Choosing Open Source:

A decision making guide for civil society organizations

Developed by:
Mark Surman and Jason Diceman
The Commons Group

This document was develop by The Commons Group for the Association for Progressive Communications (APC). APC has published a curriculum version of this guide at www.itrainonline.org .

What is this guide?

Open source software holds a great deal of potential for civil society organizations. The most obvious benefit of open source software is free to use, something that is attractive to organizations on a tight budget. However, it also offers much more – better security, increased flexibility and the ability to adapt software to meet local and organizational needs. And, open source software is based on the kind of collaborative and cooperative principles that many civil society organizations embrace.

This document helps to unlock the potential of open source for civil society. It provides an introduction to the topic, tackling questions like 'what is open source?' and 'how will it benefit my organization?' It also includes practical, down to earth advice on how to review open source software packages and select the right ones for you.

Who is this document for? Civil society managers responsible for the implementation of IT systems – whether they are IT specialists or not. This document is not a detailed technical manual, but rather an organizational planning and decision making tool. You don't need to be a computer expert to benefit from this guide.

What is open source?

The term "open source" refers to software that can be freely redistributed, analysed and modified by anyone. Key characteristics of open source software include:

- **Free:** The software doesn't cost anything to obtain – although there are costs involved in configuring and maintaining it, just like with other software;

- **Open:** Because the software code is open and readable by anyone, it also means that the software can be modified by anyone with programming skills ... allowing for the translation of software into local languages or the addition of features needed by a particular group;

- **Collaborative:** Open source software draws its strength from the fact that people who improve, modify or customise programs must give it back to the open source community so others can benefit from their work.

How does a piece of software become open source? The people who created it decided to release it under a license such as the General Public License (GPL) that meets the criteria of the "open source definition" outlined above. It is often said that open source licensing is "viral" because people who modify a piece of software **must** release their modifications under the same open, free and collaborative conditions of the original software.[i]

> ***Open source: the electronic equivalent of generic drugs***
>
> *Like the generic drugs that have transformed health care provision in the South, open source software is royalty and license free, and is therefore substantially cheaper to acquire than branded alternatives. The reason for this is that open source software is developed by volunteer collectives who are not seeking to profit from its sale.*
>
> *In addition, just as the recipe for generic drugs is made public, so the source code or inner workings of open source software is accessible to the user. Any qualified person can see exactly how the software works and can easily make changes to the functionality.*
>
> *- IICD Open Source in Africa Brief*[ii]

Who's using open source?

More and more organizations in business and government are turning to open source. Large computer companies like IBM, Sun and Apple have embraced open source as a part of their software strategy. Governments have turned to open source both as a way to share software they create and as a way to lower costs, with the City of Munich switching all 14,000 of its computers to Linux, an open source operating system. And, of course, the large majority of computers used to run the Internet run open source tools, with 65% of web servers currently running the Apache web server.

The same trend is beginning to happen within civil society, especially in relation to server software and web applications. Civil society examples include:

- An estimated 90% of **Greenpeace** servers worldwide run Linux. Also, Greenpeace's global virtual private network is built on top of open source tools.[iii]

- **SchoolNet Namibia** uses all open source operating systems, e-mail clients and office applications tools to provide Internet access and training to the nation's schools.

- Human rights organizations around the world have begun to adopt the **Martus** software, an open source package that allows users to document incidents of abuse by creating bulletins, uploading them at the earliest opportunity, and storing them on servers located around the world.

- The **Association for Progressive Communications** uses open source for a wide variety of applications including its global network of web mirrors that are used to fight corporate and government censorship around the world.

- Heritage seed bank **Seeds of Diversity** uses open source database tools to store and management of horticultural data.[iv]

- **OneWorld.net** has adopted open source content management and online community systems to run its global network of portals.

These examples are only the tip of the iceberg. This is because the open source revolution is in some ways a quiet one. Most organizations just download open source and use it, not taking the time or effort to tell anyone about it. Yet, certainly, the numbers are growing.

Why use open source?

There has been a great deal of discussion of the benefits of open source for civil society of late. There have been workshops, papers, speeches and even manifestos expounding on the virtues of the open source / civil society connection. Amidst all of this, it may seem that open source is being presented as some sort of cure all. Certainly, it is not. Yet, used with an understanding of its limitations and challenges, open source software does present some clear benefits to civil society organizations:

- **Lower information technology costs:** Open source eliminates the need to pay for software licenses. Of course, this does not mean that you can eliminate information technology costs altogether – there is still a need for staff and equipment to make it all work. But open source can definitely lower overall costs, especially for organizations which need advanced servers, databases and web applications.

- **Flexible software solutions:** The fact that the code is "open" means that software can be modified to respond to needs not addressed by the original developer. The most obvious benefit of this responsiveness is the ability to translate software into local languages. However, it is also useful for web applications where it may be desirable to add on new features that accommodate the unique needs of civil society organizations.

- **Better security:** Mature open source applications – Linux, Apache, SendMail, OpenOffice.org – tend to be more secure than their proprietary counterparts. This is partly due to the fact that the code is open, allowing system administrators and others concerned with security to quickly identify problems and propose solutions to the lead software developers. Also, open source applications tend not to be targeted by people who make viruses. This kind of improved security is vital for civil society organizations who want keep unwanted "snoopers" out of their network and who cannot afford the downtime that comes with virus attacks.

- **Collaborative, cooperative values:** Open source is based on a set of collaborative and cooperative values. It assumes that we can create more useful tools by working together than we can by competing. Using open source is not only a way to support these values, but it can also provide a way to learn from them. The open source community presents a very practical and adaptable example of how organizations can work collaboratively.

It is interesting to note that civil society organizations are starting to turn to open source methods for more than just software. The "free, open, collaborative" model provides a useful approach for developing written content, multimedia, analytical and evaluation frameworks and many other "knowledge products". For example, this guide has been released under a Creative Commons open content license that encourages others to modify it as long as they release their modifications for free.

More information on the benefits of open source are included in the "Benefits and challenges' section of this guide.

What's in a name?

There are many debates in the computer world about whether to use the term "open source" or "free software", and currently a variety of terms and acronyms are being used to describe the concept discussed here: open source, free software, FOSS (free and open source software), FLOSS (free/libre and open source software). While many people feel strongly about which term is used, the core concepts being described are the same. "Open source", "FOSS" and “FLOSS" all refer to the idea of free, open and collaborative software. For the sake of simplicity we have just used the term "open source" in this guide[v].

Other related software license terms are defined in the Glossary component of this unit.

Choosing open source

As mentioned above, this guide has been designed to help civil society organizations select open source software solutions that will best meet their needs. With this goal in mind, the guide has been divided into five sections:

- **Open source landscape:** Overview of the different types of open source software. Also includes civil society open source case studies.

- **Why open source?** Overview of the primary benefits of open source for civil society organizations.

- **Questions about open source:** Responses to frequently asked questions about open source.

- **Choosing the right software:** A step by step process for the review of open source software options.

- **Beyond decision making:** Information about costing, migration, training and other issues that must be considered after you have selected your software.

Taken together, this information should give you a basic understanding of open source and how it can benefit your organization. You should also have the skills necessary to review and select open source software packages.

The open source landscape

When they hear the term "open source", many people immediately think of Linux – the free open source operating system that has become popular as a server platform over the past 10 years. However, there is much more to open source than just Linux. In fact, there are at least three distinct open source software "types" – server software, desktop applications and web applications. In order to understand the open source options for your organization, it is important to first understand these different categories of software.

Server software

A server is a central computer that is used to store data and house applications that are shared by many different people. Typical uses of a server within a civil society organization include storing documents that need to be shared, hosting web pages and other online tools, providing e-mail gateways and housing information repositories and directories. Common server software components include: a network operating system, a database system, e-mail and web servers and programming tools.

It is in this area of server software tools that open source is both best known and most mature In fact, the open source web server Apache is by far the most popular web server on the Internet, far surpassing similar software provided by Microsoft and others. The open source e-mail server SendMail is also leader in its field. Almost any software tool that you would want to run a server is available as open source, and most are packaged automatically with Linux distributions.

If setting up their own servers, civil society organizations should seriously consider open source software. The features and choices available are equal or better than commercial options – and the software is free. In contrast, commercial server software tends to be very expensive, much more so than commercial desktop software. Also, open source server software is often more secure than commercial equivalents

Open source servers support Namibia's schools ...

SchoolNet Namibia is non-profit provider of internet services, hardware and training to the nation's schools. It uses open source server tools to provide inexpensive LANs and ISP services. Examples include the Squirrel Mail web mail system and SuSe Linux operating system. Using these tools, 200 Namibian schools and institutions are connected with full feature internet services at around 60% cost savings from Microsoft alternatives.

Desktop applications

When most people hear the word "software" they think of the e-mail and word processing applications on their own individual computer. These applications fall into the category of "desktop applications". Common desktop applications include e-mail clients, web browsers, accounting software, spreadsheets, and presentation tools.

The availability of open source options is much more patchy in the area of desktop applications than it is in the realm of server software. In certain areas – e-mail clients, web browsers, instant messaging clients – there are good open source options that work on all operating systems including Windows, Macintosh and Linux. In other areas such as word processing and spreadsheets, good open source options such as OpenOffice.org are emerging. However, the fact that so many people are already using proprietary tools like Microsoft Office makes it difficult for many organizations to adopt these tools. And, there are some areas – graphic design tools and accounting software, for example – where good open source options exist only for Linux or not at all. It should be noted, however, that the area of desktop applications is changing and improving quickly, with major companies like IBM and Dell committed to making desktop Linux a reality.

Deciding whether or not an organization should adopt open source desktop software requires systematic needs analysis and the use of decision-support tools such as those included in the

MMTK unit on "Choosing Open Source Software". However, there are some general rules and guidelines which can help you start thinking about open source. If starting from scratch or only deciding on software for a small group of users, open source tools are probably a good bet in many categories. However, if an organization already has an installed base of proprietary software, more caution should be used. The costs of switching from proprietary to open source on the desktop are not only technical – users must be trained and allowed time to adapt to changes. This is a major cost that people often overlook.

Greenpeace Manila switches to the open source desktop

As a general rule, Greenpeace has decided to move to open source to save on license fees and avoid potential law suits that would be incurred from pirating proprietary software. The Greenpeace Manila office has taken this commitment all the way to the desktop, with almost all of their computers now running Red Hat Linux and OpenOffice.org. These computers are mainly used for word processing, e-mail, web browsing, spreadsheets and presentations. According to an article in *Linux Journal*, Greenpeace South East Asia says free software has saved the office "a lot of money" that naturally "is better spent on winning campaigns than paying for very expensive licenses".[vi]

Web applications

The final type of software to consider is "web applications" – the kind of programs that are used to make a web site and other online platforms interactive. At the generic level, this includes tools such as web discussion forums, online surveys, database-driven content management systems, groupware and online collaboration tools, and mailing lists. In a civil society context, it might also include online campaign tools, petitions, volunteer recruitment systems and other software designed specifically for the kind of work that civil society organizations do.

As with server software, there are many mature and well-established open source web applications to pick from. In areas such as mailing lists the "best" options are fairly well established and making a decision is reasonably easy. However, in many areas – content management, e-newsletters, online campaigning – leading options have yet to emerge So, while good options exist, it is sometimes difficult to know which option to choose. Our decision making framework below is designed to help with this.

In many ways, it is in the area of web applications where the idea of open source is most compelling for civil society organizations. Why? Because web sites need to be designed to reflect the "organizational logic" and strategic goals of the people who set them up. As a result, it is often desirable to modify or add a feature to a web application to match a particular organizational need. For example, an advocacy organization might want to add an online petition feature into its content management system. The open aspect of open source not only makes this possible ... but it also ensures that any changes made will always be open and changeable so that they can grow with an organization's needs.

Canadian environmental groups create network of web sites with open source

The Canadian Partnership for Children's Health and the Environment (CPCHE) – a coalition of 12 leading environmental groups – has used open source tools to create an interconnected network of NGO web sites. Using APC's open source ActionApps content management system, CPCHE has created a major portal site and upgraded six partner web sites so that content is easily managed by non-technical staff and automatically shared with the portal. Using open source tools meant that CPCHE did not have to buy expensive licenses for all of its partner sites. More importantly, the organization was able to use a small part of its budget to add new features to the ActionApps software such as the automatic distribution of articles by e-mail. With this approach, CPCHE got the software it needed and the whole ActionApps community received new features and upgrades.

The benefits and challenges of open source

As outlined below, the choice of which software to use for a particular application really should be driven by your specific needs and circumstances. However, there are a number of broad benefits and challenges that are common across the whole realm of open source. Understanding these principles will give you a general idea about whether or not open source is right for your organization.

Benefits of open source

There are many good reasons to consider open source software for your organization. Some of these reasons include:

- Software and updates can be obtained at low or no cost. No royalties of license fees.
- Software can be customized to suit specific user needs (e.g. creating Linux interfaces for more than 10 different Indian languages at indlinux.org[vii]).
- Upgrade development can be completed at a pace set by the users who pay for it.
- Much less likely to be dependent on a single software provider or trapped into long term software support contracts (no "lock in").
- Using open source makes you a part of a larger community that generally shares cooperative values.
- Software and documentation upgrade expenses can be distributed among groups.
- Creates an opportunity to hire small, local developers to improve the software rather than big foreign software publishers.
- Better security. Open source model allows more programmers to participate in the debugging of code. Specifically Linux and applications running on Linux have proven to be more reliable and secure than Windows equivalents. Also, less likely to have virus problems.[viii]
- Hardware flexibility. With some smart coding, Linux can be scaled to run on almost any kind of hardware. This is especially important for organizations who aim to use older computers.
- Bugs tend to get fixed more quickly.
- More likely to find and attract technical volunteers to work with open source applications.
- Usually use open standards that are gaining in adoption and most likely to be supported even more in the future (e.g. XML).
- Software creation is needs driven as opposed to commercially driven.
- Free and widely available support for popular applications (if you know where to look).

Common open source challenges

There are, however, some challenges that one may face using open source. Some of these challenges include:

- Open source desktop applications are less common in some application categories. Many categories of software needs are yet to be supported by mature open source applications.

- Some open source software packages do not have the same level of documentation, training and support resources as their common equivalents.
- Making an organization-wide switch from proprietary software can be costly. Sometimes the costs outweigh the benefits.
- Many open source tools are "designed for programmers" – they are not user friendly and therefore have a heavy learning curve.
- There are still real costs with open source, specifically around configuration and support. Many people get caught thinking that using open source will be totally without costs.
- Sharing files with proprietary applications can be difficult. This is a serious problem in the area of desktop applications such as word processing ... although this is changing.

It is important to realize that as open source applications mature and the user community grows, many of these challenges may be overcome.

Open source questions and answers

The following is a collection of questions and answers about open source software. These questions are asked quite commonly when an organization is new to open source.

Are any serious organizations using open source? As outlined above, an increasing number of civil society organizations are adopting open source, especially for server and web applications. Also, the majority of the internet servers are running open source web and e-mail servers. Major companies like IBM, Sun, Apple, Netscape, HP and Oracle have adopted an open source model for at least part of their business.

Will open source really save money for my organization? Generally, yes. The elimination of license fees is a significant cost saving. However, it is important to consider all costs including training and support. Open source tools can sometimes be more expensive in these areas.

Is open source software hard to use? While many open source applications are designed by programmers for programmers, there are an increasing number of packages that are easy to use and have been designed with users in mind. This said, a review of usability should be included in the assessment of open source software options – just as it should with commercial software. If an application is hard to use, it may not be the right choice.

Is open source software "buggy"? The bugginess and stability of any released software (i.e. version 1.0 or above) is dependent on the quality of the programming and interface design. Both proprietary and open source applications have bugs, but only FLOSS software lets you and your community fix the bugs yourself. Also, open source projects tend to make it easier for users to report and discuss bugs with developers than commercial software companies do. For example, most open source projects include a mailing list where users and developers can exchange ideas.

What are the support resources like for open source? Who do I call when something doesn't work? Support resources vary in quality depending on the software package in question. "Mainstream" applications like operating systems and office suites tend to come with good electronic documentation. Support can usually be obtained for a fee from companies like Red Hat or from one of the many consulting companies that are emerging to support open source tools. When you pay for this kind of support, you should receive the same kind of service and accountability you would expect from a commerical provider. It is also worth noting that most open source projects have free support forums which can be surprisingly useful.

Is open source sustainable as a business model? Open source is growing in popularity, quality and innovative business models. Some of these business models are commercial, with consulting companies using open source as a way to lower overall project costs and win more contracts. Other business models are non-profit, with civil society organizations banding together in consortia to create a software application that will benefit the whole group.

Isn't open source only Linux? No. While the Linux operating system is the most well known and popular open source software, it is possible to find open source software to meet almost any computer need. This includes e-mail, word processing, spreadsheet and web browsers.

Does open source software run on Windows? Many people assume that you need to be running Linux to use open source tools. This is not the case. Many open source tools have been written to run on Windows, Macintosh and other platforms. This includes everything from the Apache web server, to the PHP scripting language, to OpenOffice.org.

How many open source options are there? Is there open source software for my needs? There are full and growing suites of quality open source applications covering a wide range of user needs on a full assortment of operating systems. Open source software is available for all popular operating systems, including Mac, Windows, Palm, and most prominently Linux, which itself is open source.

Where do I find open source software? Popular Linux distributions with bundled software suites are available on CD-ROM at better computer stores. Online you can find hundreds of applications at Freshmeat.net, SourceForge.net and OSdir.com.

Does taking advantage of open source mean I have to work with people who speak in acronyms (a.k.a. "geeks")? There are many information technology professionals and technical consultants that can help implement open source solutions while communicating in common everyday friendly language.

Don't open source projects often get abandoned? Just as many software businesses go out of business, so may some open source developers decide to leave a project. But unlike commercial software, open source code is always available for other groups to pick up and continue, as was the case for GIMP[ix] and Apache[x].

Is open source software available on CD-ROM? Can I only download it off the internet? Linux distributions (which include one or more suites of popular desktop applications) are available on CD. Outreach programs are distributing CD-Roms of open source applications. Both proprietary and open source software benefit from regular patches and updates downloaded off the internet. Linux and other open source operating systems can be completely downloaded from the internet.

Are open source desktop applications compatible with common proprietary applications? In some cases. StarOffice and OpenOffice.org generally do support MS Office formats; for example, you can open an MS Word document in OpenOffice.org Writer (though not vice versa). However, there may be discrepancies on advanced features and formatting. In practice, this means that simple documents are easily exchanged by MS Office and OpenOffice.org, but the exchange of more documents using advanced formatting and layout can be a problem.

Choosing the Right Software

Whether you are working with open source or commercial tools, picking the right software can be a difficult process. Often, we don't discover that we have made a software choice until it is too late – when we are already using the software in our organization. Also, it is easy to be swayed by an attractive new feature or compelling promotional language on a web site, even if the software in question has certain deficiencies.

The best way to avoid these pitfalls is to invest in a thorough and thoughtful review process before choosing a new piece of software. The simple **3 step method for open source decision making** outlined below is designed to guide organizations through exactly this kind of process. Considering both organizational needs and technical issues, this method can be used by any one with basic information technology and organizational planning skills.

This process works well for all kinds of software – server, desktop, web applications. However, the degree of detail and analysis required will vary depending on the situation and type of software being reviewed. For example, the process of selecting a file server platform to support a single work group can be done very quickly and with very little testing. In contrast, choosing a new word processing application or e-mail client for hundreds of different people in multiple locations will require a more rigorous process with extensive testing. Also, projects that require software customization – especially web application projects – require additional planning related to the programming and development process.

Step One: Define your needs and constraints

The first step in the process is to clearly define your needs. This should include both the overall needs of your organization as well as the needs of individual users. Specific issues to consider include:

- **Organizational needs** – What problem are you trying to solve? Why is your organization seeking new software?

- **User needs** – What do individual users need to be able to do with the software? Are there particular things users have already asked for?

- **Features** – What are the actual features that must be provided by the software? How important are each of these features?

- **Language** – What languages does the software need to accommodate?

At the same time, it is also important to consider the constraints that your organization is under in considering software. Issues to consider include:

- **Budget** – While the software may be free, there will definitely be training and integration costs. How much do you have to spend?

- **Timeframe** – How quickly do you need to implement the software? Does it need to be something that can be up and running tomorrow? Or is there time for customization and configuration work?

- **Compatibility** – Are there legacy systems that the new software must work with? Does it need to run on a particular platform?

- **Skills** – What skills do your existing information technology staff or volunteers have? What skills do end users have? How adaptable are people to new software?

All of these factors will play a major role in the process of identifying your software options and making a final software selection. Given this, is important to write down information about all of these factors.

Step Two: Identify your options

The next step in the process is to come up with a short list of three to five software packages that are likely to meet your needs. This is basically a process of reading through information about various software packages and comparing them against the needs and constraints you listed in the previous phase. There are a number of places that you can look for open source packages to review:

- **Recommendations** – Ask people you know what packages they have used and liked in the past. These people could be from other NGOs or from organizations that provide technical support to NGOs.

- **This guide** – The list of additional resources which accompanies this guide includes a list of mature open source software options in a number of categories. This is a good starting point.

- **Reviews and directories** – There are a number of good open source directories and review web sites. Good places to start include OSDir.com and OpenSourceCMS.com.

- **Software package sites** – Most open source software packages have their own web site. These sites usually contain promotional information and documentation that will help with your review.

Using directories, reviews and software package sites, you should be able to determine which packages are able to meet your basic requirements. When you find a package that seems to fit the bill, you should compare it against the detailed needs and constraints list using the **Software Package Review Worksheet**. If a package meets most or all of your needs and constraints, you should add it to the short list for detailed review.

Step Three: Undertake a detailed review

Once you have identified your options, you are ready for the final step – reviewing and choosing a software package from your short list. At this stage, all of the packages that you are reviewing should be generally suitable for your task. The aim of this section is to assess which of the possible options will be best for your organization. This assessment can be done by rating each package against the following criteria:

- **Quality** – How well do the features you need seem to work? Do you like how they have been implemented?

- **Ease of use** – Is the process of using the software intuitive and obvious given the skills of the people who will be using the software? Or is there a steep learning curve?

- **Ease of migration** – If moving from another software package, how hard is the migration process? Is it likely that users will have a difficult time adapting?

- **Stability** – Does the software crash often? Is a lot of effort required to maintain it and keep it running?

- **Compatibility** – Does the software use file formats and communications protocols that are based on widely accepted open standards? Is it compatible with other systems you are using?

- **Flexibility** – How hard is it to customize and adapt the software to your organization's needs? Will the software grow with your needs? Is it scalable?

- **User response** – When given a chance to test the software, how did users respond? Were they able to figure it out? Were they excited about the way the software worked?

- **Buy-in** – Is there broad support for a particular package within your organization? Are there any major detractors? Active support or resistance for a package can have a major impact on successful implementation.

- **Wide use** – Is there evidence that others are using this software package? Does the popularity of the package seem to be increasing or declining?

- **Support community** – Is there an active online support community? Are there recent postings to the support mailing list? If you post a question, does someone from the community get back to you with a helpful response?

In ranking software against these criteria, hands-on testing is the key. Each piece of software should be installed and tested for quality, stability and compatibility. A group of key users should also be given the chance to try out the software in order to assess factors such as ease of use, ease of migration and user response. Information about usage and support can be gathered by looking at the software package's web site. If the support forums on the site are not active, it is unlikely that the software is widely used or that support will be available.

In terms of ranking, each software package on the short list should be rated against each of the criteria above. A score from one (insufficient) to five (excellent) should be given for each criteria. Ideally a short note on the rationale for each score should also be included. This information can be collected using the **Software Package Review Worksheet**. Scores can be compared using the **Software Comparison Worksheet**.

Making a decision

Once you have completed the 3 step method, you should have a score for each of the software packages on your short list. This score should be a good indicator of which package is best for you. If a package scores very low, it is likely that it will cause problems once implemented and should not be used. This said, you should also use your intuition. If two packages are close in score, your gut feeling about the "right" package is probably more important than the actual numbers.

Looking for a second opinion? *Check out David Wheeler's "How to Evaluate Open Source Software / Free Software (OSS/FS) Programs" document. Wheeler offers a methodology similar to the one offered here, but more from the perspective of a hands-on technical person. The document can be found at: http://www.dwheeler.com/oss_fs_eval.html*

Beyond Decision Making: Costing, Migration and Training

Will it save you money?

One of the big questions with any software decision is "will it save me money"? Answering this question properly is best done using a "total cost of ownership" (TCO) approach. This means considering all of the different costs that will be incurred over the lifetime of a particular piece of technology – hardware, software, maintenance, training, programming, testing, upgrades. Without all of this information, it is impossible to really know which software solutions are going to be the most cost effective.

While there are great debates on the topic, there is a great deal of evidence that mature open source applications offer a lower total cost of ownership than their commercial counterparts. In an article entitled "Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!", David Wheeler lists the main reasons why open source comes out cheaper:

- **Open source costs less to initially acquire** because there are no license fees;
- **Upgrade and maintenance costs are typically far less** due to improved stability and security;
- **Open source software can often use older hardware** more efficiently than proprietary systems, yielding smaller hardware costs and sometimes eliminating the need for new hardware;
- **Experience shows that open source is cheaper** especially in server environments, with many case studies now demonstrating lower TCO for open source.

Wheeler's article also includes many detailed examples and links that show how open source has saved money in particular circumstances. See: http://www.dwheeler.com/oss_fs_why.html for these details.

For more information on calculating total cost of ownership, you may want to visit TechSoup (see: http://www.techsoup.org/howto/articlepage.cfm?ArticleId=295) or the Council on School Networking (see: http://classroomtco.cosn.org/gartner_intro.html)

How will we make the switch?

If you are planning to use open source software to replace an existing system, you will need to deal with the question of "migration". Migrating from one platform to another should be handled using a careful and phased approach. The European Union has published a document entitled the "IDA Open Source Migration Guidelines" that provides detailed suggestions on how to approach migration. The document starts with the following recommendations:

- Before starting have a clear understanding of the reasons to migrate;
- Ensure that there is active support for the change from information technology staff and users;
- Make sure that there is a champion for change - the higher up in the organization the better;
- Build up expertise and relationships with the open source movement;
- Start with non critical systems;
- Ensure that each step in the migration is manageable.

The IDA guide is an excellent place to turn for information before starting the migration process. It includes information about leading open source applications in a variety of categories as well as detailed scenarios describing migrations from common platforms (e.g. an all Windows desktop and server environment). The guide is online at: http://europa.eu.int/ISPO/ida/export/files/en/1618.pdf

How should we approach training?

The other issue that needs to be seriously considered is training. Getting the most out of any technology requires investment in both formal and informal learning for users. Once the migration and testing phases are done, consider:

- **Formal training workshops** for all of people who will be using the software you have put in place. This is especially important for non-technical users who may not be used to learning new tools on their own.

- **Informal peer learning sessions and networks** that encourage users to help each other. Many people learn most effectively from knowledgeable colleagues than they do from a "teacher". Consider ways that users can help each other learn outside of the classroom setting.

- **Printed "cheat sheets"** that provide basic information about how the software works and can be useful in the context of your organization. This approach is very useful if you are using open source to replace an existing tool as the cheat sheets can highlight and explain features that have changed.

- **A list of links to online support resources** that relate to the software you have put in place. Each link should have a short description explaining the kind of help it can provide.

- **Information about the open source community** that supports the software you have chosen. At a minimum, this should include web site links and mailing list sign up instructions. This information will be very valuable for system administrators and developers in your organization.

The exact mix of approaches that make sense for your organization will depend on your circumstances. Installing a new office suite on everyone's desktop probably will require some kind of training.

Appendix A – Annotated Worksheets

Software Needs Worksheet

The following worksheet is meant to capture information about software needs. It is the first document that should be completed during the assessment process.

Software Needs Worksheet		
Organization name	*<Name of organization>*	
Contact person	*<Person responsible for software review process.>*	
Organizational needs	*<What are the problems that the software needs to solve? Why is the organization looking for new software?>*	
User needs	*<What do individual users need to be able to do with the software? Are there particular things they users have already asked for?>*	
Language	*<What languages does the software need to accommodate?>*	
Feature	***Description***	***Priority***
<Name of feature>	*<Brief description of what the feature must do and why it is needed. Add more rows if needed.>*	*<Must have, should have, nice to have>*
Constraints	***Description***	
Budget	*<How much budget is available for this software?>*	
Timeline	*<How quickly do you need implement the software? Does it need to be something that can be up and running tomorrow? Or is their time for customization and configuration work?>*	
Compatibility	*<Are there legacy systems that the new software must work with? Does it need to run on a particular platform?>*	
Skills	*<What skills do your existing information technology staff or volunteers have? What skills do end users have? How adaptable are people to new software?>*	
Other notes		

Software Package Review Worksheet

The following worksheet is used for assessing individual software packages. Complete one of these worksheets for each software package on your short list.

Software Package Review Worksheet			
Name	*<Name of package>*	Website	*http://www.softwaresite.org*
Description	*<Basic information about what the package does and who has created it>*		
Language	*<What languages does the software need to accommodate>*		
Feature	***Yes***	***No***	***Notes***
<Name of feature>			
Criteria	***Rating and notes – 1 = insufficient / 5 = excellent***		
Quality	*<How well do the features you need seem to work? Do you like how they have been implemented?>*		
Ease of use	*<Is the process of using the software intuitive and obvious given the skills of the people who will be using the software? Or is there a steep learning curve?>*		
Ease of migration	*<If moving from another software package, how hard is the migration process? Is it likely that users will have a difficult time adapting?>*		
Stability	*<Does the software crash often? Is a lot of effort required to maintain it and keep it running?>*		
Compatibility	*<Does the software use file formats and communications protocols that are based on widely accepted open standards? Is it compatible with other systems you are using?>*		
Flexibility	*<How hard is it to customize and adapt the software to your organizations needs? Will the software grow with your needs? Is it scalable?>*		
User response	*<When given a chance to test the software, how did users respond? Were they able to figure it out? Were they excited about the way the software worked?>*		
Buy in	*<Is there broad support for a particular package within your organization? Are there any major detractors. Active support or resistance for a package can have a major impact on successful implementation.>*		
Wide use	*<Is there evidence that others using this software package? Does the popularity of the package seem to be increasing or declining?>*		
Support community	*<Is there an active online support community? Are there recent postings to the support mailing list? If you post a question, does someone from the community get back to you with a helpful response?>*		

Software Comparison Worksheet

The following worksheet is used for final step of the process – comparing the packages on your short list. The information at the top of the page can be copied from the Software Needs Worksheet. The numerical scores can be copied from the Software Package Review Worksheets.

Software Comparison Worksheet			
Organization name	*<Name of organization>*		
Contact person	*<Person responsible for software review process.>*		
Organizational needs	*<What are the problems that the software needs to solve? Why is the organization looking for new software?>*		
User needs	*<What do individual users need to be able to do with the software? Are there particular things they users have already asked for?>*		
Language	*<What languages does the software need to accommodate?>*		
Criteria	**<package #1>**	**<package #2>**	**<package #3>**
Quality	*<numerical score>*	*<numerical score>*	*<numerical score>*
Ease of use			
Ease of migration			
Stability			
Compatibility			
Flexibility			
User response			
Buy in			
Wide use			
Support community			
Total score	***<total>***	***<total>***	***<total>***
Notes			

Appendix B – Useful Online Resources

See the ItrainOnline section on open source for an updated list of resources:
http://www.itrainonline.org/itrainonline/english/opensource.shtml

Articles on open source for civil society and development organizations

Open Source in Africa: Towards informed decision-making
Martin Bruggink, International Institute for Communication and Development (IICD)
http://www.ftpiicd.org/files/research/briefs/Brief7.pdf
Last accessed: Nov 29 2003
"This Brief summarises research carried out in Tanzania, Uganda and Burkina Faso and other countries, asking where, how, and why open source software is being used in Africa. Researcher Martin Bruggink highlights the opportunities as well as the risks facing IT managers in African institutions who are thinking of making the switch to open source."

Open Source and the voluntary sector
Openflows, Networks Ltd.
http://www.openflows.org/article.pl?sid=03/05/29/1349220
Last accessed: Nov 29 2003
Topics covered in this report include Advantages of Open Source Software, Case Studies of NGOs Using OSS, Open Source Packages and Support Using Open Source Software. While the report has a Canadian focus it will also be useful to readers in other countries.

Realizing the promise of open source in the non-profit sector
Jonathan Peizer, Open Society Institute
http://www.uploads.nten.org/gems/RealizingthePromiseofOpenSou.pdf
Last accessed: Nov 29 2003
In this report, Jonathan Peizer of the Open Society Institute argues that while open source technology is a viable option for NGOs, it is not a "magic bullet", and needs to be approached in a more strategic and realistic context by civil society actors.

How open source can open doors for nonprofits
Jamie McClelland and Reuben Silvers
http://www.techsoup.org/howto/articlepage.cfm?ArticleId=426
This article by Jamie McClelland and Reuben Silvers argues that the natural connection between nonprofits and the open source software movement has not yet been fully realized, and that "by developing this connection nonprofits will not only benefit from the advantages of open source software but will be able to uniquely contribute to the development of new and richer open source patterns to the benefit of both nonprofits and the greater open source software movement."

Open source is on the map
Dan McQuillan
http://www.lasa.org.uk/cgi-bin/publisher/display.cgi?1427-10103-12611+computanews
Last accessed: Nov 29 2003
Open source software is making inroads into the voluntary sector. Dan McQuillan of Lasa's Information Systems Team and the Multikulti project highlights the potential, the pitfalls and gives some proposals for the future.

Licence fees and GDP per capita: The case for open source in developing countries
Rishab Aiyer Ghosh
http://firstmonday.org/issues/issue8_12/ghosh/index.html
Last accessed: Dec 2 2003
There is a strong case for free software (also known as open source or libre software) being deployed widely in developing countries. As argued in this note, the open source development community provides an environment of intensive interactive skills development at little explicit cost, which is particularly useful for local development of skills, especially in economically disadvantaged regions. Further, this note argues that the controversy over total costs of ownership (TCO) of free vs. proprietary software is not applicable to developing countries and other regions with low labour costs, where the TCO advantage lies with open source, and the share of license fees in TCO is much higher than in high labour cost countries. The note

concludes with a table comparing license fees for proprietary software against GDP per capita for 176 countries.

General articles and resources on open source

How to evaluate open source software / free software (OSS/FS) programs
David A. Wheeler
http://www.dwheeler.com/oss_fs_eval.html
Last accessed: Nov 25 2003
Describes a general process for evaluating programs, with specific information on how to evaluate open source programs.

Why open source software / free software (OSS/FS)? Look at the numbers!
David A. Wheeler
http://www.dwheeler.com/oss_fs_why.html
Last accessed: Nov 25 2003
Provides quantitative data to prove that, in many cases, using open source software / free software is a reasonable or even superior approach to using their proprietary competition according to various measures. This paper's goal is to show that you should consider using OSS/FS when acquiring software.

The IDA open source migration guidelines
European Commission: Interchange of Data between Administrations
http://europa.eu.int/ISPO/ida/export/files/en/1618.pdf
Last accessed: Nov 25 2003
Detailed guidelines to help administrators decide whether a migration to open source should be undertaken and to describe in broad technical terms how such a migration could be carried out.

Windows/Linux comparison chart
Red Hat India
http://www.in.redhat.com/AppComparisonList.php3
Last accessed: Nov 25 2003
List of Linux applications in comparison to their Windows equivalents.

CoSN/Gartner TCO Tool & Case Studies
http://classroomtco.cosn.org/gartner_intro.html
Last accessed: Dec 25 2003
Online tools for calculating total cost of ownership in the school environment.

Examples of NGOs using open source

Greenpeace Southeast Asia Moves to Free Software
Fred Noronha, Linux Journal
http://www.linuxjournal.com/article.php?sid=6598
Last accessed: Nov 27 2003
"Globally, Greenpeace runs an estimated 90+% of their servers on GNU/Linux. But so far, Manila is the only Greenpeace office to fully deploy GNU/Linux on the majority of desktops. Other smaller GP offices are planning to migrate in the coming months."

Jhai Foundation
Jhai Foundation
http://www.jhai.org/jhai_remoteIT.html
Last accessed: Nov 27 2003
The Jhai Foundation is localizing the Linux-based KDE Graphical Desktop and productivity resources, allowing for communications, word processing, and simple spreadsheets, all in the Lao language.

Human rights groups using Martus
Martus.org
http://www.martus.org/resources/case_studies.shtml
Last accessed: Nov 27 2003

Martus is a software tool that allows users to document incidents of human rights abuse by creating bulletins and uploading them to servers located around the world. This article includes NGO case studies from Sri Lanka, Philippines, Guatemala, and Russia.

Where to look for open source software

FSF/UNESCO Free Software Directory
Free Software Foundation (FSF) and United Nations Education, Scientific and Cultural Organization (UNESCO).
http://www.gnu.org/directory/
Last accessed: Nov 25 2003
Huge list of categorized free software for Linux and other free operating systems.

OSdir.com
O'Reilly
http://osdir.com/
Last accessed: Nov 25 2003
Directory of stable open source applications for various platforms and functions.

Freshmeat
Open Source Development Network, Inc.
http://freshmeat.net/
Last accessed: Nov 25 2003
Lists thousands of open source software projects covering a full range of functions, environments and stages of maturity.

SourceForge
Open Source Development Network, Inc.
http://sourceforge.net/
Last accessed: Nov 25 2003
Hosts thousands of open source software projects covering a full range of functions, environments and stages of maturity.

David A. Wheeler's Generally Recognized as Mature (GRAM) OSS/FS programs
David A. Wheeler
http://www.dwheeler.com/gram.html
Last accessed: Nov 25 2003
A categorized short list of the most well established open source applications. Scroll down to "4. GRAM List".

Appendix C – Glossary

Adware

Adware or advertising-supported software is any software application in which advertisements are displayed while the program is running. These applications include additional code that displays the ads in pop-up windows or through a bar that appears on a computer screen.
Source: **Wikipedia:** http://en2.wikipedia.org/wiki/Adware

Bugs

An unwanted and unintended property of a program or piece of hardware, especially one that causes it to malfunction. E.g. "There's a bug in the editor: it writes things out backward". The identification and removal of bugs in a program is called "debugging".
Source: **Free Online Dictionary of Computing:** http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?query=bug

Civil society organizations (CSOs)

Voluntary associations, organizations, movements and networks outside the state and the private sector.
Source: **International Institute for Sustainable Development Word Watch Glossary:** http://www.iisd.org/didigest/glossary.htm#C

Commercial software

Commercial software is software developed by businesses which aim to make money from its use. Most commercial software is proprietary, but there is commercial free software, and there is non-commercial non-free software.
Source: **GNU.org: Categories of Free and Non-Free Software:** http://www.gnu.org/philosophy/categories.html

Creative Commons licenses

The Creative Commons licenses are a set of easy-to-use copyright licenses which are free for public use. They promote the free use and distribution of creative content such as literature, web sites, and music, while retaining the author's copyright.

CSOs

See **Civil Society Organization**.

Demo software

Commercial software that is available for free download and use but with limitations, e.g. certain features are disabled, or the software works only for a 30-day period.

FLOSS

Acronym for Free (Libre) Open Source Software.

FOSS

Acronym for Free/Open Source Software.

Free software

Free software is defined by the users' prescribed freedom to run, copy, distribute, study, change and improve the software. By definition free software is open source.
Source: **GNU.org:** http://www.gnu.org/philosophy/free-sw.html

Freeware

The term "freeware" has no clear accepted definition, but it is commonly used for packages which permit redistribution but not modification (and their source code is not available).
Source: **GNU.org:** http://www.gnu.org/philosophy/categories.html

General Public License (GPL)

General Public License: a specific set of distribution terms for free software that specifically does not let redistributors add any additional restrictions when they redistribute or modify the software. This means that every copy of the software, even if it has been modified, must be free software.
Source: **GNU.org - GNU General Public License:** http://www.gnu.org/copyleft/gpl.html

GPL See **General Public License**.

Linux A trademark name for an open source version of the UNIX operating system. A free operating system used extensively for running servers and increasingly desktop PCs.

NGO Non-governmental organization.

Open source software Software released under a license which gives users the right to both use the software and modify the underlying source code.

Operating system The software which handles basic computer functionality such as the user interface, hardware, and file storage. Popular operating systems include Microsoft Windows, Macintosh operating system and Linux.

PC Personal computer.

Proprietary software Proprietary means that some individual or company holds the exclusive copyright on a piece of software, at the same time denying other people access to the software's source code and the right to copy, modify and study the software.
Source: **Wikipedia:** http://en.wikipedia.org/wiki/Proprietary_software

Public domain The term "public domain" is used to describe publications, software, and other resources which are not protected by copyrights or patents.

Semi-free software Semi-free software is software that is not free, but comes with permission for individuals to use, copy, distribute, and modify (including distribution of modified versions) for non-profit purposes.
Source: **"GNU.org:** http://www.gnu.org/philosophy/categories.html

Shareware Shareware is software which comes with permission for people to redistribute copies, but says that anyone who continues to use a copy is required to pay a license fee.
Source: **"GNU.org:** http://www.gnu.org/philosophy/categories.html

Software license A software license sets out the terms under which the software may be used, and serves as an agreement between the producer and the users of the programme.

Source code The form in which a computer program is written in a programming language such as C++ and Java.

Spyware Spyware is computer software that aids in gathering information about a person or organization without their knowledge. The most common use of spyware is to gather information about the user and relay it to advertisers or other interested parties.
Source: **Wikipedia:** http://en.wikipedia.org/wiki/Spyware

TCO See **Total Cost of Ownership**.

Total cost of ownership (TCO) The sum of all expenses directly related to the ownership and use of a product over a give period of time, usually a few years. For software, these expenses can include license fees, upgrade expenses, required hardware purchases, staff training, migration time, changes in staff efficiency, vendor management and support fees.

Usability

Usability refers to how easily and efficiently a programme allows users to carry out their required tasks. Bad usability can result in user frustration, wasted time and mistakes.

Vendor lock-in

Vendor lock-in is a situation in which a customer is dependent on a vendor for products and services and cannot move to another vendor without substantial costs. It is often used in the computer industry to denote the lack of compatibility between different systems which intentionally or unintentionally forces a customer to continue to use products and services from a particular vendor.
Source: **Wikipedia:** http://en.wikipedia.org/wiki/Vendor_lock-in

[i] There are actually many different licenses that can be used to create open source software. All that is required is that a license meet the criteria of the "open source definition" (see: http://www.opensource.org/docs/definition_plain.php). A list of licenses meeting these criteria can be found online (see: http://www.opensource.org/licenses/).

[ii] Bruggink, Martin "Open Source in Africa: Towards Informed Decision-Making" International Institute for Communication and Development (IICD). (August 2003).
http://www.ftpiicd.org/files/research/briefs/Brief7.pdf (Nov 29, 2003)

[iii] Noronha, Fred. "Greenpeace Southeast Asia Moves to Free Software" *Linux Journal*.
http://www.linuxjournal.com/article.php?sid=6598 (Nov 27, 2003)

[iv] "Open Source and the Voluntary Sector" Openflows, Networks ltd.
http://www.openflows.org/article.pl?sid=03/05/29/1349220 (Nov 27, 2003)

[v] See endnote I.

[vi] Noronha, Fred. "Greenpeace Southeast Asia Moves to Free Software" *Linux Journal*.
http://www.linuxjournal.com/article.php?sid=6598 (Nov 27, 2003)

[vii] "Indian Linux Project – Language Teams." http://www.indlinux.org/lang/ (Dec 1, 2003)

[viii] Wheeler, David A. "Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!" September 8, 2003.
http://www.dwheeler.com/oss_fs_why.html#reliability(Dec 1, 2003)

[ix] "A Brief History of GIMP" http://www.gimp.org/~sjburges/gimp-history.html (Dec 1, 2003)

[x] Bowen, Rich. "Introduction to the Apache Server"
http://apache.rcbowen.com/ApacheServer.html#Introduction_What_is_Apache (Dec 1, 2003)