the barest essentials. It is rare for a NFP to have an IT department, or even staff with advanced knowledge of IT. This makes F/LOSS attractive because of the costs, but highlights the support need due to the lack of skilled staff within the organization.

Maintenance of software can be a significant challenge in NFPs, especially the maintenance of niche applications. The challenge is due to the lack of skilled IT staff and/or the focus on project-based funding rather than operational funding. To be successful, software applications deployed in NFPs should be as close to “zero maintenance” as possible. While users may never need to upgrade their copy of OpenOffice.org to remain productive and effective in their jobs, the same cannot be said of niche applications unique to a particular NFP.

NFPs often experience a high turnover in their human resources, especially volunteers. High turnover makes the effective use of software applications extremely challenging as each subsequent round of users only scratches the surface of the system. A particular emphasis on usability and training can alleviate some of these issues. However, due to financial constraints, most NFPs are not interested in spending money on usability. Moreover, training is only feasible if there is an available support organization which can provide it.

Lastly, NFPs almost always lack any sort of Wide Area Networking (WAN) connectivity between their office locations, and/or a Virtual Private Network (VPN) between their staff.

In some cases, there is no real office and everyone works from home, or working from home is a large part of the way the NFP gets things done. Often, the network available to people in a NFP organization is the Internet. This means that Internet-based software, and particularly web-based software, is of singular importance to NFPs.

If an Internet-centric approach is to be successful, all potential software applications must be analyzed from the perspective of how Internet-aware they are. For example, a desktop conference registration database is useful, but not nearly as useful as one that allows many staff from their home offices to access a common database through the web.

In light of these unique constraints that are common to many NFPs — scarce revenue, lack of skilled IT staff, high turnover, and no common network — the only traditional benefit of F/LOSS that aligns with the needs of NFPs is the low cost. With regard to a lack of skilled IT staff, for instance, whether an application is F/LOSS or commercial software is completely irrelevant. What is especially relevant in a NFP context is support. The NOSI primer is correct that a real or perceived lack of support is a barrier in the adoption of F/LOSS, but what is equally important to appreciate is that support is a more fundamental issue for NFPs than for other kinds of organizations.

Technology Assistance Providers

Organizations like Freeform Solutions (http://www.freeformsolutions.ca/) have emerged to provide support and assistance to NFP organizations. Sometimes called Technology Assistance Providers (TAPs), these organizations provide a variety of services to NFPs, often specializing in a certain kind of work.

In some cases, TAPs are for-profit organizations, in other cases they are co-ops, collectives, or, as in the case of Freeform Solutions, they are NFPs themselves. The unifying factor is that they all seek to support NFPs in the use of OSS.

TAPs can help with support at all stages of using software. They are focused on understanding the F/LOSS landscape so they are familiar with the available software applications, and have expertise in installing and supporting them. Some TAPs are also capable of building or customizing particular applications, including niche ones.

At Freeform Solutions, a partnership approach has been found to be the most effective way to provide support. Because of the constraints NFPs operate under, they often do not know what they do not know. They are not only unaware of the available software options, but also of the nature of their problems from an IT perspective.

NFPs, within their domain of expertise, understand their problems and their needs. However, they lack the ability to translate those needs into prescribed software and IT solutions. When NFPs have particular software solutions in mind, it is just as often because someone heard something that sounded interesting, or they read a magazine article about a trend or buzzword, as it is because they have considered the problem space and the available solutions.

To address this, a TAP can closely partner with an organization to understand their needs from the inside out. A TAP that already knows the F/LOSS space and which fully understands an organization's needs can bridge the gap of understanding.

In terms of addressing lifecycle support, a TAP with development expertise is uniquely positioned to provide the kind of reassurance for a F/LOSS application that a commercial development team can provide for commercial software. Just like Microsoft has its own programmers on staff, and IBM provides developers to support Linux and other F/LOSS projects, TAPs can provide some development resources to F/LOSS projects. This has many beneficial effects, including improving the TAPs' ability to provide lifecycle support for a client, as well as improving the success of the particular F/LOSS project, which in turn should attract more developers to that particular open source community, which in turn improves the lifecycle support generally available for that F/LOSS project.

However, unless a TAP has some independent revenue stream of its own that it chooses to expend on a F/LOSS project, it is not feasible for a TAP to devote a great deal of development effort to a particular F/LOSS project. Development is expensive, and one of the reasons NFPs are interested in F/LOSS in the first place is the low cost. It is unreasonable to expect that the money paid by a NFP to a TAP would be sufficient to fund significant development efforts, unless the TAP has many clients using the same software.

Development Commons

When a TAP supports many users of a given software, that TAP can leverage the small contributions of all clients towards the development of additional software. TAPs that pursue this approach can be especially valuable to NFPs because their development effort translates into significant expertise in the software, expertise now at the disposal of the NFP, without the organization having to fund development.

In essence, a TAP so engaged is an economic manifestation of the open source philosophy of sharing, and the clients are all participants in open source through their relationship with the TAP.

At Freeform Solutions, this approach is dubbed the “development commons”. A particular area of our focus is the development of a generalized software toolkit that can be configured to provide a wide range of applications on the same code base. This allows a supplier to meet the needs of many clients, including those with niche requirements.

This approach is also effective at addressing the “zero maintenance” requirements of NFPs. Because the development effort is shared among many clients and the larger community around the F/LOSS project, clients can expect periodic updates even when they are making no direct contributions themselves.

The basis of the generalized toolkit is the abstraction of many applications so they can be described functionally as a series of forms and reports tied to a database. The toolkit allows for the creation of forms and reports without the need to custom program behaviours and logic into the application. When new needs arise for which there is no configuration option, we add that capability to the code base; all other applications based on the same system benefit from the addition after the next upgrade.

For example, imagine a conference registration system that uses one form to be filled out by conference registrants. Several reports may be necessary, such as a list of registrants' names for creating badges, a payment report for accounting purposes, and a summary of food choices for use by meal planners. Each report is simply a different view of data submitted to the same form.

The generalized toolkit provides the capability to create the form by specifying the questions required, and what type of form element, such as a textbox or dropdown list, each question uses. The logical and behavioural properties of the form and the questions can be specified, such as "visible only to accounting staff". The toolkit includes the ability to build reports with a graphical user interface which doesn't require knowledge of SQL, and which allows for customizing the appearance of the results. The end product is a series of screens which can be tied into a menu or any other navigation structure that makes sense given the particular workflow of the application.

A key concept of this software is the distillation of niche application requirements into generalized capabilities. So rather than custom coding a part of a form to be visible only to certain people, the capability is added to make any question visible or not visible to a pre-defined group of users.

We have found this approach to be effective at empowering NFPs to use software effectively, and for enabling the rapid deployment of services. This approach is cost effective for deploying and maintaining services to NFPs, since it minimizes the custom work required, and standardizes the deployments on a common code base. The administrative interface for the generalized software toolkit can be used by NFP staff, without the need for them to operate at the API level of many programming abstractions, which would limit their usefulness to programmers only. Since NFP staff have access to the same software tools that Freeform Solutions itself uses, through an interface they can learn and become comfortable with, it enables a different kind of "zero maintenance" for NFPs. Although staff time is not free, it is in some cases a more affordable alternative than paying money to an outside group for maintenance.

Conclusion

Support, encompassing traditional installation, desktop, and software lifecycle support, is a significant issue for NFPs. The emergence of TAPs has helped mitigate the installation and desktop support issues. Certain kinds of TAPs can also provide lifecycle support, which is equally necessary for successful adoption of F/LOSS, especially the more specialized, niche applications. The NOSI primer has correctly identified that support for F/LOSS continues to be a barrier to adoption. However, various ways to overcome that barrier now exist. As TAPs become more experienced in serving the NFP sector, the end result can be a win-win for both NFP users and F/LOSS projects as the user base and NFP contributions to projects increase.

Jason Côté is the President and CEO of Freeform Solutions. He has held senior positions at CANARIE, as well as chief executive positions at Actua, and CanadaHelps, all three of them leading national not-for-profits that excel in the use of information technology. Today, Jason oversees all operations at Freeform, and works with clients and staff to build the IT capacity of the not-for-profit sector. He holds a Bachelor's degree in Computer Systems Engineering and an Executive MBA in Leadership.

Julian Egelstaff is the Senior Director of Consulting Services. Prior to co-founding Freeform Solutions, Julian held a variety of positions in project management, documentation, and internal tool development at Corel and later Cognos. Today, Julian oversees all consulting projects, manages client relationships and coordinates Freeform's open source development efforts. Julian has 8 years experience in PHP development, and is a ZCE. He holds a Bachelor's degree in Journalism and Philosophy, and has completed some graduate studies in Cognitive Science.

"For years the software industry has largely competed on the basis of symmetry: Oracle versus IBM in databases; BEA versus IBM in application servers; etc. Feature wars, price wars, but not true competition wars. That is, competing by playing a different game, with different rules. Open source enables an alternative battleground upon which to compete, with community, code, and culture the new competitive tools."

Matt Asay, VP Business Development at Alfresco

Early in 2006, Microsoft CEO Steve Balmer made a public statement that Microsoft had no trouble competing with open source rivals on features (http://tinyurl.com/227wbf). Balmer was downplaying the fact that Open Source Software (OSS) has changed the nature of competition in the software industry from one of feature-based competition to a much more complex model for competition.

Traditional firms like Microsoft are not simply competing against a loose group of programmers who produce OSS; they are competing against other firms who have chosen to use OSS as part of their product or service offering. This article is an examination of competition in this new environment where firm-to-firm competition includes those firms making use of OSS.

The evolution of the marketplace becomes evident when competition is viewed as a series of small, but discrete actions that are carried out by a firm. Each of these actions is intended to improve the firm's standing in the marketplace. Such actions can be classified as one of six different types of competitive action: (i) pricing, (ii) marketing, (iii) new products, (iv) capacity, (v) service, and (vi) signaling. Through an examination of each of these types of actions, a clearer picture of competition emerges.

Freedom in Pricing

One of the common misconceptions about OSS is that it must be offered free of charge. This is not true as most open source licenses allow copyright owners to sell code, and the more permissive licenses allow anyone to charge for the code. This has given rise to the creation of commercial open source such as the commercial database provided by MySQL AB and the commercial content management package offered by Alfresco.

Traditional pricing models for software would fall into one of three categories: (i) licensed software, (ii) leased software, and (iii) Software as a Service (SaaS). Licensing models are the dominant form of software pricing, but all of these pricing models share common characteristics. Buyers and vendors are comfortable with these models as they are well understood and provide for predictable revenue and expense. Even SaaS models often include a component to cover software licensing costs.

By using the resources of OSS projects, firms are able to invest less in Research and Development (R&D) while closed source competitors are forced to expend ever increasing amounts on the resources required to develop a commercial software package. By charging less, or nothing, for the software and instead charging for value added services, OSS firms are better able to reach customers who could not have otherwise afforded the same level of service or customization (http://tinyurl.com/37bw9m). This allows OSS firms to shift focus to providing better customer service instead of investing in expensive R&D.

As a class of software applications matures into a commodity, the large investment in code R&D eats at the profit margins of large firms. This maturation, combined with OSS pressure, may be spawning price wars that will force many firms to reevaluate their place in the market.

This is most evident in the database segment of the software market. The bottom-up pressure coming from OSS database players MySQL and PostgreSQL has resulted in aggressive pricing by Microsoft to lure high-end Oracle and IBM customers to the Microsoft camp. While Microsoft may have a temporary refuge by moving up the market, Oracle and IBM are already at the peak. The question remaining is how long any of the proprietary database competitors can hold off the OSS alternatives when what most customers require is now considered as basic database functionality.

Support as Marketing

When marketing an OSS product, many of the tried and true methods still apply. However, OSS does change the playing field in a number of important ways. Consider the way in which OSS products change the method to reach an audience of potential customers and how an OSS product is positioned relative to a commercial competitor.

In their 1981 book, Positioning: The Battle for your Mind (http://tinyurl.com/yq4ody), Ries and Trout counsel that any marketing action should attack a competitor's weaknesses. For a proprietary competitor, that weakness is the large investments in R&D, sales, marketing, and other fixed overhead. OSS based firms attack this weakness by positioning their solutions as open, accessible alternatives to the "big brother" approach.

Closed source competitors respond in one of two ways: by attacking the feature set of the OSS product and by attacking the OSS product based on the lack of commercial support.

Unfortunately for the proprietary firms, the rapid evolution of software has resulted in a myriad of features while there is an increasing trend to start with a basic product and customize the rest to suit the unique needs of the purchasing firm. The attack on the support of OSS used to be valid, but we have since moved beyond that point. With large reputable firms backing OSS products, long-term, reliable support is available for costs similar to proprietary products. For the OSS firm, it is important that their messaging convey this changed reality, in cases where the OSS firm can provide a more customized solution than the proprietary competitor with the same level of support.

New Products

Market entry for OSS based firms may be easier when compared to proprietary firms. Entry into a new technology market requires reaching early adopters, those leaders who are willing to take a risk as they see the potential in the product. While marketing to these potential customers, the OSS firm is also reaping the benefits of attracting potential contributors to the project.

The two strongest motivations for OSS contribution are a need-driven desire to solve a current problem, and participation in the community in order to gain status and recognition. When the OSS firm can successfully convey the benefits of their product to the community, their marketing efforts reap the benefits of attracting both potential consumers and customers who will assist in the co-development of the solution.

Capacity

For commodity-type businesses, decisions about capacity expansion typically involve the commitment of resources based on expectations of future conditions ranging from future demand to the competitive environment.

In the software industry, capacity actions are often taken to increase the speed of release for a software product or to undertake the R&D required to launch a new product. OSS communities tend to produce a wider variety of applications under a faster release cycle than their equivalent proprietary competitors.

Perhaps even more significant, a large portion of new OSS capacity is added outside the project's boundaries. For the OSS firm, much of the effort required to add capacity, either in the form of faster release cycles or new products, is in influencing the governance structure of the OSS projects they are utilizing as part of their offerings. This effort may include paying contributors, but it is more often about leadership actions taken within the community to set direction, address outstanding quality issues, and attract new contributors to the project. The attraction of new committers to the project adds both development capacity and the opportunity to reinvigorate innovation within a community.

Specific actions that can be taken by the OSS firm to assist and provide leadership to the OSS community include:

- Contributing missing components and documentation
- Providing development infrastructure for the OSS project
- Acting as an external quality assurance team
- Providing organizational staff to coordinate complex cross-community efforts such as standards development
- Providing direct monetary support through sponsorship of events, or funding for purchase of other software and equipment
- Releasing existing propriety code to the OSS community

Service

A recent IDC software industry forecast projects that over the next five years, open source services will grow at a rate three times that of the overall IT services industry (http://www.idc.com/getdoc.jsp?containerId=206681). IDC attributes this rapid growth to an increased awareness among customers and solution providers of the possible economic benefits of OSS alternatives. While this may be true, there are some telling differences in how an OSS firm provides services compared to proprietary competitors.

Open source service takes a number of forms. The most common forms are the: i) stack integrator, ii) support provider, iii) systems integrator, and iv) consultant. The latter three are commonly provided through a professional services division or a network of partners.

Where the major difference arises is in the area of stack integrator. Stack integrators take many disparate OSS applications and combine them to form a single cohesive solution. These stacks can be assembled on a client-by-client basis or productized and packaged.

OSS firms develop expertise to evaluate and incorporate OSS as part of a larger solution, allowing the best combination of products for a particular client.

The proprietary competitor is often limited to a selection of in-house developed applications, largely as a result of a business model driven by sales of software licenses. If a services firm is making use of proprietary applications, they are further limited as they can only perform integration to the level that is exposed by the commercial product's Application Programming Interface (API). The OSS firm has no such restrictions and can offer seamless integration.

Signaling

Marketing signals are an action by a competitor to provide direct or indirect indication of its intentions, motives, goals, or internal situations. The signaling action will often precede the actual action or, in the case of signaling being used for misdirection, the inaction. Traditional views hold that marketing signals can result in a preemptive advantage and may be expected as part of an industry norm. The most commonly discussed risks of signaling are the revelation of too much information to a competitor, product line cannibalization, reduced reputation, and a perception of anti-trust behaviour.

OSS firms reap the same positive advantages of signaling as their proprietary competitors. However, they are as not as susceptible to the negative aspects of signaling. Consider that an OSS product's code base is maintained in a publicly accessible repository; that repository and code serves as the basis for OSS product path signaling. This complete revelation of the software product diminishes the discussed risks. Similarly, as the OSS firm reveals everything about the product as part of their business model, there is no worry of over-revelation to a competitor. Instead, the firm gains the trust of users and consumers by openly sharing its intentions and plans for the product.

Conclusion

OSS firms have access to a wide variety of competitive avenues. They can opt to compete in the same manner as the traditional software firms; however, the use of OSS allows these firms to more safely engage in price wars, reach an unserved customer base, turn customers into employees, provide a superior level of service and support, and remove from customers' minds any fear of vendor lock-in.

The challenge facing an OSS firm is how to best utilize the external resource that is the OSS community. Treading a careful line between consumption and contribution to maintain a productive community is an activity that requires finesse and a form of management that is still evolving.

Glen McInnis is the Practice Area Lead for Enterprise Content Management at non-linear creations in Ottawa, ON (http://www.nonlinear.ca/). Glen holds a B.Sc. in computer science and is currently completing a thesis titled "Competitive Actions of Open Source Firms" as part of the M.A.Sc. in Technology Innovation Management at Carleton University.

"Finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase."

Software Defect Reduction Top-10 List (http://www.cebase.org)

Open Source Software (OSS) has been embraced by individuals for decades, but only recently have organizations around the globe looked upon open source as an attractive and practical alternative to proprietary software. In addition to its appealing price tag, usually free, OSS can be inspected, modified, and freely redistributed according to the terms of its license.

In spite of this, open source developers still find their code quality and security challenged by a question that has followed them from the beginning: "If it's free, how good can it be?"

Scanning for Defects

The short answer to this question is "Quite good", thanks to the determination of open source developers and the success of the Scan project (http://scan.coverity.com/) a collaborative venture between Coverity, Inc. and Stanford University.

Launched in early 2006, Scan is a key component in the "Vulnerability Discovery and Remediation Open Source Hardening Project," a three-year initiative funded by the U.S. Department of Homeland Security to identify and correct vulnerabilities in widely used open source projects.

The goal of Coverity's Scan is to reinforce the security and improve the overall quality of OSS. By applying the latest innovations in automated defect detection, the Scan site has uncovered some of the most critical bugs in OSS.

The technological underpinning of Scan is Coverity's Prevent SQS (Software Quality System) solution, which automatically identifies and helps developers eliminate defects in source code. Using static source code analysis, Prevent allows developers to find and fix defects at the earliest stage in the software development lifecycle. In static analysis, the code being analyzed is not executed; therefore, test cases and specially designed input datasets are not required and examining code for defects is not limited to lines of code that are run during some number of executions of the program.

Static analysis provides a comprehensive examination of all lines of code in a given codebase, and all of the different paths through that code that can be triggered by varying application input. Coverity Prevent SQS is a robust static analysis solution that pinpoints buffer overflows, memory allocation bugs, and other vulnerabilities that become a target for malicious hacking attacks. It also reveals quality defects that may become larger issues over time, such as insufficient checking of error codes.

Hunting for these types of defects in software code is a meticulous and time-consuming process that most software developers would prefer not to do manually. Prevent automates this process, providing workflow that allows developers to assign and monitor software defects and overall code quality. Despite these obvious benefits, initially we weren't sure what reaction we could expect from the open source developer community.

The initial plan was to perform daily security audits of approximately 40 of the most popular open source software packages, including Linux, Apache, MySQL, Sendmail, FreeBSD, Mozilla, and Samba.

Stanford University would perform a detailed analysis of the results and maintain a database of the findings. From the start, we intended to make the results available to each project's developers; how they chose to respond was up to them.

At Coverity, we had performed earlier, non-government funded scans of the Linux kernel and MySQL. The developer response was promising. For instance, the original 2004 study of the Linux kernel revealed five file system buffer overrun conditions and one network buffer overrun condition, both of which were considered serious defects. The 2005 study, performed six months later, showed zero defects of the same type: all had been resolved. Even though the size of the Linux kernel had increased significantly during those six months, there was a significant decrease in the number of potentially serious defects, thanks to the response of Linux developers.

An Auspicious Beginning

In the initial analysis, we scanned more than 17.5 million lines of code from 32 open source projects. On average, we found 0.434 bugs per 1,000 lines of code which prompted a response from developers of other open source projects. More than 200 developers registered for access to Scan's online database the week after we published our initial analysis results.

Over the next seven days, more than 900 defects were resolved, or an average of more than 5 bug fixes per hour. After one week, defect density for the same 32 projects dropped from 0.434 defects per thousand lines of code to 0.371. Samba, the popular program that allows end users to access and use files, printers, and other resources on a company's network, showed the fastest developer response, reducing their defects from 216 to 18 in the first seven days.

"Coverity found bugs in parts of Samba that we had previously considered completely robust and tested," said Jeremy Allison, head of the Samba development team. "Coverity is making a major contribution to the code quality of the Samba project."

The Amanda project was another clear-cut success story. Amanda is the Advanced Maryland Automatic Network Disk Archiver, a backup system that allows the administrator to set up a single master backup server to back up multiple hosts over the network to tape or optical media. The initial scan of Amanda's code revealed 108 defects. Within the first week, Amanda's developers resolved all issues, a fact that was verified by a scan revealing zero defects.

Also within the first week, Scan revealed a major security vulnerability in X Windows software, a graphical windowing system used in most distributions of Linux and Unix systems. The vulnerability would allow any user with a login to execute arbitrary code with root privileges or cause a denial of services with root privileges. X.org, the developers of X Windows, responded immediately by issuing a security advisory which included a patch to fix the problem.

Scan Celebrates its First Birthday

By the end of the first year, Coverity had witnessed some remarkable successes in the field of OSS development: Developers fixed 6,132 software defects across 53 open source projects, including 13 projects that remedied all outstanding defects. Hundreds of developers were using Scan's analysis to improve their projects and many others had contacted us asking to be included in the Scan. Based on demand from the open source community, we decided to expand the program.

On the first-year anniversary, March 6, 2007, we unveiled the expansion of the Scan project. In addition to a site redesign, 100 new projects were added. More information was made available for developers and others interested in understanding what Scan is and how developers use it.

We also put a new framework in place to help open source developers learn how to use Scan results by gradually introducing them to more advanced features of Coverity's Prevent SQS solution. Projects that actively use the Scan results became eligible to move up a ladder of 'rungs' and receive access to additional functionality. Finally, within the new framework of the Scan Ladder, additional analysis results that were not enabled during Scan's first year were made available to the developers. The response has been overwhelmingly positive.

The Scan Ladder Grows

Because of the extraordinary successes we've seen through Scan, Coverity has invested in dedicated resources to the project beyond the requirements of its contract with the U.S. Department of Homeland Security. The volume of requests for access to results and for the inclusion of additional projects has shown us that the open source community recognizes the benefits of static analysis.

In addition to adding many new projects, we've also included projects outside the scope of critical infrastructure originally defined by the Department of Homeland Security, since preventing crashes and data loss are obviously worthwhile contributions in other code bases.

These new projects come from a wide array of OSS categories, from developer utilities to graphic tools.

The projects are displayed on the Scan Ladder (http://scan.coverity.com/rungAll.html), organized by the degree of experience the developer has built with the project, their communication, and their progress in addressing the issues found by the analysis. Currently the Scan Ladder consists of two rungs:

- Rung 0: the first rung includes projects that have been built and analyzed, but representatives of the project are yet to register for access to the results

- Rung 1: once a project provides a set of official contacts to represent the project to Coverity, developers have access to a mailing list designed to facilitate the discussion of results and questions surrounding Scan and Prevent SQS functionality

In the near future, a number of projects are poised to progress beyond the first rung, as 14 of the original Scan member projects successfully reached zero defects within their first year.

There are currently 265 projects on the Scan ladder. We have found that most of the open source developers don't need much encouragement to participate in correcting security and quality defects in their code. Open source developers take a lot of pride in their code, which—being open—is already subject to public scrutiny. They tend to be quick to fix issues that have obvious consequences, and many of them want to fix defects that may have potential future consequences.

It should be noted that Scan is not the only way these projects check for bugs and defects. Each project has its own methods, and several have some form of regression tests, as well as development and release branches and a formal release engineering process.

Looking to the Future

The Scan site currently analyzes over 32 million lines of code daily. There have been more than 7500 defects fixed in open source projects since the Scan project started, which equates to more than one bug fixed every two hours.

Since Prevent points directly to the root cause of a problem, it's difficult to say how much developer time would have been required to identify and fix these problems if they had been manually tracked down by examining reports of the bug's effects. End-user reports take time to process because of the difficulties in getting clear explanations of the problem, its context, and then having a developer duplicate it. For example, notoriously subtle heisenbugs which appear in the binary but not under debug mode, can take weeks to track down.

According to some people, OSS shouldn't have any bugs because the source code is public, and so many people can look at it. That theory doesn't take into account that many of the "eyes" don't belong to programmers with domain-specific knowledge and the interest to spend time working on that code.

Open source developers regularly give me feedback about how useful the Scan service is, so I have no doubt it is contributing value to the projects. Individuals have reported that their coding methods have changed as they know that certain sloppy programming habits will be called out by the analysis, and they feel that they've become better programmers by correcting these habits. Scan developers have seen what Prevent can identify through static analysis on their code. As a result, when the defects identified at the Scan site have all been fixed, they have a higher degree of confidence regarding the security and quality of their code.

The goal of the Scan project is simple: to further improve the quality and security of OSS. If, in doing so, we can better understand how the combination of our technology and the open source development model leads to defects being fixed at such a tremendous pace, we believe that we'll be able to apply what we've learned to improve the development process and security of all software, open source and proprietary. So far, Scan has taken us a long way towards that goal.

David Maxwell is Coverity's Open Source Strategist, and is tasked with the continuation and expansion of Coverity's DHS-sponsored open source scans. An open source security specialist, Maxwell has over 20 years of experience as an open source user and developer, and he is particularly active in the NetBSD community. He currently sits on the advisory board for the BSD Certification Group and the program committee for the annual BSDCan conference. He was also a NetBSD Security Officer from 2001-2005 and a contributor to the best-selling O'Reilly title "BSD Hacks." Maxwell has previously worked as a lead kernel developer for Nokia, and architected the Internet Service offering for Fundy Cable in New Brunswick.

"OERs will really start to succeed when they can augment our experience of the learning space that is the entire internet, instead of sitting off to the side and requiring learners to self-identify that they want an OER."

Scott Leslie, researcher
(http://www.edtechpost.ca/wordpress/2007/09/27/oer-client-tools/)

Open Education Resources (OER) are educational material and resources that the general public can freely use for teaching, learning, and research. Like Open Source Software (OSS), OER contain content that is freely reused and redistributed without the traditional restrictions imposed by copyright. OER also includes the tools used to develop, improve, and distribute this content to communities.

The Open Education 2007: Localizing and Learning conference was held at Utah State University in September (http://cosl.usu.edu/events/opened2007). The conference focused on the ability of people to learn as simply publishing OER content online does not guarantee that it can be effectively used for learning. This theme was investigated from two perspectives. The first was from the developer perspective with the sharing of efforts to localize the educational content and make it more relevant to the people using it. The second was from the user perspective where organizations utilizing or deploying OER spoke of their experiences, challenges, and successes.

This report provides an overview of the sustainability, localization, technological, legal and interoperability issues raised at the conference. It concludes with the authors' views regarding the future direction of OERs, based on their research being conducted at Carleton University's Technology Innovation Management program.

OER Sustainability

OER localization and sustainability were central to many presentations. Many of the represented OER projects are suffering from sustainability uncertainty, either on the education front lines or in application development and enhancement. Most organizations developing OER are dependent on grants and institutional funding for their survival, and admit that this is not a robust sustainability model.

The authors argue that a first step to building OER sustainability is developing a thorough theoretical understanding of OER and the issues impacting those participating in the OER community. Yet, only three presentations were primarily based on academic or management theory.

The first was a presentation of a grounded qualitative study into the benefits for professors who develop Open CourseWare (OCW) content. The researchers uncovered a wide variety of benefits including increased exposure, improved networking, increased class enrolment, and easier curriculum dissemination. The study represents a first attempt to perform a rigorous study on the topic of OER adoption motivations using a generally accepted research methodology (http://www.51weeks.com/events/3/presentations/26).

The second theoretical study used discourse theory (communication analysis) to analyze the different ways in which two OER communities, MIT and UK Open University, convey their goals and objectives to different audiences. The findings show that the messages conveyed to the faculty and students developing and using OER are very different from the messages conveyed to the universities' upper leadership.

This research suggests that the current sustainability of these two programs is precariously dependent on very different messages being conveyed, and accepted, by different parties who have varying interest in the future of the OER projects (http://www.51weeks.com/events/3/presentations/35).

The third theoretical presentation was given by the authors of this article. Their conference paper was co-authored with Steven Muegge and can be found at http://www.51weeks.com/events/3/presentations/38.

There were numerous non-theory presentations which addressed sustainability from a more practical point of view. Continuously changing intellectual property issues such as copyrights and licenses were carefully considered. Closely related were problems surrounding standards for content and content management. These issues are becoming more significant in the OER domain as localization and adoption in developing countries exposes new cross-jurisdictional and technological difficulties.

OER Localization

Many individuals and organizations presented either their efforts to use and localize OER in their regions or expressed their desire to investigate OER to support their particular education efforts. Entities such as the United Nations (UN), non-governmental organizations (NGOs), and governments of developing nations figured significantly throughout the conference and there was a keen interest in localization efforts in developing regions of the world. The problems associated with localization in both the developed and developing world has also been a focus of recent efforts to enhance the applications that support OER.

A wide variety of issues were raised surrounding the roadblocks impacting the ability to localize OER. Two key sets of roadblocks were addressed, namely the physical barriers due to ineffective technology and the lack of institutional infrastructures, and the legal barriers created by intellectual property protections.

It is very difficult to engage groups in using OER if the content is not sensitive to the language, cultural, and political differences between OER developers and users. When content must be localized, who is ultimately responsible for that localization? Should the developers be the ones who adapt it to the different markets they are targeting or is it up to the users in the market to adapt the content themselves?

Some argued that content must be intelligently adapted by developers if they want to see content adoption by the OER community. Others argued that it is impractical for the developers to be sensitive to all of the issues important to a particular market. Allowing users in the market to adapt the content engages and empowers them to develop and adapt OER on their own, therefore increasing the likelihood of the content being effectively used.

Underlying Technology

Many NGOs expressed a keen desire to see continued growth of open content being produced in the OER community. Their main concern with delivering education to developing countries is the difficulty in making OER available in areas lacking significant technology infrastructure. Many of the representatives insisted that the OER community must solve the issues associated with rights and licenses and make it transparent for OER users.

The proliferation of OER technologies and licenses presents major challenges to the ultimate sustainability of the OER community as the content becomes less interoperable. Similar to OSS, different OER content licenses can be incompatible with other licenses. This creates difficulty when remixing content to meet specific educational or localization requirements when a single source does not suffice.

Making a general repository for OER also becomes complicated since a system must clearly track and convey to the user the license associated with a particular piece of content. Some OCW systems attempt to address this tricky problem by allowing every piece of content, paragraphs or even individual words, to have an accompanying license placed in metadata so that the license can follow the content as it is reused and remixed.

Ultimately, license incompatibilities prevent the remixing of content, regardless of how smart a particular content management system may be. One solution is to encourage authors to put their work in the public domain. However, this is not always possible as some jurisdictions, including the EU, do not allow citizens to put anything in the public domain as that right is reserved for government institutions.

Creative Commons licenses (http://creativecommons.org/) allow authors to give users the same rights as if the content was in the public domain, but with some limitations, such as attribution, while avoiding the EU problem. Unfortunately, commonly used share-alike clauses still make it almost impossible to adapt content to meet the license limitations of other content.

With several issues at stake – education, empowerment, gender equality – the NGO representatives felt it was vital to use OER without concern about which content is suitable for use. In other words, it is their responsibility to overcome the physical barriers while it is the responsibility of the OER community to find a solution to the legal barriers associated with rights and licenses.

The following NGOs shared their experiences using OER to achieve their goals:

- Youth Managed Resource Centers (http://cosl.usu.edu/projects/ymrc/), a provider of technology and computer skills to rural communities in Nepal; given their limited resources, they were interested in OER that could be used offline and independent of Internet connectivity

- Teachers Without Borders (http://teacherswithoutborders.org/), an international NGO devoted to closing the education divide, is trying to harness the potential of OER as a tool to supply their volunteer teachers and learning centres with appropriate, free and customizable content

- Teacher Education in Sub-Saharan Africa (http://tessaprogramme.org/), a research and development programme creating OER and course design guidance for educators working in Sub-Saharan African countries

Numerous OER applications were shown. OpenCourseWare (http://ocw.mit.edu) represents one of the largest concentrations of OER content, but the content is not always easy to remix. This can be circumvented by downloading the content and modifying it in a manual way, but this is inefficient.

To address this, OpenCourseWare In Motion (http://www.ocwinmotion.com) harvests OCW-based eduCommons (http://cosl.usu.edu/projects/educommons) course content and allows the import of selected portions into a wiki system. Through the wiki, users can modify content and allow other users, such as students, the ability to further modify, extend, or discuss the content. This technology also benefits from other wiki capabilities including basic user administration, tagging and tag clouds, and the ability to effectively build links to navigate the content.

WikiEducator provides a collaborative system for developing course content (http://www.wikieducator.org/). Unlike OCW systems that often employ a rigorous interface to collect and structure the content, WikiEducator was designed to provide flexibility to authors to create content and to simplify methods to remix and discuss. Sufficient structure is supplied so the author can identify consistently formatted objectives and activities for the students to use as learning objectives for the material.

There were presentations by numerous universities, such as Utah State, Penn State, and Yale, demonstrating their different implementations of OCW or other competing OER deployment technologies. Many of these organizations gave valuable insight into their particular successes or failures in trying to implement and localize their own OER systems.

There were a number of efforts to connect this growing supply of OER content and capabilities to make it easier for individuals to find the specific content, irrespective of the content's location. Google is participating by providing support for searching, OER content organization, and means to identify licensing.

Presentations from the National Repository of Online Courses or NROC (http://montereyinstitute.org/nroc/) and Flat World Knowledge (http://flatworldknowledge.com/) showed promising efforts to provide micro-publishing support for OER content. Basically, the goal is for professors and students to be able to pull together the content they desire and then have it printed and bound like a textbook, but at a fraction of the cost since the content is freely available.

Rights and Interoperability

Another solution proposed at OpenEd2007 is an Open Education License or OEL (http://opencontent.org/blog/archives/355) that confers all of the advantages of the public domain without any of the restrictions of other licenses. Most importantly, this will make OEL licensed content completely compatible with other licenses while meeting cross-jurisdictional requirements of not waiving rights to content. Granted, when OEL licensed content is mixed with other licensed content, the product will fall under that latter license, but the original content will still remain freely useable by anyone. This will also open the door to contributors in jurisdictions that do not give the right to put content in the public domain.

Standards and Interoperability

At the moment, interoperability is essentially limited to content within a particular content application or OER project. The development of standards to support interoperability is underway in a number of projects, including Curriki (http://www.curriki.org/). These efforts attempt to address broad issues related to the metadata problems associated with OER, such as technology frameworks and rights alignment.

There are fears that if the OER community does not begin adopting technical standards, OER will continue to evolve as a collection of incompatible content that is as collaboratively limited as traditional web content.

OER must also differentiate from regular web content by supporting educational standards meeting the specific learning objective standards set at the state or federal level. The benefits to OER projects adopting standards are twofold. First, it will enhance interoperability; however, the proposed standards are for the construction of content that meets learning objectives and is not inherently concerned with meta-data, licensing, and the other limitations of technical and legal interoperability. Secondly, meeting educational standards increases the ability of OER to be adopted as approved curriculum in the jurisdictions to which the standards apply.

Unfortunately, the benefits of OER meeting government educational standards are not practically realizable at this time. There is a standards framework in place in the US, for example, but only one state has adopted the standard. However, there is great potential in the standards and there are efforts to adopt and implement them by many state governments.

Insights

OpenEd2007 was successful at educating people and organizations from different disciplines about how to take OER to the next step in different environments. All conference participants gained new insights about the potential for widespread education enabled by OER.

Localization was highlighted by powerful presentations from NGOs working in developing nations. These demonstrated that progress is being made to meet the goals of the United Nation's Education For All (http://tinyurl.com/yt5rjk). However, all participants in the OER domain must overcome physical, technological, and legal barriers to localization. There is also the question as to what roles each organization plays in localization efforts and who is responsible for overcoming these barriers.

Academic and management theory can be used to assess the current sustainability problems and suggest possible solutions. Intellectual property concerns were also central to the sustainability discussion. The growing number and nature of licenses is creating problems for interoperability and compatibility between education sources. On a larger scale, cross-jurisdictional problems regarding open content require more attention.

OpenEd2007 raised many issues that researchers and OER practitioners can take away and hopefully develop solutions to be discussed at future OER conferences.

Future Directions

OER have a significant and growing presence in the education systems of developed countries. Unfortunately, their impact in developing countries remains below its potential. There is, however, a growing desire from a variety of organizations to bridge the digital divide and bring both technology and education to the developing world.

Currently, it is unclear how different forces are impacting innovation in the OER domain. The motivation and ability framework described by Clayton Christensen in "Seeing What's Next: Using Theory to Predict Industry Change" is one possible perspective (http://tinyurl.com/2b5vf5).

In systematically applying this framework, we find that although organizations such as the United Nations (UN), various UN agencies, and other NGOs have mandates that provide them with the motivation to provide localized OER to developing countries, they lack the ability to effectively do so. On the other hand, Western businesses, universities, and governments of developed nations have the financial and technological capability to enable this evolution of education, but they lack the required motivation. We observe that various forces are currently at work to shift the motivation and ability of these organizations. The signals of change we assess are an environment conducive to innovation, the availability of low cost computing and OER supporting technology, and increasing NGO technical capabilities and localization initiatives.

Strategic choices by participants in this emerging industry will shape its future direction. Management theory predicts that progress from innovation requires both the motivation and ability to innovate. For those groups with the motivation to improve education in developing countries, the development of new technologies and OER will complement their existing educational infrastructure. For those groups with the required ability, partnerships will create a viable option for education.

OER complement existing educational technologies by enabling better learning without compromising other national priorities for education. We also identify and assess the impact of potential hurdles in the path to widespread innovation of OER in developing countries, and summarize the implications of our analysis for researchers and policy-makers (http://www.51weeks.com/events/3/presentations/38).

Andrew Pullin received a B.Sc. in Combined Honours Chemistry and Computer Science from Carleton University in 2006. He is currently a Master's student in the Technology and Innovation Management within the Faculty of Engineering at Carleton University. His research interests include open source project ecosystems, open source licensing and open educational resources. He currently acts as Associate Director for Shad Valley Carleton (http://carleton.ca/engineering/shad/) and serves on the Board of Advisors to Virtual Ventures (http://virtualventures.ca).

Kamal Hassin received a B.Eng. in electrical engineering from Carleton University in 2004. He is currently a Master's student in Carleton University's Technology Innovation Management program. His research interests include ensuring clean intellectual property in software projects, intellectual property law, open source licensing, and open educational resources.

Monica Mora received a B.Eng in Mechanical and Industrial Engineering from the University of Panama in 2004. Since then, she worked for the Technological University of Panama in different positions, including assistant professor and assistant of the President of this university. She is currently a graduate student in Technology and Innovation Management, Faculty of Engineering, Carleton University.

Kevin from Ottawa writes: Regarding the OSBR issue on open source licensing, I couldn't help but be amused when I came across this item in the November issue of New Scientist:

Intrigued, he decided to read the licensing conditions. These told him that he could use the program free of charge indefinitely only if he took "at most 4 flights (2 return flights)" in any 12 month period and if he did not "own or regularly drive an SUV (sports utility vehicle)". Haig assures us that happily he meets both these requirements, but they set him wondering if these are unusual licence conditions for free software. We'd like to hear if there are others like them.

That quote is from the "Strict conditions" section of http://www.newscientist.com/article.ns?id=mg19626282.300.

Editor: That reminds me of the Beerware license which predates the open source definition (http://en.wikipedia.org/wiki/Beerware). It doesn't predate the free software definition and seems to have prompted the need to explain the distinction between "free as in beer" and "free as in speech" (http://c2.com/cgi/wiki?FreeAsInBeer).

Glen from Ottawa writes: The October issue of the OSBR mentioned that one of the advantages of open source software is that it is not subject to orphaning like proprietary software. In 1999 at the Canada Institute for Scientific and Technical Information we needed to be able to authenticate a squid (http://www.squid-cache.org) proxy server with an LDAP. At that time, there was no tool that could do this. So I went home and wrote one, squid_ldap_auth (http://forge.novell.com/modules/xfmod/project/?sqauthldap), and posted it to the web. I had no interest in maintaining it, etc. I was contacted by Henrik Nordstrom who asked me some questions, and I helped him as I could, but I made it clear to him that I had no interest in maintaining the code, and gave him my blessing to update it as he saw fit. It has been extended and is being used extensively. This project was orphaned from the point-of-view of my not being interested in maintaining it, but this orphan was picked-up by the community.

PIKA Extends Appliance Offering to Linux Community

October 24, Ottawa, ON

PIKA Technologies Inc., a developer of media-processing hardware and software, today announced the release of its Appliance for Linux, the second member of PIKA Warp, its new Appliance family. It provides Linux computer telephony (CT) application developers with a smaller-sized and lower-cost alternative to traditional off-the-shelf computers and plug-in board network connectivity. The Appliance for Linux is equipped with the latest version of PIKA's field proven host media processing (HMP) software.

With the expansion of PIKA Warp, the company continues to fulfill its mission to provide solutions for the Linux open-source development space. The inaugural product, the Appliance for Asterisk released last month, is purpose built for the Asterisk open-source communications platform; the PIKA Appliance for Linux extends the benefits of the Appliance to the general Linux developer community. This Appliance allows IP/PBX, Integrated Voice Response, predictive dialing, appointment reminder and other Linux-based CT applications to be deployed on a cost-effective and reliable technology platform.

http://www.pikatechnologies.com/news/10-23-07%20-%20Appliance%20offering%20to%20Linux.htm

Aheeva Wins 2007 Digium Innovation Award

October 25, Montreal, QC

Digium has announced Aheeva as the winner of the 2007 Digium Innovation Award in the Big Business category for its Aheeva Contact Center Suite (CCS) that enables clients to establish a superior full-service contact center for a fraction of the investment required to operate a traditional telephony switch-based call center. The flexibility of Asterisk has allowed Aheeva to create a multimedia customer contact and relations management platform that can be deployed quickly and easily integrate with other open source peripherals to meet changing contact center needs. The company leveraged input from the global Asterisk community and, in the true spirit of open source, gave a portion of the resulting work back to the community.

http://www.tmcnet.com/usubmit/2007/10/25/3043356.htm

January 22

Workshop on Open Source Best Practices

Montreal, QC

The commercial use of open source is hindered by many factors. These include a lack of integration with traditional requirements-driven product development approaches, licensing issues, a clash with existing corporate culture, and the perception that in order to benefit from open source you need to open your source to the outside world. The goal of this workshop is to bring together researchers and practioners with experience in open source adoption and value creation from open source, and to document the best practices.

http://www.carleton.ca/tim/events/wosbp2008/

January 23-25

Montreal Conference on eTechnologies

Montreal, QC

MCETECH2008 will feature a special track on open-source software for e-businesss, which brings an additional twist to the usual technical, organizational, and regulatory aspects of e-business. We also welcome contributions that deal with the extent to which open-source e-business software helps bridge the digital divide that exists between developed and developing countries.

http://www.mcetech.org/

The goal of the Open Source Business Resource is to provide quality and insightful content regarding the issues relevant to the development and commercialization of open source assets. We believe the best way to achieve this goal is through the contributions and feedback from experts within the business and open source communities.

OSBR readers are looking for practical ideas they can apply within their own organizations. They also appreciate a thorough exploration of the issues and emerging trends surrounding the business of open source. If you are considering contributing an article, start by asking yourself:

1. Does my research or experience provide any new insights or perspectives?

2. Do I often find myself having to explain this topic when I meet people as they are unaware of its relevance?

3. Do I believe that I could have saved myself time, money, and frustration if someone had explained to me the issues surrounding this topic?

4. Am I constantly correcting misconceptions regarding this topic?

5. Am I considered to be an expert in this field? For example, do I present my research or experience at conferences?

If your answer is "yes" to any of these questions, your topic is probably of interest to OSBR readers.

When writing your article, keep the following points in mind:

1. Thoroughly examine the topic; don't leave the reader wishing for more.

2. Know your central theme and stick to it.

3. Demonstrate your depth of understanding for the topic, and that you have considered its benefits, possible outcomes, and applicability.

4. Write in third-person formal style.

These guidelines should assist in the process of translating your expertise into a focused article which adds to the knowledgable resources available through the OSBR.

Upcoming Editorial Themes

December 2007	Clean IP
January 2008	Interoperability
February 2008	Data
March 2008	Procurement
April 2008	Communications
May 2008	Enterprise Readiness

Formatting Guidelines:

All contributions are to be submitted in .txt or .rtf format and match the following length guidelines. Formatting should be limited to bolded and italicized text. Formatting is optional and may be edited to match the rest of the publication. Include your email address and daytime phone number should the editor need to contact you regarding your submission. Indicate if your submission has been previously published elsewhere.

Articles: Do not submit articles shorter than 1500 words or longer than 3000 words. If this is your first article, include a 50-75 word biography introducing yourself. Articles should begin with a thought-provoking quotation that matches the spirit of the article. Research the source of your quotation in order to provide proper attribution.

Interviews: Interviews tend to be between 1-2 pages long or 500-1000 words. Include a 50-75 word biography for both the interviewer and each of the interviewee(s).

Newsbytes: Newsbytes should be short and pithy--providing enough information to gain the reader's interest as well as a reference to additional information such as a press release or website. 100-300 words is usually sufficient.

Events: Events should include the date, location, a short description, and the URL for further information. Due to the monthly publication schedule, events should be sent at least 6-8 weeks in advance.

Questions and Feedback: These can range anywhere between a one sentence question up to a 500 word letter to the editor style of feedback. Include a sentence or two introducing yourself.

Copyright:

You retain copyright to your work and grant the Talent First Network permission to publish your submission under a Creative Commons license. The Talent First Network owns the copyright to the collection of works comprising each edition of the OSBR. All content on the OSBR and Talent First Network websites is under the Creative Commons attribution (http://creativecommons.org/licenses/by/3.0/) license which allows for commercial and non-commercial redistribution as well as modifications of the work as long as the copyright holder is attributed.