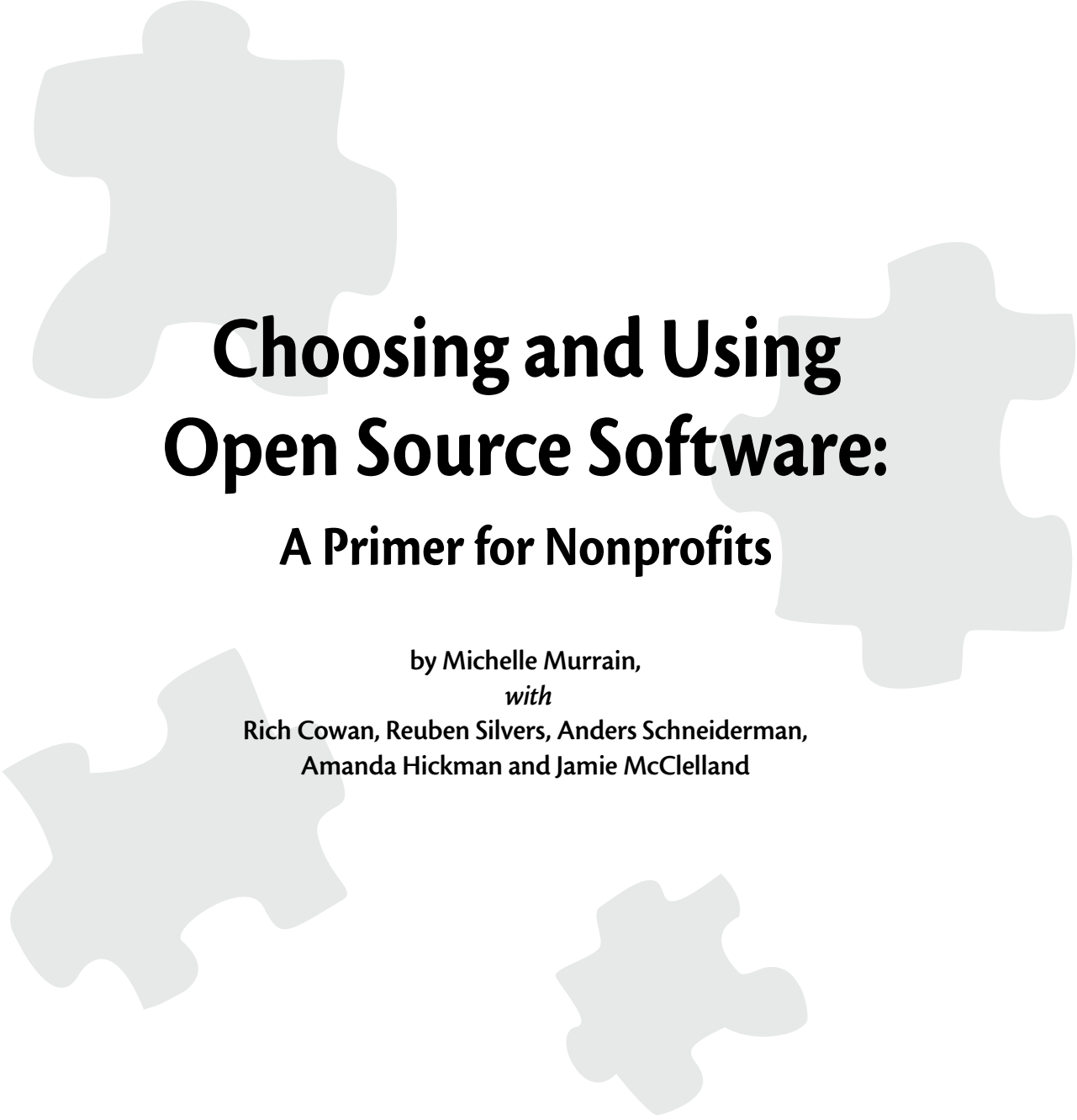




Choosing and Using Open Source Software:

A Primer for Nonprofits

by Michelle Murrain,
with

Rich Cowan, Reuben Silvers, Anders Schneiderman,
Amanda Hickman and Jamie McClelland

**Published by NOSI
(Nonprofit Open Source Initiative)**

Choosing and Using Open Source Software: A Primer for Nonprofits

by Michelle Murrain,
with Rich Cowan, Reuben Silvers,
Anders Schneiderman, Amanda Hickman
and Jamie McClelland

**Published by NOSI
(Nonprofit Open Source Initiative)
2004**

Reviewed by Katrin Verclas,
Carla Schroder and Andrew Gianni

Designed by Linda Roistacher

© 2004
Nonprofit Open Source Initiative

This document is released under
the Creative Commons Attribution
NonCommercial-ShareAlike 1.0 License

You are free to copy, distribute, display, and perform the work and to make derivative works under the following conditions: you must give the original author credit; you may not use this work for commercial purposes; and, if you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the author.

This primer was made possible
by a grant from
IBM International Foundation



*IBM is a registered trademark of
International Business Machines Corporation*

In-kind contributions of staff time were
provided by SEIU, Media Jumpstart
and Organizers' Collaborative

What Is NOSI?

NOSI (<http://nosi.net>) is a collaboration of nonprofit technology service providers, consultants specializing in nonprofit support, and open source software developers working to facilitate greater adoption of Open Source Software (OSS) in the nonprofit/NGO sector.

How can I become involved in NOSI?

NOSI currently has four opportunities for volunteer involvement or contributions:

- First is our online collection of open source case studies. We welcome contributions from additional nonprofits who have used OSS, especially if your case study tries things that are not mentioned in any of the existing case studies on our web site.
- Secondly, we are looking for people to help maintain our online open source projects database (nonprofit-focused).
- Third, we are looking for folks who can help maintain our web and email infrastructure.
- Finally, people who are already making a significant contribution to NOSI or the world of nonprofit open source software may be invited to become involved in the NOSI steering committee.

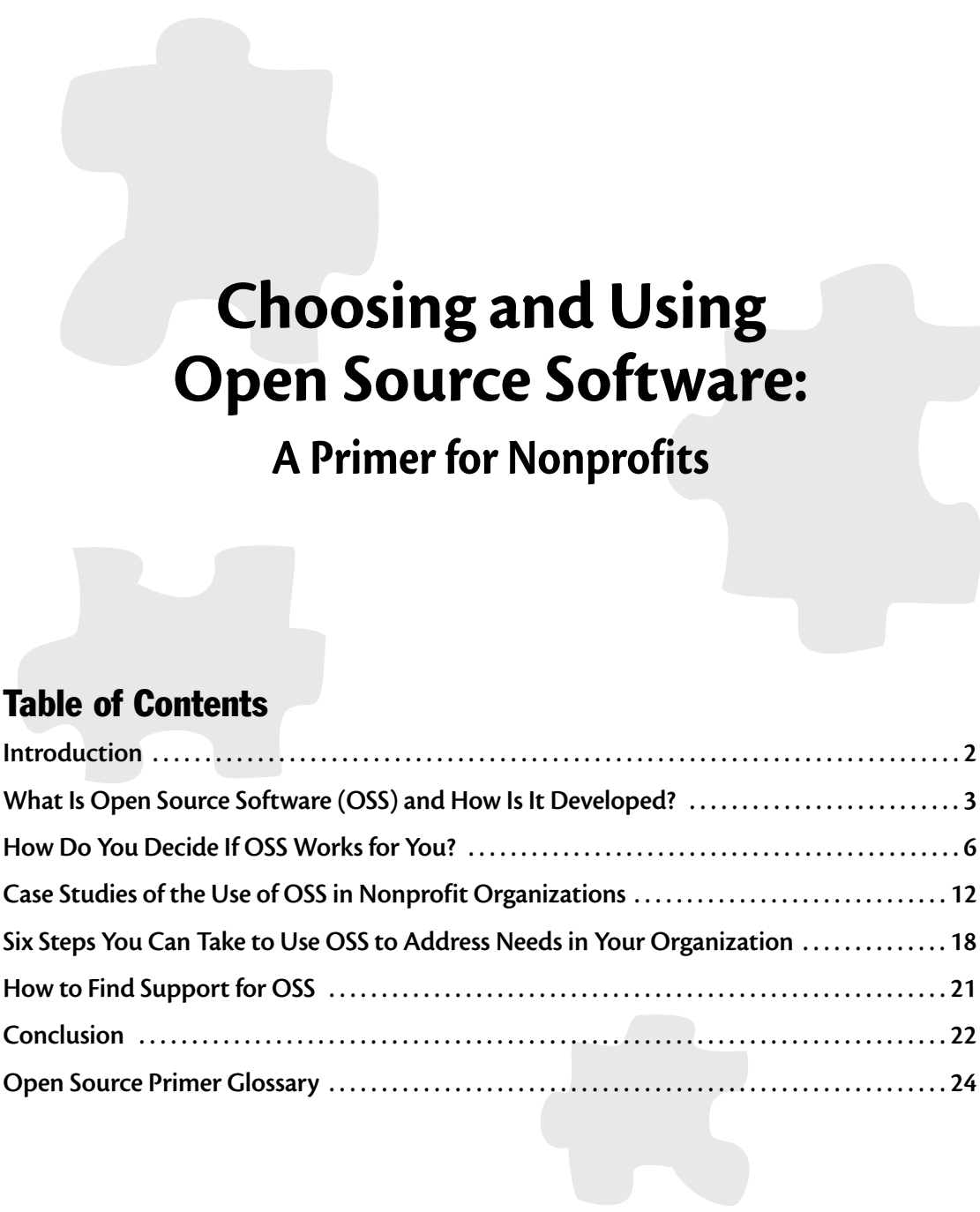
To volunteer, please contact us through our web site:
<http://www.nosi.net/contact.shtml>

NOSI currently runs three email lists:

- nosi-discussion@nosi.net**
general discussion, questions on nonprofit OSS issues
- nosi-announce@nosi.net**
low volume announcements of OSS issues
- nosi-desktop@nosi.net**
general discussion on OSS options for the desktop

To provide feedback on the primer, email **feedback@nosi.net**

To inquire about the availability of bulk copies, email
copies@nosi.net



Choosing and Using Open Source Software:

A Primer for Nonprofits

Table of Contents

Introduction	2
What Is Open Source Software (OSS) and How Is It Developed?	3
How Do You Decide If OSS Works for You?	6
Case Studies of the Use of OSS in Nonprofit Organizations	12
Six Steps You Can Take to Use OSS to Address Needs in Your Organization	18
How to Find Support for OSS	21
Conclusion	22
Open Source Primer Glossary	24

Introduction

Many nonprofit organizations that have previously relied almost entirely on closed commercial software in their organizations are now considering the "open source" alternative. Open Source refers to software distributed with the condition that anyone using it must have access to, and the ability to give away unlimited copies of, both the program and the source code that is needed to make changes to the program.

Commercial software, also called proprietary software, is software that is distributed under commercial license agreements, usually for a fee, and without access to the source code.

The open source model is proving to be a revolutionary development because it makes it easier for software designers to build on the work of others. As a result, the quality of open source software (OSS) has steadily improved so that in many cases open source software products rival or exceed the performance of their proprietary commercial counterparts. Although the true cost of software is more than its initial purchase price, the fact that OSS is freely available, and freely upgradeable, along with other characteristics (security, stability, access to source code for customization, and the absence of marketing gimmicks in the software) have led to its recent popularity.

The for-profit and academic sectors have long recognized that OSS can be more secure, stable and cost-effective. This explains why they have adopted OSS in larger and larger numbers, particularly for network and web servers. Today OSS is also gaining acceptance for use on individual desktop computers.

This development is significant for charitable organizations. While in the 1980s and 1990s most small-to-medium

sized nonprofits managed to get by with donated, "borrowed," and 5-year-old copies of software products, new approaches to licensing and copy protection are bringing an end to the era of "free" proprietary software. Microsoft, for example, is requiring that every product installed on every machine require a unique install key. Some companies do away with even this approach, replacing software as a product with a new model of software as a service. Under this model, software customers will have to pay annual fees, thus removing the cost-saving option of using older versions of software.

Given these trends, it is easy to see why nonprofits are finding the community model of software development appealing. Social justice nonprofits have always struggled to build diverse coalitions based on shared resources and a commitment to strengthen the entire movement, rather than enriching an individual or single organization. Community development groups strongly advocate on behalf of community ownership of resources and property. The concept of community has always been of real importance to nonprofits seeking to build genuine relationships with the individuals and groups with which they work. The proliferation of small, special-interest nonprofits attests to the belief that one size does not fit all – every community and issue is different and requires new and original solutions.

In this primer, we examine how OSS is developed, how its costs and benefits can be evaluated, how nonprofits are using it today, where to find technical support for OSS, and how you can take concrete steps to advance it in your organization. We also offer ideas on how to further advance the applicability of OSS to the nonprofit sector.

What Is Open Source Software and How Is It Developed?

The Promise of a New Model

In the past 10 years, computer and Internet technology have caused radical changes in the way organizations carry out their missions. The existence of more powerful hardware is certainly significant. But even more important is the software installed on these machines – the technology that allows those "boxes and wires" to communicate over the Internet, and accomplish mission-specific tasks.

In the past, the primary model for acquiring software was to obtain it at the time of your computer purchase, or as an add-on that you would purchase later. The software might come preinstalled or on a separate installation CD, but in no case would you have access to the source code. The software vendor would retain the exclusive right to make changes, and thus you would be at their mercy for updates. Quite often, enhancements and technical support for products you use would be discontinued after a time, possibly to encourage you to purchase newer products.

With the advent of the Internet, a new model of software development and distribution has gained acceptance that can potentially give you a lot more power. The key advantage of the new, open source model over the traditional, or "proprietary," model is that you actually have access to the source code of an application. This may not seem very significant to most nonprofits; after all, few of us can afford a software engineer on staff.

In practice, the fact that a software program is open source makes a huge difference, because it enables hundreds of technical people, many working as volunteers with no commercial incentive, to collaborate on continual enhancements to software. Eric Raymond, in one of the seminal papers on the OSS model entitled *The Cathedral and the Bazaar* (<http://catb.org/~esr/writings/cathedral-bazaar/>), wrote: "open-source software [is] the process of systematically harnessing open development and decentralized peer review to lower costs and improve software quality."

In addition, unlike many proprietary products, OSS is primarily based on open standards, which enhance the potential interoperability of software that organizations need to use, and also enhance the ability of different software to communicate with each other. Further, the use of open standards helps prevent "lock-in," a common problem where organizations are forced to continue using the same product because data migration would be too expensive.

How good can software be that is free?

Although there are many OSS packages that are not as high quality as their proprietary alternatives, this is actually not the rule. Many well-developed popular open source projects are better in quality than their proprietary alternatives. There have been a number of studies on this issue, all of which have concluded that most major open source projects are at least as good as their proprietary alternatives, and open source software has the capacity to be of higher quality than proprietary alternatives. See "Further Readings" below, and we'll be giving examples later.

Who Creates Open Source Software?

The movement toward open source software was originated by computer science researchers as early as the 1970s. Early open source projects were concentrated in academia, where developers popularized the idea of copyrighting one's computer programs with a statement mandating that the software would always be open to the public. This made a lot of sense, as much of the initial research in the field was funded with public tax dollars.

The contribution of Richard Stallman, the prime author of two open source software programs (one for compiling software code and the other for creating and editing documents) was particularly important. Tens of thousands of students worldwide used these products, learning about the values of what is commonly referred to as the "free software movement." The proceeds from the sale of user manuals for these programs were used by Stallman to develop the Free Software Foundation (FSF – <http://www.gnu.org/fsf/fsf.html>), which has evangelized since the late 1980s on behalf of **Free Software**.*

* Words that appear in boldface within the text are defined in the Glossary on page 24.

As a result of this movement the practice of "giving back to the community" by writing OSS was adopted by tens of thousands of software engineers. Many organizations followed suit, agreeing to release the source code of software developed for internal use. As time passed, OSS programmers were emboldened to take on almost any task. For example, in the early 1990s, a Finnish researcher named Linus Torvalds built upon the substantial work of FSF to create what is now known as **Linux**, or GNU/Linux, to replace the proprietary Unix **operating system**.

In the mid 1990s web sites like SourceForge.net arose to serve as clearinghouses for open source development efforts, so that open source programmers would be less likely to reinvent the wheel. The body of software that has resulted from this process is so powerful that most large corporations are now using OSS to carry out some of their IT needs, and some have chosen to replace proprietary software entirely.

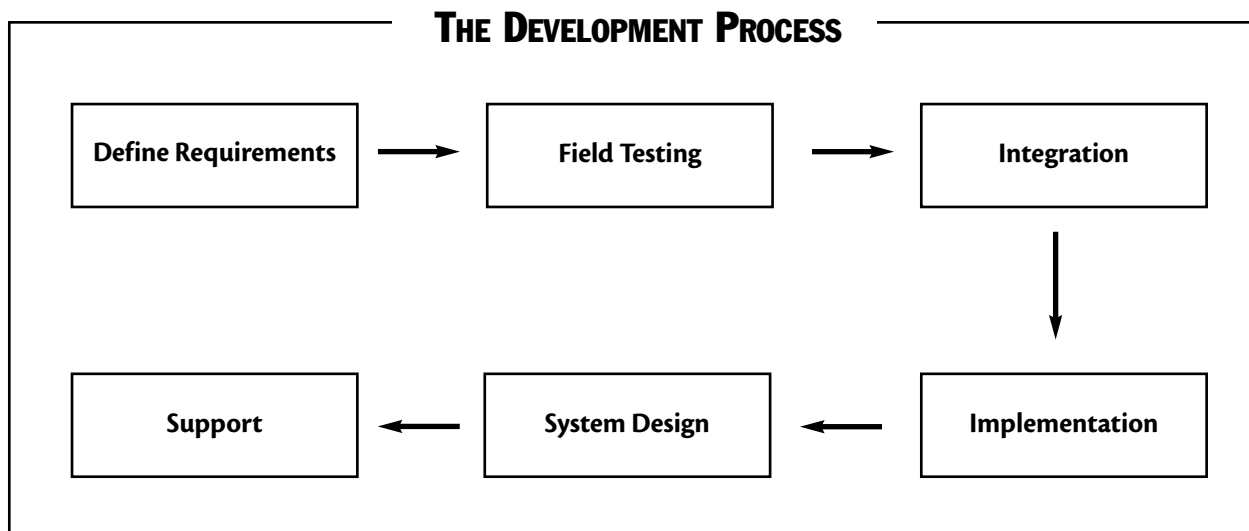
The ability to reuse code has also resulted in many **forks**, where one project is split into two or more independent efforts. Sometimes a "fork" will die out; other times it will take over the original in terms of popularity. This results in a proliferation of projects (both finished and unfinished) that to a novice can be overwhelming. We'll talk more later about choosing from the variety of OSS options, and how to identify the projects that have momentum.

The Development Process

Despite the many differences from proprietary software, it is important to note that OSS usually goes through the same stages as a proprietary product.

Some key differences are:

- In open source project development, this process may happen much more organically – starting with a single developer doing a relatively small project, then having the project involve more developers, and attract institutional support, as it develops.
- The pace of open source development can be slower, due to the voluntary nature of many development projects.
- The quality of open source software can be much better than proprietary software, because programmers learn from each other, the additional "sets of eyeballs" viewing the code tend to catch potential bugs, and there is less commercial deadline pressure to rush the software out the door in an unfinished state.
- The **version numbering** of open source software tends to be more conservative. Some very usable open source software is still in versions that are below 1.0. And even the most mature, robust projects (like Apache) may have only recently reached version 2.0.



Licenses

A license is basically an agreement between the user and the developer on how that software can be acquired and used. Whenever you install software, and click on the "I Agree" button, you are agreeing to a EULA (or End User License Agreement). The most popular open source license is called the GNU Public License (known as the GPL). The GPL stipulates that not only does the source code need to be available, but also the program can be modified and re-distributed, as long as that re-distributed program is also given the GPL. People have called this a "viral" license, because all modified code must also carry the GPL.

Unlike proprietary or commercial software, one of the hallmarks of OSS is that there are no unit or per-seat licenses – you can take one copy of the software, and install it on as many machines as you want, with no added licensing cost. You don't have to track licenses, worry about whether you are running 12 copies of a software package you have 10 licenses for, etc.

How Is the Development of OSS Supported?

While many OSS projects are completely staffed by volunteers, it is quite common for commercial companies, universities, and nonprofit groups to provide resources, usually in the form of developer time, to OSS projects that they can use internally. The range of people involved can be broad, from volunteers who report bugs and write documentation, to professional QA staff that might perform end-user testing. Almost all developer communities are open to providing user support (more on support in the "Administration and Support" section below).

The Apache project, which produces the most widely used program for powering web sites, formed a nonprofit foundation to raise funds for future development of their

There are many licenses that are less strict than the GPL. The best known of those is the BSD (Berkeley System Distribution) license. It allows you to modify and distribute the code as you see fit, but it need not carry the same license. There are many more Open Source licenses (see: <http://www.opensource.org/licenses/>).

project in 1999. Since then, a number of projects have followed suit, including Mozilla, a program used for browsing web sites (like Internet Explorer).

Open source projects have also spawned for-profit companies, whose primary goal is to develop and support the open source software project. Examples of these include MySQL (a popular database program), Zope (a tool for building interactive web sites), as well as many companies that package and support the Linux operating system, such as RedHat or SuSe.

Further Reading:

<http://www.nature.com/nsu/030623/030623-6.html> - How OSS can be of better quality than proprietary software

<http://www.reasoning.com/library.html> – *Reasoning Magazine's* examination of OSS

http://www.internetnews.com/devnews/article.php/10792_2230481 - News story about comparisons in quality of OSS and proprietary software

How Do You Decide If OSS Works for You?

Background

The adoption of open source software can be as simple as downloading one software package, installing it, and using it on a single computer, or implementing an entire Linux server cluster to do complex tasks, or anything in between. This section will focus on smaller-scale implementations of OSS, which are more relevant to most nonprofit organizations.

All organizations should consider implementing OSS. But in any organization, it is necessary to make a case for what can be a significant internal change.

This section lays out some of the factors you might need to address in weighing the costs and benefits of OSS versus proprietary solutions. (Later sections of the primer offer more concrete examples.)

Concepts

There are two concepts to consider:

Total Cost of Ownership: TCO is a familiar term to many people – it represents a calculation of how much technology costs to implement and maintain over time.

Strategic Value: Strategic Value takes additional factors into account beyond the costs related to the technology itself. In the nonprofit context this means accounting for the mission-based value that a technology might bring to an organization; for example the impact on staff productivity, or on the quality of services delivered to clients.

OSS can influence both the TCO of technology, as well as the strategic value that it brings to an organization.

TCO Considerations

TCO is a calculation of the *entire* cost of implementing a technology solution. This includes the initial cost of acquiring the software (purchase price or license fees), hardware costs, installation costs (staff time or consultant costs), end-user training costs, and the cost of maintaining the software (annual maintenance fees, support costs, and upgrade costs). This entire spectrum of costs should be considered when comparing *any* two solutions, whether one, both or neither is OSS.

The most obvious places where OSS has an advantage in terms of costs are in the software acquisition costs, maintenance costs, and upgrade costs. OSS is almost always freely available, usually has no license fees or annual maintenance fees (there are some exceptions, although they are not mandatory fees – they are generally for support) and upgrades are also generally free. However, the acquisition costs of many kinds of software packages are far outweighed by the other kinds of costs (consultant time, staff training, administration, etc.), so software that is free is not necessarily less expensive in TCO terms than software that you have to pay for.

■ Compatibility with Mission-Critical Applications

Key questions to ask when evaluating OSS technology:

What are the software applications critical to your organization?

What operating systems will they run on?

Another important factor to take into account is compatibility between the open source solution you are considering and any mission-critical applications. In particular, when considering using Linux as an operating system, be aware that many software vendors, especially smaller ones, still do not support versions of their software that can run on Linux. This is especially true of "niche" software products for the nonprofit sector, such as case tracking programs for legal-services organizations, or mortgage-tracking programs used by affordable-housing groups. Organizations that depend heavily on one of these programs find that, if they want to use Linux, they must also maintain a Windows machine dedicated to running that mission-critical program. If that is likely to be the situation, the costs of maintaining that additional machine need to be included in the TCO. In addition, any inconvenience to users could be considered a drain on strategic value.

Doing a TCO analysis on an OSS vs. proprietary solution is dependent on the complexity of the software, the end user impact, the administration costs, as well as the acquisition cost of the proprietary solution.

■ Software Acquisition Cost

Key questions to ask about software acquisition costs:

With a proprietary solution, how large will software acquisition costs be relative to other costs?

With a proprietary solution, how easy will it be to get discounted software?

Some proprietary products, like simple tools or small applications, have very low software acquisition costs. Other products, like complex databases, financial or fundraising packages, or server operating systems have high acquisition costs. In some cases, nonprofit organizations can get many software packages donated or at reduced costs, which may reduce or eliminate the acquisition cost for the software.

Sometimes, however, organizations are limited in how many copies of a product will be reduced-price or donated (for example, an organization can only get 50 user licenses of Microsoft Office XP from DiscounTech, so a medium to large organization would not be able to take full advantage of this donation.) However, virtually all open source software is available for no acquisition cost whatsoever, and no multiple licenses are needed.

Another factor to keep in mind is that with proprietary software, you may have to pay for separate software tools that are not needed with the OSS solution. For example, separate software is available to perform automated backups of Microsoft Windows servers, whereas with the Linux operating system, automated backup tools are included.

■ Implementation Costs

Key questions to ask about implementation costs:

What is the ease of implementation in terms of resources needed (time and money)?

What kind of expertise might be needed for this software, whether it is proprietary or open source?

What kind of expertise do you have available?

For some software, implementation is quite easy, and will take a staff person 10-30 minutes to install. More complex applications (financial, client management packages, etc.) take days of staff and/or consultant time to implement, and convert information from an old system.

When evaluating the options for a particular solution, be aware that in some cases open source projects can be more

difficult to install than their proprietary counterparts, especially if people are new to OSS. It is worth looking carefully at the installation documentation for any solution you are planning to implement.

Something else to consider is that if your organization needs consultant support, it is not always as easy to find consultants who are familiar with open source technologies, although that is slowly changing. You may need to consider seeking out a different consultant in order to move to open source technologies, if you now depend on a consultant who is unfamiliar with them.

■ Hardware Costs

Key questions to ask about hardware:

Is this a situation where you may need multiple servers?

Does the proprietary software being evaluated have any special hardware requirements? Does it require vendor-certified hardware?

In many situations, you will be implementing software on existing hardware, so there will be no additional hardware costs. However, if you are implementing a network for the first time, server hardware costs will likely be an issue. In general, the larger your network needs (in terms of capacity) the more hardware savings you are likely to realize with the Linux operating system and other OSS. Research has shown that servers based on Linux (as opposed to Microsoft Windows) can handle more traffic, hold more accounts and do more processing given the same hardware (see: www.apache.com/pdf/linux-windows.pdf). Thus in a situation where you might need multiple Windows servers, you'll be able to buy fewer Linux servers.

Another way that Linux lowers hardware costs is by allowing the use of generic, industry standard servers. Proprietary software companies sometimes require that their programs be run on specially "certified" – and therefore more expensive – servers. For example, an organization that uses Novell for their network will often have to buy Novell-certified hardware in order to get Novell support. Certified hardware tends to be significantly more expensive. Linux, by comparison, is designed to run on cheap, industry-standard hardware.

In general, the more high-end system, the more using Linux is likely to save money on hardware.

■ Training Costs

Key questions to ask about training costs:

Will implementing this software (whether OSS or not) require end-user training?

Will you rely on in-house staff to support this solution, and if so, what is their current knowledge of the solution?

For end-user solutions (like office applications, financial packages, etc.), training is by far the most costly part of implementing new technology. Staff who are going to use this software on a day-to-day basis must be trained to use it. Most non-technology staff will be unfamiliar with non-Windows operating systems and applications, so implementing an open-source solution that replaces a well known and used application on a wide-spread basis must be considered carefully. The benefits of using a solution like OpenOffice.org, for example, may or may not outweigh the wide variety of training costs that will be incurred. But the long-term training implications (once the staff are trained, you only require standard on-going training, and training of new employees) should also be factored in.

For software that has little or no end-user impact (file servers, database servers, etc.), the only training costs to think about are training your technology staff (or "accidental techies") to handle the new systems. These costs become a larger factor when a) you rely on in-house staff, rather than outside consultants, to support the solution, and b) your in-house staff does not have experience using OSS.

■ Maintenance Fees

Key question to ask about maintenance fees:

Does the proprietary alternative require annual maintenance fees?

Some software products have annual maintenance fees. For all practical purposes they can be thought of as annual license fees, since they are typically calculated as a percentage of the original software license acquisition fee. These fees should be included in your TCO analysis.

Most OSS has no annual maintenance fees, since it had no license acquisition cost to begin with. Some enterprise-level Linux distributions (like RedHat) do have annual maintenance fees, usually for support. However, very few nonprofits fit into the categories of organizations that would make use of these enterprise-level packages, and they are not mandatory.

■ Upgrade Fees

Key questions to ask about upgrade fees:

How often might you need to upgrade this software?

Are upgrades available at a nonprofit discount, and if so does your organization qualify?

Keeping software relatively up to date is important. It increases stability, security and features. It is not necessary to upgrade at every version change, but it is critical to install security patches, and when features that you want are improved, or there are dramatic stability improvements (for example, the upgrade from Windows 98 to Windows 2000), upgrading is important.

Obviously, the cost of upgrading a single copy of a software product is much less expensive than having to do a site-wide upgrade. You can often get upgrades of proprietary software at discounted fees or donated. The vast majority of open source software has no upgrade fees at all. You can simply download the upgraded version, and install.

In some situations, it might be easier to go to a local computer store and get a new boxed version of a Linux distribution, than spend the time downloading the CD image, and burning it to CD. But it is always possible to get CD images of some Linux distributions at no cost.

■ Administration & Support

Key questions to ask about administration and support:

What sources of support are available for the open source option? How would they fit into your overall IT support strategy?

How important a factor is reliability in your choice of solution?

Are viruses and other security problems common with the proprietary option?

All software – whether applications, databases, or operating systems – requires administration and support of some type. In some cases, you will have your staff provide the support; in other cases, you will hire consultants to do it. Characteristics of the software that determine the level of support needed include how reliable the software is, how vulnerable it is to security problems, and how complex it is to use both for end-users and the administrator. Factoring these into your consideration of OSS is important.

Using Linux as the primary example, there have been numerous studies and surveys that show fairly conclusively that Linux is more stable and more secure than Windows. In a survey of companies that implement Linux solutions, quoted in *LinuxWorld* (<http://www.linuxworld.com/story/33838.htm>):

The poll was conducted among 2,500 corporate customers around the world, and no fewer than 95% of them rated Linux's reliability as its most important contribution to business value, followed by cost of acquisition (89%).

Nonprofits interviewed for this primer that use Linux placed a similar importance on its reliability, as well as security (see Case Studies section). Without exception, every one said that after they moved to Linux, their network was more stable and stayed up longer. Participants also agreed that it is easier to secure their network using Linux. In part this is because Linux and related operating systems were used for the Internet several years before Windows adopted Internet technology, and the open nature of the code enabled developers to snuff out many potential security problems even before the Internet became popular. In part because of these design advantages, computer viruses have not affected Linux, while they are prevalent on Windows.

While the complete value of increased reliability and security is difficult to quantify, IT labor costs are an area where several nonprofits we talked to experienced clear cost savings. For example, the Springfield Neighborhood Housing Services' consultant said that organizations using Windows for networking usually spend about twice the amount of time on server maintenance than those using Linux, primarily due to more virus problems and system crashes with Windows.

For an organization that relies on consultants for support, this time saved translates directly into costs saved. For an organization that uses in-house support staff, the savings may be more complex to calculate. However, if you are in a situation where you have multiple servers, one administrator can handle more Linux servers than Windows servers.

It is also reasonable to assume that the greater reliability and security of Linux systems improves the productivity of end users. Staff can get more work done if their systems are out of commission less often. Fewer network outages are also likely to improve morale. As most of us have learned the hard way, having your computer crash and destroy the document you were writing or being unable to use e-mail for several crucial hours during the day is incredibly frustrating.

While it is often the case, one cannot generalize Linux's reliability to all OSS. There are many OSS projects that are neither more stable nor more secure than their proprietary alternatives, so doing research will be important for comparison.

In addition to the reliability and security of a solution, you also need to account for its complexity. Complexity can increase support costs in one of two ways: either by increasing the time needed to perform certain tasks, or by requiring a more highly skilled (and thus more expensive) person to do the job. With regard to the first point, our experience and that of the organizations we interviewed is that OSS is not necessarily more difficult (or easier) to administer than proprietary software. An important caveat, however, is that this assumes that the administrator is familiar with the OS solution. If not, additional training costs will be incurred (see "Training" section above).

With regard to the second point – whether OSS support staff and consultants are more expensive – our experience is that this is less of a concern than many nonprofits expect. For example, Linux administrators tend to be only marginally more expensive (0-10%) than Windows administrators. The larger issue faced by the organizations we interviewed tended to be finding staff and consultants who work with nonprofits and have OSS skills. In the case of networks, there are many more Windows administrators than Linux administrators available in most areas. Again, this does not necessarily hold for all OSS. And, in addition, the open source community affords its users many more informal sources of support than are available with proprietary software (see "Support" section).

Summary Worksheet

Below is a worksheet you can use to help you calculate the TCO of a proprietary vs. open source system.

	PROPRIETARY SOLUTION	OPEN SOURCE SOLUTION
Software Costs		
Hardware Costs		
Installation Costs		
Training		
Upgrade Costs		
Maintenance Costs		
Support Staff		

EXAMPLE: File/Mail/Web Server providing mail for 25 users with a 4 year investment period

	PROPRIETARY SOLUTION (Windows 2003 server with Exchange)	OPEN SOURCE SOLUTION (Knoppix, Debian, Suse or Fedora Linux) ¹
Software Costs	Windows 2003 Server: \$30. 25 client access licenses: 25. Symantec AntiVirus: 20. (These rates available only to nonprofit groups purchasing via Discount Tech)	0.
Hardware Costs	1,304.	1,304.
Installation Costs (Consultant at \$75/hr.)	4 hours = 300	4 hours = 300.
Training ²	0.	0.
Total Installation Cost	\$1,679.	\$1,604.
Upgrade Costs ³	\$55.	\$0.
Maintenance Costs ^{4,5} (Consultant at \$75/hr.)	\$3,600./yr x 4 years = \$14,400. (average of 4 hours/month)	\$1,800/yr. x 4 years = \$7,200. (average of 2 hours/month)
Total (over 4 years)	\$16,134.	\$8,804.

¹ There also exist commercial distributions such as RedHat Enterprise or SuSE that range in cost from \$80 upwards depending on levels of support and inclusion of proprietary high-end features. However, for the purposes outlined here, any distribution listed above will have the necessary features.

² Since this is server software, there will be no training implication for end users.

³ Based on the assumption that upgrading the OS in 2 years or so will cost the same as the acquisition cost – which is not always the case.

⁴ This estimate is based on the assumption that regular, scheduled maintenance will take place.

⁵ The assumption that Linux maintenance will take fewer hours than Windows maintenance is based on the information in one of the case studies, and is borne out by the experience of four of the authors of this primer who have experience with both operating systems.

Strategic Value

Besides TCO (Total Cost of Ownership), you need to consider what we will refer to as Strategic Value. This type of value is harder to quantify, but can often be more important in the decision process.

One facet of strategic value for open source solutions is the ability to solve problems in ways that would not be possible with proprietary solutions. This includes customizing software to exactly fit the needs of organizations. It also includes having software that is secure enough to enable a wider variety of remote communication and data sharing. For example, an environmental organization put small, inexpensive Linux servers in remote locations that can be easily remotely accessed for administration purposes, but are still very secure. Because of the lower cost, and flexibility of OSS, it is possible to solve problems that may not have had affordable, ready solutions in the past.

Control (or lack thereof) is another strategic consideration that leads some nonprofits to opt for OSS. They may have had a bad experience relying on a proprietary software package designed to serve their particular "niche" of the nonprofit sector. If the vendor of that software goes out of business, or decides to stop supporting that product, then its customers will likely have nowhere else to turn for support. With OSS, if the original developer disappears the product can live on, supported by its community of users and other developers. So, in the long run, the open source approach can provide a degree of risk mitigation.

Lastly, we would point out another strategic factor unique to nonprofits: philosophical. The philosophical underpinnings of OSS (community-based development, volunteer effort, freely available software, community support) are very much in line with the mission of organizations in the nonprofit sector. Some organizations have focused on this more than others, specifically if their mission is more related to technology (like alternative media organizations, for example). In general, we would argue that all other things being equal, making a choice to implement OSS in a nonprofit organization would add more mission-based value.

Further reading:

<http://www.newsfactor.com/perl/story/22012.html>

http://www.infoworld.com/infoworld/article/03/08/29/34FElinux_1.html

<http://eweek.com/arcle2/0,3959,1234349,00.asp>

<http://www.ibm.com/linux/RFG-LinuxTCO-vFINAL-Jul2002.pdf>

<http://www.itweek.co.uk/News/1131114>

Case Studies of the Use of Open Source Software in Nonprofit Organizations

In order to really understand how OSS can be useful in nonprofit environments, we have included here five case studies of actual nonprofit organizations that have implemented Linux on the server or on the desktop.

LINUX ON THE SERVER

■ CASE STUDY #1:

Springfield Neighborhood Housing Services

Springfield Neighborhood Housing Services (SNHS) is a seven-employee nonprofit in Springfield, Massachusetts whose mission is to help low income families purchase and rehabilitate houses. In the late 1990s SNHS decided to move from stand-alone computers to an office network. A key goal was to network several software packages they used to do mortgage tracking and related tasks. These programs were on stand-alone PCs and they wanted to make them available to everyone in the office.

SNHS went through a strategic technology planning process in 1999-2000 to plan the transition to a network. Based on the SNHS's needs and transition plan, their consultant, Paul Wright, recommended looking at Linux for their network server operating system.

In comparing Linux to Windows NT, Wright noted that with Linux, there would be initial cost savings from not having to buy software licenses for the server, including user license costs. For SNHS, this would only add up to a few hundred dollars in savings. However, the long-term benefits from the stability and security provided by Linux seemed more important. Wright thought these would help reduce ongoing support costs for SNHS.

In addition, Wright and SNHS liked the philosophy behind Linux and open source. "There's a lot of focus on cost savings," says Wright, "but it really comes down to an issue of control."

On the downside, the mortgage tracking programs would require Microsoft Access to work well over a Linux network. Microsoft does not support running their applications on Linux. For basic applications like Word and Excel, storing data files on a Linux server is not a problem. But running a Windows-based database from a server is a more complicated proposition, and potentially more problematic.

The Decision

Despite this potential complication, SNHS decided to use Linux for their network server. Wright chose a free distribution of Linux called Debian. The server would use Samba, a program that comes with Linux, to share data between the PCs running Windows on users' desktops and the Linux server. For e-mail they used a program called Sendmail as the mail server, and Eudora for the e-mail clients on users' PCs.

The implementation took two to three months overall to migrate users, but only a day to set up the Linux server. As for the loan tracking programs, one of them did end up causing problems. It kept freezing when users tried to access the database. Wright contacted the small software company that sold the program, but when the technical support person heard they were running it on Linux, they immediately blamed the problem on that. Eventually Wright went back to running that program on a separate Windows computer, not connected to the network.

Business Impact

According to Wright, the biggest impact from using Linux has been in terms of support costs. "Once it's installed and running, Linux is less work," he says.

He spends about five hours per month providing support to SNHS, but only an hour or so of that is related to the network. The rest is spent solving problems with the Windows desktop computers. What little maintenance the server requires includes updating security patches, and adding and removing users.

At organizations Wright supports that use Windows for networking, he estimates spending about double the amount of time on server maintenance (3-5 hours per month as opposed to 1-2 hours with Linux). Typically this is due to more frequent virus problems and system crashes with Windows, according to Wright. Also, more of the Linux system problems can be handled remotely.

Other than spending less consulting time on server problems, SNHS's support situation has not changed much. After the Linux implementation, SNHS staff received minimal one-on-one training so they could run backups and re-boot the server if necessary. For anything else, they call the consultant, as was also the case with Windows.

Learning basic Linux tasks was initially a challenge for the

"accidental techie" at SNHS. In general, Wright finds that the difference in ease of administering Linux versus Windows changes as the administrator moves up the learning curve.

Future Plans

As was noted, most computer problems at SNHS that require consultant help are now Windows desktop problems, including problems caused by computer viruses. Wright has therefore suggested moving to Linux on the desktop as well as the server. He set up a Linux machine for SNHS staff to experiment with. SNHS management has been very receptive to the idea and is seriously considering the move to the desktop.

However, the problem remains that some of the programs users rely on run only on Windows. To address this, they are trying to run these programs using Win4Lin, a program that allows you to run Windows programs on a Linux machine. The idea is to use OpenOffice on Linux for most things, and only open a Win4Lin window when necessary. However, since users are more familiar with Windows, once they get into the Win4Lin window they tend to stay there and use MS Office. The Executive Director is "slowly breaking that habit, but until he does we're reserving judgment on moving to Linux desktop."

■ CASE STUDY #2: Community Resource Exchange

Community Resource Exchange (CRE) is a nonprofit management and fundraising consulting group, with 45 staff located in their New York City office. There is only one office, but many staff work from home.

Media Jumpstart, a technology support provider for nonprofits, shares the same office with them. CRE receives IT consulting and in exchange provides space for Media Jumpstart's five staff people. This creates a somewhat unique situation in that Media Jumpstart are their tech consultants, but always available on site.

Business Problem

CRE began to use the Linux operating system to host some of their network functions beginning in 1999. Within several years they had a mixed environment where they used Linux and other open-source software for their e-mail and web server needs, and Microsoft Windows NT for their network file server, to host their accounting system, and on end-users' desktop computers.

The mixed environment was not itself a problem, but computing tasks were poorly distributed across the various network servers. Jamie McClelland, a Media Jumpstart con-

sultant, wanted to re-configure the network to make it more cost effective and to make use of features of Linux that would help CRE's staff who work remotely.

In doing so he had to consider two factors. First was the organization's accounting system, a program called MIP, which only runs on Windows. Second, the organization relied on a FileMaker Pro database. Their FileMaker database had to stay on a Windows box. FileMaker had a demo version for RedHat's version of Linux, but Jamie could not get it to work. And since MIP only works on Windows anyway, he felt it wasn't a big deal to also keep FileMaker on Windows.

The second consideration was that many users working from home needed a cheap, easy way to access their files across the Web. They also wanted the ability to easily get backups of their data remotely. If they lost a file they did not want to have to wait for help from a network administrator in order to get it back.

Jamie didn't feel comfortable opening up their Windows NT server over the Internet because of security issues. But the Linux e-mail/web server was already on the Internet. He recalls, "Our attitude was that the NT file server was behind the **firewall** and thank God it was!"

The Decision

Jamie and the CRE management decided on a major overhaul of CRE's network. They moved the file server to Linux. Security, flexibility and long-term reliability were the main reasons. In addition, they made another Linux server into a dedicated backup server.

With this new setup, Jamie used Linux and Apache to set up a secure extranet for CRE. Remote users log in and get directed to their personal home page that includes links to webmail, a link to browse their files on the file server (a feature of Apache), and links to an online FAQ. Users are even able to restore their backed-up files over the web.

CRE kept one Windows server to use for hosting FileMaker and MIP. It also served as the authentication server to authenticate users logging in at their Windows desktop PCs. For the most part the transition went smoothly. However, they encountered several problems while setting up Samba, the open source program that comes with Linux and is used for sharing data between Linux and Windows machines. "The transition was surprisingly smooth," said Jamie. "It was a typical experience – there was only one tiny piece that was awful. People think that if they install Linux they're going to spend hours and hours figuring out how it works. But we forget that we do that with proprietary software also."

Business Impact

The move to Linux impacted CRE's network reliability and maintenance more than any other aspect of their IT setup.

Software

Software license costs were not a big issue because as a nonprofit CRE could get discounted versions of Windows. However, Jamie points out that there is the long-term issue of depending on Windows discounts to continue. In addition to Windows itself, CRE would have had to buy licenses for software to perform file server backups, which Jamie estimates might have cost \$500. With Windows they also might have had to purchase some sort of VPN software in order to allow secure remote access.

Reliability

Using Linux at CRE has decreased the number of network problems in several areas. First, the file server itself does not need to be re-started as often. With Windows NT, every month or so the entire system would slow down, requiring a Media Jumpstart consultant to reboot it.

Second, backups are more reliable. With NT, they used a proprietary program called Retrospect for backups and had experienced occasional problems. When they had to reboot NT, the timed backup would stop working and things wouldn't get backed up. Jamie spent a lot of time trying to figure out these problems. With Linux he wrote a simple script that has been much more reliable.

Third, remote administration is another benefit of Linux being more secure than Windows. "It makes a world of difference in terms of troubleshooting. I'd be very nervous doing something similar with Microsoft."

The only major incident they have had maintaining Linux came from RedHat's update program, called Up To Date. When Up To Date updated their e-mail server, it overwrote some configuration they had done to the e-mail program. This took several hours to fix.

"Up To Date seems better for desktops than for servers," says Jamie, "since servers have more custom configurations."

Administrative Skill

Jamie observed that learning simple network administration is easier on Windows than on Linux: "If you took someone who knew nothing [about network administration], learning Windows would be quicker. ..."

At the same time, Jamie says learning advanced troubleshooting is the same amount of work for either Windows or Linux. For this reason the key question for a small nonprofit like CRE, that does not have dedicated in-house IT

staff, is whether their outside consultant has knowledge and experience with Linux.

Lessons Learned

Reflecting on his experience with CRE and other clients' networks, Jamie offers several pieces of advice for an organization considering Linux:

"Plan ahead. Plan what each server's going to do. Think through every step in the migration process beforehand."

"Don't get caught up in the fact that you'll save money on licenses. Do get caught up in the long-term issues: if we go with Windows now, what will that mean two or four years from now? If we go with Linux, what will that mean two-to-four years from now?"

"Get a concrete idea of what you want. For us it was file sharing over the Internet. Then figure out how you would do that using each platform."

■ CASE STUDY #3:

The Edna McConnell Clark Foundation

The Edna McConnell Clark Foundation (EMCF) is a private foundation located in New York City. With a staff of about 25 employees it makes approximately \$25 million in grants per year, with a focus on funding youth development programs.

Business Problem

Up until the late 1990s, EMCF had an office network with about 30 computers connected to multiple Novell servers, including one to run Groupwise, Novell's e-mail and collaboration program. But over time IT Director Ricardo La Matta and other members of the foundation's management became increasingly dissatisfied with the costliness of the Novell solution. An upgrade to a Novell server could cost \$15,000 for hardware and software alone. Consultants with Novell expertise were an even larger expense. At one point La Matta calculated that they were paying \$30,000 per year for outside support for Novell.

Besides cost, La Matta was also looking for an environment where he could experiment with bringing the EMCF web server in-house. This would also require firewall protection.

The Decision

In 1998, EMCF replaced their Novell network file server with a Linux server. At the same time La Matta added five additional Linux servers. One was an in-house web server to host the foundation's website, as well as receive e-mail. Another server acts as a firewall. The remaining three servers

are backups. If one of the three Linux production servers were to break down (which has never happened – in fact they've been up for three and a half years) the backup would take over. The broken box could then be replaced with a new one.

E-mail was an issue because EMCF staff were used to using Groupwise and liked its features. But Groupwise did not have a version that runs on a Linux server. La Matta therefore decided to keep a server running the Novell operating system and Groupwise application. E-mail goes from the firewall to the web/e-mail server, and then is passed along to Groupwise on the Novell server. This way an e-mail archive exists on the web/e-mail server as a backup.

La Matta chose to use all open-source software for running the EMCF website – Apache as the web application server, PHP as the web scripting language, and MySQL for database requirements.

Business Impact

Since moving their network to Linux, EMCF has saved money on the ongoing cost of maintaining their network. The most obvious cost savings have been in software and hardware costs. But additional benefits include better reliability and fewer network problems that previously distracted IT staff from new EMCF mission-related projects.

Software

With Novell, software upgrades to the servers could add up to \$8,000 per upgrade, which typically broke down as follows:

- Novell server operating system upgrade: \$4,000
- Groupwise application upgrade (Groupwise server only): \$3,000
- Upgrades to ARCserveIT for backups and InnoCulateIT for virus check: \$1,000

For the servers now running Linux – the file server, web server and firewalls – there are no longer any software license upgrade costs.

Hardware

Besides software, another way that Linux lowers costs is by allowing the use of generic, industry standard hardware. Proprietary software companies like Novell sometimes require that their programs be run on specially "certified" -- and therefore more expensive -- servers. For example, La Matta notes that the last Dell power unit they bought cost \$5,000-\$6,000 (although this included extras like RAID control cards and SCSI cards).

When EMCF switched from Novell to Linux, they no longer had to buy hardware that was Novell certified in order to get Novell support. The new Linux servers only cost about \$1,000 each because Ricardo and his staff put them together, not having to worry about them being "certified."

On an ongoing basis, with Linux EMCF has not needed hardware upgrades the way they do with Novell. They are still using the original machines, which have 800MHz processors. As has been the experience in the corporate world, EMCF found that Linux and other Open Source network software does not tend to require machines that are as fast or have as much memory as proprietary network software.

Consultants

Because La Matta is familiar with Linux and the other open source software that EMCF uses, he can do any necessary upgrades himself. This eliminates the substantial costs for outside consultants, for the servers now running Linux.

Reliability

La Matta offers the following advice to someone considering Linux for their network servers: "Go Linux and you will forget what it means for a server to go down. You will be able to take uninterrupted vacations, and during working time you will be able to dedicate your efforts and energy on projects you could not do so otherwise."

By comparison, the Novell server they have retained to run Groupwise tends to have more problems: "Our Novell server continues to go down every once in a while, but it affects only our GroupWise use. Unfortunately even the Novell engineers were not able to help us fix the problem, so they suggested another upgrade. We will do it because GroupWise is what our staff is used to, and it is indeed a very useful program. I just wish it would run on Linux."

■ CASE STUDY #4: Greenpeace, USA

Greenpeace is an international organization with a closely-knit network of national and regional offices located in over 25 countries spanning the globe. The organization's US headquarters is located in Washington DC, with 100 staff and interns.

The organization's international headquarters is located in Amsterdam, the Netherlands. Greenpeace International has adopted a policy of encouraging its affiliate organizations to use OSS. Any custom-built software Greenpeace produces will be released under the open-source GPL software license.

Business Problem #1

The DC office also serves Greenpeace staff throughout the Americas, so they needed an affordable, easy-to-use system for remote access. For example, staff in Guatemala needed to be able to get their e-mail and access files stored at the DC office. A virtual private network (VPN) of some sort was the most likely solution.

Because Greenpeace is always adding new small offices, many in developing countries, they also needed the ability to add new sites to the VPN quickly and easily. In many cases this would have to be done by non-technical staff at the new location.

Approach to Problem #1

Greenpeace used Linux and cheap, off-the-shelf hardware components to create an innovative system for setting up remote offices. They use custom-designed Linux "green boxes," about the size of a pizza box, to connect local offices to the VPN. They ship a green box to the Greenpeace staff in the new office, none of whom are technical staff. One of the staff hooks up the green box to their Internet connection, turns it on, and then is talked through a very basic set-up procedure by a technical staff person on the phone. If they run into problems, the staff person in the new office hooks up a modem to the green box and the DC office solves the problem remotely.

Business Problem #2

The Greenpeace DC office then wanted to move to Linux for their own staff's computing needs. This was partly in agreement and accordance with the parent organization's pro-open-source policy, partly out of a desire to save money, and partly in order to create a more uniform IT environment instead of having a Novell or Windows network internally, and a VPN built out of Linux machines.

Approach to Problem #2

Laptop Experiment

To begin expanding the use of Linux at their DC headquarters, Greenpeace DC equipped six of their organizers with laptops running Linux as the operating system and OpenOffice for word processing and spreadsheets. There were a few file conversion issues between OpenOffice and Microsoft Office, but aside from that it went fairly smoothly for the organizers.

Server Change-Over

As a second step, they began changing the network servers to Linux from Novell. This came at a time when they would otherwise have had to upgrade their Novell server software. With 100 users, the license upgrades alone would have cost several thousand dollars.

At first they used Linux servers to host their database, inventory, and calendaring software, plus firewall and proxy servers. They use the open source databases called MySQL and PostgreSQL, and proprietary software for inventory and calendaring.

They then began moving from NT to Linux for file serving and print sharing. At the time they were interviewed for this study, 10 of their 16 servers were running Linux. Out of 100 people, 60 used Linux for file sharing and printing, another 15 for just printing.

Tom Camerlinck, the IT manager, says that moving from Novell to Linux was "tedious." Printing was particularly difficult, because at the time they started moving over there wasn't great software available for printing services. Camerlinck also found Samba to be tricky to set up: "If you don't get it, hire a consultant," he advises.

Once they move the entire network file and print servers to Linux, their e-mail system is all that will remain on a Novell server. They use Pegasus for e-mail. Its biggest advantage is that it is very easy to share e-mail folders with several people, which approximately 20 percent of their users do, mostly for organizing work. Pegasus doesn't run on Linux -- in fact, they're using a very old version of Novell (3.2) to run it, largely because the old version is extremely stable (when they moved, it had been running for 600 days) and doesn't cost them money anymore.

Business Impact

One of the issues they struggle with is difficulty finding good Linux support. Cameron says it is harder to find Linux experts, "but if for example they know Samba, they know what they're doing. In contrast, there are too many people who have taken Windows classes but don't really understand the system." And he notes, "If you have somebody good, it [the total cost of using Linux] is really cheap."

LINUX ON THE DESKTOP

■ CASE STUDY #5: GRO

GRO, or Grass Roots Organizing (www.gromo.org), is a community based nonprofit organization of low-income families in Missouri. They have three full time staff members in two offices in Missouri and a board of about a dozen low-income activists spread across the state.

The Low Income Networking and Communications Project (LINC, www.lincproject.org), a project of the Welfare Law Center, works with welfare rights organizers around the country to improve their access to and strategic use of computer technology. LINC's Circuit Riders, Dirk Slater and Arif Mamdani, were interested in improving LINC's ability to provide stable desktop systems using donated computers for board members and leaders of low-income led organizations.

Background

LINC began to explore the idea of using GNU/Linux when they learned of Microsoft's decision to discontinue support of older operating systems, including Windows 95. With or without manufacturer support, LINC was also troubled by frequent crashes, prevalence of viruses and other problems associated with Microsoft Windows. LINC was looking for a pilot organization where they could improve their ability to provide stable desktop systems using donated computers to leaders of the low-income led organizations that they serve.

GRO wanted to get its leadership on computers and online. A local university had donated 14 Pentium computers, and GRO's director, Robin Acree, turned to LINC for help getting the computers running in board members' homes.

LINC's work with GRO proceeded in two phases. First, Arif Mamdani and Dirk Slater went to Mexico, Missouri to install GNU/Linux on these donated computers for GRO's board. A few months later, Mamdani went back to Missouri to follow up with GRO's leadership and set up open source desktops and a server in the GRO offices. The board members and staff organizers came from a range of backgrounds: some had almost no computer experience whatsoever, while others were used to using Windows and Microsoft Office.

The decision to use Linux was easy – Microsoft was phasing out support of Windows 95, the other obvious choice for older, slower computers. All the donated computers were 100 MHz Pentiums with 32 to 64 MB of RAM. The machines barely met the minimum requirements for Windows 98, for which Microsoft also expected to phase out support. The local Linux Users Group was prepared to work with GRO and support their computers, though GRO has called on them

more for software training and installation than actual maintenance of their systems.

LINC decided to install RedHat Linux 7.x on the GRO computers. LINC installed OpenOffice.org (an OSS suite that includes word processing and spreadsheet applications), Kmail (an e-mail client), Mozilla (web browser), and Konqueror (a web and file browser) on the desktop machines. In the office Mamdani set up an internal web server running Apache, PHP and MySQL and a proprietary calendar program built in PHP. The GRO website (hosted offsite) uses an open source content management tool called Pagetool.

Problems Encountered

LINC was starting out with faulty hardware – not all the CD drives worked, not all the network cards and modems worked, and the computers are still slow machines, no matter what the operating system. For GRO leaders and organizers who were used to working in Windows, adjusting to differences between operating systems was frustrating at times and OpenOffice.org can be buggy. Most basic tasks worked just fine, but accomplishing some tasks in OpenOffice.org took some adjustment.

The first time GRO tried to make mailing labels from their database, they discovered that OpenOffice.org could only make a single page of labels at a time. A Circuit Rider at the LINC office was able to find a workaround for the problem, but it wasn't something GRO staff would have stumbled on. Ultimately this allowed LINC and GRO to contribute to the development and usability of OpenOffice.org by identifying the bug and reporting it to the development team who have corrected the problem in the most recent release of OpenOffice.org.

Business Impact

When LINC followed up with GRO about how they were using their new computers, staff described saving documents and reusing language from previous letters. They talked about being able to check e-mail regularly, get messages out and assimilate responses quickly, keep presentation materials up to date and update their website. Regular access to e-mail has allowed them to network with organizations doing similar work, and coordinate their campaigns.

GNU/Linux and the open source software available to use on it allowed GRO to set up a computer for everyone on staff and for their active leadership. Without Linux they wouldn't have been able to use the computers that were donated to them reliably. With Linux, they all feel that they can work more efficiently and more effectively.

Six Steps You Can Take to Use Open Source Software to Address Needs in Your Organization

We have outlined so far the basics of OSS, how it's developed, why we think the OSS model is valuable for nonprofit organizations, some specific tools to use to make decisions about the adoption of OSS in your organization, as well as some specific examples of organizations that have adopted OSS as a part of their technology infrastructure. We outline here six steps you can take to begin to put OSS to work in your organization, and in the process learn more about it, and its capabilities and cost-effectiveness.

Step 1: Shared Web Hosting

It is very common for small and medium-sized nonprofit organizations to purchase a web and email hosting account from an external virtual hosting provider. These accounts cost from \$10 to \$40 per month. This is because external hosting (also called virtual hosting) requires less support and is less expensive.

There are many, many virtual hosting providers, and the vast majority of them use an open source operating system, either Linux or BSD (another Open Source UNIX variant). They use these because they are more cost effective and stable, and it is easier to administer many machines with fewer staff, than using Windows.

If you are already using a virtual host for your website, and you did not specifically ask for Windows then you are very likely using the open source operating systems Linux or BSD already, and the provider is almost certainly using Apache. You also likely have access to open source application development using the quite popular languages PHP and Perl, and the database system MySQL. Thus, you already have experience with OSS, and use it everyday, and you can check off Step 1! (Step 5 of this section will explore more on how to do it yourself).

Step 2: OpenOffice.org and Mozilla

Word processing, e-mail, web browsing and spreadsheets are the primary software programs used by nonprofit staff members. Fortunately, the proprietary software programs typically used to perform these functions all have well-developed open source alternatives that run on Macintosh and Windows platforms in addition to Linux.

You can download and install one or both of OpenOffice.org (<http://www.openoffice.org>), or Mozilla

(<http://www.mozilla.org>). OpenOffice.org is a full-featured office suite that can read and write Microsoft Office files (.doc, .xls, .ppt), and Mozilla is an open source program that does web browsing, e-mail, IRC and HTML editing. (In some versions of this booklet, a CD is included providing these programs; they also may be downloaded at no charge.) Both of these software packages install easily. They are easy to try out and evaluate.

Step 3: Small Desktop Trial

If some of your staff are primarily using only the programs mentioned in Step 2, then you could experiment by installing Linux on an extra workstation on your internal network. In addition to providing the applications mentioned in Step 2, Linux comes with many other multimedia and productivity applications.

To evaluate using Linux on the desktop, you can take an old desktop that might be gathering dust in the corner (preferably a Pentium processor of 400 MHz or better), and install a distribution of Linux on it. One of the easiest and most popular Linux distributions is RedHat. It is easiest to go to a major computer store and buy a boxed copy of the most recent version of a number of Linux distributions, including SuSE, Linux-Mandrake and RedHat Enterprise). There is a wide range of costs for these, from about \$30 to much more, depending on the distribution and the version. You can also get CD images of a number of distributions, such as Debian (<http://www.debian.org>), Fedora (<http://fedora.redhat.com/>)

A SECURITY NOTE: Don't place this test Linux desktop on a static public IP address without NAT or without being behind a firewall (talk to your tech staff member or consultant if you have one). Although generally regarded as more secure than Windows, like any computer you put on your network, you need to be aware of how to make it secure before it is open to the public Internet.

and Knoppix, which is a good distribuion to try, since you can even just boot from the CD and tes it out (<http://www.knoppix.org>.)

This will give you an idea of how to use Linux on the desktop, and introduce you to a wide range of OS packages for you to test out. It is a good way to understand how Linux works. In addition, there are several ways (see list below) to use Windows software on your Linux desktop, if needed.

Step 4: Network File and Print Server

One of the easiest ways to use Linux in a networked environment is to use it as a file and print server, to replace or retire the Windows server that you might have serving this function (note: a dedicated file/print server is recommended for organizations with 7 or more staff). The case studies show examples of the use of Linux for just that purpose. SAMBA allows the Linux server to share network directories (folders) so that they can be accessed by Windows clients.

If you would like to use Linux as a print server, and you have an unusual (or very new) printer, we recommend checking out <http://www.linuxprinting.org> to make sure that Linux supports your printer.

Step 5: Self-hosting of Web and e-mail/e-mail Lists

As mentioned above in the virtual host section, Linux is very good at Internet server functions (web and e-mail hosting, and other Internet server functions). If you have a DSL connection with a **static IP address** (you generally have to pay more for an account like that), or a T1 or higher broadband connection, then self-hosting your website and e-mail is quite easy using Linux. You can easily use an older server machine or desktop for this function. Again, you can find or download any distribution of Linux that you like.

If you do not want to take on the responsibilities and cost of hosting your server yourself, you can get a dedicated Linux server from many hosting providers, starting at around \$99/month. With this kind of server you can install any specialized OSS that you might want to use in your organization.

Unlike MS Windows servers, Linux comes with all necessary server functions in the box, and there are no per-seat licenses for anything (Windows servers do come with Internet Information Server (ISS), the Windows web server that has no additional license fees, but all additional server software, like e-mail, requires additional costs). So unlike

Exchange, where you have to spend \$2.50 (discounted) to \$40 per e-mail account, Linux will allow you to have unlimited e-mail addresses at no additional licensing costs. Linux comes with Apache, the most popular web server. There are a number of mail servers that are available, including send-mail, postfix and exmlm.

E-mail lists (discussion lists, e-newsletters, fundraising appeals) have become more and more important to non-profit organizations. There are a number of OS mailing list managers for Linux/UNIX, with a broad variety of functionalities and ease of use. Probably the most popular and easiest to use is a program called Mailman. Others include majordomo, Sympa, Smartlist, and ezmlm.

Step 6: Moving Towards An All Open Source Office

There are a variety of other open source tools that can allow you to move to an entirely open source office.

Database servers: There are two database servers that are often used in Linux/UNIX environments (and both have been ported for use on Windows): MySQL and PostgreSQL. They are both popular, although MySQL is the most popular. They can be used for any basic DBMS functions that MS SQL server (or even Oracle) can be used for. MySQL is most often used for web-based databases, and PostgreSQL is considered a possible replacement for Oracle, because of how full-featured and robust it is. Both can be used as back ends via ODBC, with MS Access serving as the **GUI** front end

There are also the membership and donor management packages eBase (www.ebase.org) and ODB (www.organizers-db.org), and although they are built upon proprietary development environments (FileMaker and Visual Basic 5 respectively), they provide access to the source code. There is an open source server-based accounting package, called SQL-Ledger, which some nonprofits have begun to use, and a desktop accounting package called GNUCash for Linux.

Undoubtedly, the options will improve as developers realize that there are needs to be addressed. We have generated an on line database (<http://www.nosi.net>) of open source projects that are specifically of interest to nonprofit organizations, and a CD (distributed with some versions of this booklet) that provides installation programs for relevant Windows and Mac-compatible OSS programs.

On the next page are two tables that compare and contrast proprietary software options with open source software options, both for the desktop and the server.

Server Software

TASK	PROPRIETARY OPTIONS	OPEN SOURCE ALTERNATIVES	COMMENTS
File sharing	Microsoft Windows 2003 Server	Samba running on Linux or BSD	Samba is very mature and robust
E-mail server	Microsoft Exchange Server, Lotus Notes	Sendmail, Postfix, Exim, SuSE Open Exchange	
Web server	Microsoft IIS	Apache	Apache is the most popular web server
Database server	Microsoft SQL Server Oracle	PostgreSQL, MySQL	Both projects are very robust and full-featured, and will run on Windows as well as Linux/BSD

Desktop Software

TASK	PROPRIETARY OPTIONS	OPEN SOURCE ALTERNATIVES	COMMENTS
Office Suite	MS Office, Corel Word Perfect Office	OpenOffice.org, Koffice, Abiword, Gnumeric	Gnumeric is comparable to Excel
Financial	Quickbooks, Blackbaud, Peachtree	GNU Cash, SQL-Ledger	GNUCash is not as complete or polished, but can be quite adequate for smaller nonprofits. SQL-Ledger is actually a web-based accounting package that is mature enough to be used by even large nonprofits
Web Design	Front Page, Dreamweaver	OpenOffice.org, Bluefish, Mozilla, Quanta	On the whole, OS alternatives produce much cleaner HTML, and are as easy to use, but not as full-featured
Graphics/Desktop Publishing	Photoshop, InDesign, Quark	The Gimp, Scribus	
Fundraising/ Contact Management	Raiser's Edge, Paradigm	Ebase for Mac or Windows, Organizers Database (ODB) for Windows	Ebase and ODB are open source, yet for now are tied to proprietary back-end formats (Filemaker and Access)
Project Management	Visio, MS Project	MrProject	

Further Reading:

<http://www.linuxprinting.org/kpfeifle/SambaPrintHOWTO/Samba-HOWTO-Collection-3.0-PrintingChapter-11th-draft.html>

<http://www.linuxprinting.org/>

<http://networking.earthweb.com/netos/article.php/15824>

How to Find Support of Open Source Software

One of the major questions that nonprofits have about implementing OSS in their organizations is "How will I get support?" We outline here the variety of ways to get support for OSS, and the increasing awareness and expertise about OSS in the nonprofit sector.

Technology Support Model

Technical support for any technology product can come from four places: internal staff, technology consultant(s), software developer/vendor, and the larger community (whether it be community of nonprofit technology workers, like the Circuit Riders, the community of users of a particular product, or the larger user community on the Internet). Support for any software product is either for an end-user, or for the person who administers that software in an organization (these may be the same people).

The standard paradigm for support for proprietary software used to be that when you bought the software, support for installation and troubleshooting problems was free for as long as you owned the software. Training on how to use that software was sometimes provided by the software vendor at a cost, or often provided by consultants or other organizations/companies. This is to some extent still the case for large, expensive and "niche" packages such as client management and fundraising programs.

Slowly, vendors of more commoditized proprietary software (like word processors and operating systems) have reduced the availability of free support over the past few years. In general, support for most proprietary software at present is only free for a very limited period of time, or only for installation issues, then you have to pay for it, often by the incident. Some software vendors provide no free support at all.

Support for OSS will come from those same four sources, but the emphasis is different. As a general rule, OSS is not created by, or supported by a company that you can call on the phone or e-mail for support. (There are exceptions, like some Linux distributions, and some packages like MySQL where you can purchase support). However, the community of support for open source software is, in general, much richer than the communities of support for proprietary software.

There are some ways that OSS may lag behind proprietary software in terms of support – OSS documentation tends to not be as user-friendly, and there are fewer available printed books on many OSS packages (however, there are many books on the most popular applications). In addition, as we'll describe below, it can be harder to find local, or sector-specific OSS expertise from consultants.

There is an increasing amount of support available for OSS, due to the following five factors:

- Open source applications have developed sophisticated user communities, and have even created free web services and applications (like SourceForge and PhpBB) that enable a volunteer-based community to collaborate on answering questions, creating tutorials, and reporting bugs and requests for new features.
- The nonprofit community itself, including foundations, independent consultants, and NTAPs (nonprofit technology assistance providers) is recognizing the inherent advantages of open source software tailored to the specific needs of the nonprofit groups they support.
- The producers of mature open source applications (PHP and MySQL are two examples) often sell their software bundled with support. Since the software itself can be obtained for free, "value added" is essential to create the sales needed for a viable business model. Other OSS producers do not offer support themselves, but refer users to a growing number of small support providers. The cost is often competitive with the price that larger firms charge for support of popular proprietary applications. And it is optional: if you can do without phone support, you do not have to pay for it.
- Where software engineers once dominated the open source community, people with additional skills (technical writers, usability experts, etc.) are beginning to more actively participate.
- Several established computer companies have recognized a self-interest in supporting the development of open source options, and are now investing tens of millions of dollars (or more) each year to address the shortcomings that hinder widespread OSS adoption.

We will focus on the first two of these, which are most relevant to nonprofit organizations.

Community Support

Both developers and end users have been active in creating rich online communities where peer-to-peer support for OSS can be found. There are many mailing lists, website and even chat rooms devoted to users of open source software products. Users of the Linux operating system can find support by seeking out the nearest Linux User Group. We list some of the possible sources of Linux community support below. If you do seek help from other OSS user or advocates, you will likely notice a culture that is cooperative and giving,

consistent with the charitable nature of most nonprofit organizations. In many cases, the developers themselves monitor these lists, and will participate in answering questions.

Each Linux distribution has one or many mailing lists that you can use for support. In general, those tend to be high traffic, and there are generally people who are very familiar with a particular distribution on those lists. Increasingly, you can find community support on more general nonprofit technology lists, as OSS is used more often in the community.

In general, it is surprising how many people have had the same question you might have, and have gotten it answered in some online forum or list. Typing the key portion of your question in Google will often lead to successful resolution of many issues.

Nonprofit, Consultant, and Foundation Support

The other trend we will address, and the most important in our opinion, is the recognition within the nonprofit and foundation community that OSS deserves more attention.

Three years ago, there were only a handful of nonprofit consultants who advertised expertise in open source solutions. Today there are literally hundreds of consultants available, and you can find them through the consultant databases available through Idealist, CTCnet, and NTEN (see list on page 24). There also exist about a dozen nonprofit technology assistance providers (see list on page 24) that provide support for OSS. Finally, there are a number of software applications designed specifically for nonprofit organizations' needs

(eBase, Organizers Database, PageTool, XINA) that are either fully open source or are moving in a more open source direction.

These trends are now gaining attention among philanthropists. Tech-oriented philanthropist Mitchell Kapor recently founded the Open Source Application Foundation, a nonprofit dedicated to the creation and maintenance of quality end-user open source software. Philanthropic organizations like the TechFunders Collaborative and the Technology Assistance Group of the Council on Foundations are including open source sessions at their conferences. Some foundations are now requiring, as a condition of their grants, that any software developed by grantees must be made available, under an open source license, to the entire nonprofit sector.

The Meyer Memorial Trust, IBM, and the Open Society Institute (OSI) are beginning to fund the development of and support of OSS for the nonprofit sector. Jonathan Peizer of OSI, circulated a 9-page paper in September 2003 entitled "Realizing the Promise of Open Source in the Nonprofit Sector." The paper addressed the challenges of developing nonprofit-specific open source applications, including the need to develop a rich support infrastructure that can be made available along with nonprofit-specific software titles.

These developments indicate that the days when OSS was relevant to computer geeks are now over. In the next year, there is no reason to believe that the pace of these changes will slow down.



Conclusion

Open Source Software is rapidly developing, and new alternatives for nonprofit organizations are emerging and maturing. We have described here the types of software now available to nonprofits, and how they are being implemented in organizations.

As we have articulated, there are many places where OSS is not as mature or ready for use in nonprofits as older proprietary applications. But it is our opinion that in the next two to five years, the maturity, stability and ease of use of OSS will rival, if not exceed that of standard proprietary solutions.

This will not happen by accident. In a paper recently published by the NTEN, Foundation executive Jonathan Peizer has argued that to realize the full promise of OSS in nonprofits, "a proactive, well thought out strategy by a collaborative of progressive funders, developers and technology

service providers" is required. (<http://www.uploads.nten.org/gems/RealizingthePromiseofOpenSou.pdf>).

And that is also why we at the Nonprofit Open Source Initiative (NOSI) are engaged in a series of activities to better educate the community. These activities include providing the NOSI open source café at nonprofit conferences, the publication of additional case studies on our website, the development of a nonprofit-focused OSS mailing list, in addition to dissemination of documents like this primer.

We hope that this introduction to the potential of OSS within nonprofit organizations will make it possible for you to participate in this movement, whether it be through NOSI, or other organizations of which you are a part. Thanks to the power of collaboration, the full potential of open source software in the nonprofit sector is only beginning to be realized.

Mailing Lists, Websites and IRC Channels Available for Support of OS

- <http://www.justlinux.com/>
- <http://www.tldp.org/> – The Linux Documentation Project
- <http://www.linuxhelp.net/>
- <http://www.linuxchix.org> – Multiple mailing lists
- irc.us.freenode.net – Many distribution specific channels, like #debian, #gentoo
- mozilla.org – Support for using mozilla
- nosi-discussion@nosi.net – Discussion of open source software
- nosi-desktop@nosi.net – Discussion of open source software for desktops
- riders-tech@npogroups.org – Circuit riders technology discussion

- **CTCNET** (www.ctcnet.org) – A network of community technology centers serving urban and low-income areas
- **Free Geek CTC** (www.freegeek.org) – Developing a tech assistance project using OSS
- **Free SW Foundation** (www.fsf.org) – Advocates for free software, enforces GPL license
- **Gilbert Center** (www.gilbert.org) – Strong advocate of open source
- **Idealist.org** (www.idealist.org) – Online directory of nonprofit consultants
- **LINC Project** (www.lincproject.org) – Open source for low-income comm.
- **LSTech** (lstech.umich.org) – A technology assistance provider serving legal assistance communities
- **Tactical Technology Collective** (<http://www.tacticaltech.org/>) – An international organization working to bring technology (including OSS) to NGO sectors in developing countries
- **Media Jumpstart** (www.mediajumpstart.org) – Technology collective for nonprofits employs OSS
- **NTEN** (www.nten.org) – Provides a directory of nonprofit consultants
- **One Northwest** (www.onenw.org) – Environmentally focused provider includes open source options
- **Organizers' Collaborative** (www.organizenow.net) – Promotes, creates open source tools for organizers
- **Open Source Application Foundation** (www.osafoundation.org) – Nonprofit established to write open source software
- **Debian-NP** (<http://www.debian.org/devel/debian-nonprofit/>) – A specific Linux distribution project for nonprofits

Open Source Primer Glossary

Applications – software on your computer that does basic user tasks, like word processing, accounting, etc.

CD Image – This is a large file you can download (only if you have a fast Internet connection and lots of space on your hard drive), that usually contains the entire contents of a software installation CD. With this file – that usually has a ".iso" file extension – you can use any CD burning software program to create copies of the CD.

Firewall – a hardware or software device that controls what traffic comes in and out of a network or computer. It can protect networks and computers from intrusion and worms.

Fork – When a developer takes an open source project developed by someone else in an entirely different direction. This is done for a variety of reasons – disagreements with the philosophy of the original developer, slowness in development of the original, different purpose for the new software (an example of this is FilmGimp, a fork of the Gimp image processing program developed specifically to work with film).

Free Software – Free software, in this context, does not just mean software that has no license or acquisition fees. It pertains to software that is written under an open source license, and can be freely copied, modified and re-distributed (and you are free to sell it.) See <http://www.fsf.org/philosophy/free-sw.html> for a full definition.

GUI – Graphic User Interface – a method of interacting with a computer program that allows point and click with a mouse as opposed to typing commands on a command line. For example, Windows and Macintoshes use a GUI, DOS does not.

HTML – Hypertext Markup Language, the language used to create web pages.

IP Addresses – any computer that connects to a network has to have an IP (for Internet Protocol) Address, so that other computers know how to get at them (to send web pages that were requested, for instance). An IP address is in the form of 4 numbers separated by periods. These numbers range from 0.0.0.0 to 255.255.255.255 giving around 4 billion IP addresses.

IRC – Internet Relay Chat – IRC is a way to chat with others that are on the Internet in real time. There are many IRC clients for all platforms. IRC has been around for a very long time, and has many purposes, but there are several IRC servers that are dedicated to providing support channels for open source software (Freenode.net is the major one).

Kernel – the software that controls the most important tasks on your computer. It is responsible for process management (what applications are running and how), disk management and memory management.

Linux – A UNIX variant, is the most commonly known open source operating system. It was developed in the early 90s, and has become quite mainstream, especially used as servers.

ODBC – Open DataBase Connectivity – is a standard that allows databases to talk to one another easily, either within one system, or across a network or the Internet.

Operating System – The entire collection of kernel, libraries and utilities that provide the interface between the hardware and the user (and other applications).

Static Public IP Addresses – An IP address that is permanently assigned to one server. Generally that server is listed in the Domain Name Service (DNS) as connected to that IP address.

Version Numbering – All software that is released to the public (and most that is not) is assigned a version number, which helps to keep track of the status of that software, and allows users to know whether it is the most up to date.

There is an Appendix and additional material on the NOSI website, which includes detailed information on specific open source projects, a database of OSS for nonprofits, additional case studies, and an ever-changing FAQ. Find these at:

<http://www.nosi.net/primer>

I want to help NOSI promote nonprofit use of Open Source Software!

Individual Donation Options: ____\$50 ____\$100 ____\$250 ____\$500 ____\$1000 \$____ Other

Corporate Support (support for NOSI cybercafes at nonprofit conferences):

____ \$500 (contributor) ____ \$1000 (sponsor) ____ \$2,500 (underwriter)

____ If you gave \$250 or more and would like your gift acknowledged on our web site, please check here.

If paying by check, make your check payable to "CFNCR/NOSI." Please photocopy this form and send with your check to:

Nonprofit Open Source Initiative

c/o CFNCR

1201 15th Street, NW, Suite 420

Washington, DC 20005

Or you may pay by credit card. Please provide your credit card information below:

Circle for credit: MC | Visa Card No: _____ Expiration Date: _____

Signature _____

Also please provide address below:

Name & Contact _____ Email _____

Address _____ Phone _____

City _____ State/Prov. _____ Zip _____

Please see our website at www.nosi.net for more information on how to get involved.

Accolades for
Choosing and Using Open Source Software:
A Primer for Nonprofits

"NOSI has developed a terrific resource that can help nonprofits everywhere understand the significance of the transition from proprietary software to open source."

— *Mitch Kapor, President,
Open Source Applications Foundation*

"This booklet will be enormously helpful to nonprofits, who would find open source solutions very useful and affordable, if only they knew they existed. This is an important first step in delivering the good news..."

— *Marie Deatherage,
Program & Communications Officer,
Meyer Memorial Trust*