



Comparing Them Side by Side:

Open Source vs. Vendor-Provided Software

by Jeff Walpole, Phase2 Technology www.phase2technology.com

Imagine you have just been put in charge of selecting a software package to manage your organization's website, or information about your constituents. Your budget is tight, the deadline is looming and the options are dizzying. You know that the sales representatives for any number of commercial vendors would be willing to get you set up quickly. You have also heard that open source applications can provide a solution cheaply—or for free. But can you afford the time it will take to get to know the open source options? Can you afford the cost of licenses?

Either way, you have to figure out how to compare open source applications to vendors' commercial offerings. You also have to explain to your boss what the risks, benefits and implications of each option are—and everyone you talk to has a different take. Worst of all, you don't even know where to start.

Let's start from the beginning and get a sense of the important considerations.

Considering Vendor-Provided Software

We're all familiar with the concept of software provided by a vendor. It's hard to generalize about these packages—they fit into a huge range of categories.

The software might be a one-size-fits-all package that you can easily install on a desktop, like Microsoft Word, or a powerful, configurable system that takes time and care to roll out across an organization, like Blackbaud's Raiser's Edge. You might rent the system by the month and use it over the Internet in a Software-as-a-Service Model, like eTapestry. It might be highly configurable to your needs, like Salesforce. It

might cost nothing to use, like Google Apps, or cost hundreds of thousands of dollars per year. It might be solely supported by the vendor, or the vendor might be only one piece in a broad ecosystem of providers, consultants and implementers that can help you get up and running.

The common bond among all these software packages is that they're distributed by a vendor organization. You contact the vendor (or a distributor approved by them) for the right to use the software package, which often requires a license fee. Vendors have staff members charged with distributing the software, and who can often help you understand what the software does, support you in using it and even partner with you in strategic decisions.

Considering Open Source Software

In contrast, open source software is typically developed, marketed and distributed by a loosely organized community of individuals. The term "open source" means that the source code itself, the instructions that cause the application to do what it does, can be easily viewed, modified or downloaded by anyone with technical expertise.

Open source software is sometimes called Free software or FOSS (Free and Open Source Software), because of four tenets of freedom that are a core part of the philosophy of the open source movement. First, you are free to run these software packages for any purpose—you generally don't pay anything to acquire them. Second, the source code is free—you can see the code and understand how it works. Third, you are free to copy and redistribute the package to anyone you want. And finally, you are free to modify the software however you like, and to release those modifications.



River Network

Connecting People, Saving Rivers

CONTENTS

1 Open Source vs. Vendor-Provided Software
by Jeff Walpole

3 From the President

9 How to Evaluate Open Source Software
by David A. Wheeler

14 Exploring the Variety of Open Source Tools
by Ryan Carter

20 Opening Doors to Open Source
by Stephen Twelker

22 Open Source Mapping Solutions
by Thomas King

27 A Short List of Mapping Tools
& Geospatial Data Sources

28 Considering the Costs
by Waverly de Bruijn

30 Glossary of Terms

32 Voices from the Field

34 Resources & References

35 Partner Pitch

River Voices is a forum for information exchange among river and watershed groups across the country. River Network welcomes your comments and suggestions. River Network grants permission and encourages sharing and reprinting of information from *River Voices*, unless the material is marked as copyrighted. Please credit River Network when you reprint articles and send the editor a copy. Additional copies and back issues are available from our national office.

Editor: Julie Noble, Katherine Luscher

Editorial Assistance: Jacob Cohen, Waverly de Bruijn, Kevin Kasowski

Design & Layout: Greer Graphics

NATIONAL OFFICE

520 SW Sixth Avenue, Suite 1130 • Portland, OR 97204-1511

503/241-3506 • fax: 503/241-9256

info@rivernetwork.org • www.rivernetwork.org

River Network is a national, nonprofit organization whose mission is to help people understand, protect and restore rivers and their watersheds.

RIVER NETWORK BOARD OF TRUSTEES

Todd Ambs
Catherine Armington
Suzi Wilkins Berl
Rob R. Buirgy
Kimberly N. Charles
Mary Ann Dickinson
Barbara J. Horn

Dave Katz
Ezra Milchman
Paul Paryski
Marc Taylor, *Chair*
Balijit Wadhwa
James R. Wheaton
Robert L. Zimmerman, Jr.

RIVER NETWORK STAFF

Matthew Burke
Ryan Carter
Jacob Cohen
Waverly de Bruijn
Steve Dickens
Dawn DiFuria
Merritt Frey
Bevan Griffiths-Sattenspiel
Jean A. Hamilla
Kevin Kasowski
Gayle Killam

Katherine Luscher
Ezra Milchman
Deb Merchant
Julie Noble
Mary Ellen Olcese
J.R. Ralston
Susan Schwartz
Diana Toledo
Stephen Twelker
Wendy Wilson

From The President

W

hen River Network was first incorporated in 1988, there was a relatively small and scattered community of watershed associations and river protection groups in this country.

These organizations had to tackle environmental challenges largely in isolation, created every tool needed from scratch, and learned mostly through trial and error.

By 1994 the size of our circle had grown to roughly 1,800 freshwater protection groups operating in almost every state. Successes were mounting as our local Partners dismantled a series of harmful dams, won dozens of lawsuits that penalized and controlled industrial polluters, created hundreds of regulations that gave teeth to the Clean Water Act, and protected thousands of new river miles through Wild & Scenic River designation.

Action was taking place all around us, and then in 1998 we added the catalyst of interaction when we convened the inaugural River Rally. By this time, there were over 3,100 river conservation groups, and River Rally allowed many of them to come together for the first time. Groups that had been operating in physical isolation were able to now share information and resources face to face, learn from one another and mobilize coordinated action. Today we stand strong at over 6,000 entities that have missions dedicated to conserving and protecting our nation's rivers and watersheds.

Looking forward, should we continue along this continuum to foster a true grassroots water movement in America? To me this means we move from a community of independent groups to a community of interdependent groups. We build consensus on those key issues which hold the majority of River Network Partners back from mission achievement and demand collective action regionally or nationally for success.

In the article by Jeff Walpole about the open source software movement, he explains that what makes a movement are shared goals and interest, community rules and practices, and the ability to execute as a team. As River Network commences a process to build our next strategic plan bottom-up to represent the interests and needs of the grassroots, we will test this potential in our own community.

I hope you'll join us at the 10th Annual River Rally in Baltimore from May 29th to June 1st to learn and be part of our strategic planning. Working together, I believe we can create a movement capable of saving freshwater resources forever.

In friendship,

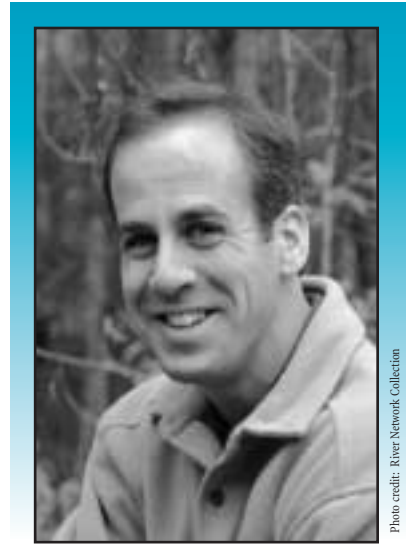


Photo credit: River Network Collection

Open Source vs. Vendor-Provided Software, cont.

cont. from page 1

Community is the key concept to understand how open source software is created—this is an organized, albeit distributed group of people with three main things in common: shared goals and interest, community rules and practices and the ability to execute as a team and create software.

It can be a challenge for institutions not accustomed to the concept to accept open source as a viable approach for developing quality software. But those perceptions are changing. Businesses and non-profits of all sizes from the Fortune 1000 to the federal government have adopted open source software packages for many different purposes. For instance, the Linux operating system and Apache Web servers power much of the Internet. The Firefox Web browser is gaining substantial market share. Content management systems like Joomla, Drupal and Plone are widely used by nonprofits, while Constituent Relationship Management systems like SugarCRM and CiviCRM are increasingly viable options. An open source model can result in powerful, secure, useful, industrial-strength software.

In truth, the idea of “vendor-provided” and “open source” solutions are not opposites. A vendor can develop a software package and then release it under an open source license—typically then making money from support or implementation work. Conversely, many firms provide services, support and even packaged implementations for open source software.

How are Vendor-Provided and Open Source Options Similar?

Many of the same decision criteria apply to open source and vendor-provided software. Even software that is free to acquire has a long-term cost to implement and maintain it—often called the Total Cost of Ownership

(TCO). Regardless of what type of software you consider, make sure you weigh the following options:

- ◆ **Feature Set.** It should go without saying that you shouldn’t consider any software package—vendor-provided or open source—if it doesn’t do what you need it to do. This means starting with a detailed understanding of your organization’s needs.
- ◆ **Hardware Costs.** What hardware will you need to purchase and support to effectively run the software?
- ◆ **Implementation Complexity.** Think through the amount of work it will take to get the software package up and running and customized as needed. Software applications always work best when customized least—this is as true for open source as for commercial. Apply the 80/20 rule, and assume if you have more than 20 percent to change, the cost and complexity is going to spike exponentially.
- ◆ **Maintenance Expenses.** Typical maintenance expenses include upgrades, patches, customization modifications and many “care and feeding” type of activities. Hosted commercial solutions may offset these costs, but usually any installed system (commercial or open source) will cost the organization for maintenance.
- ◆ **Training and Support Expenses.** Packages that are more difficult for your end users to learn will cost more money in the long run, as you’ll need to provide more training and more support down the road.
- ◆ **Module/Application Programming Interfaces (API) Availability.** Smart vendors and open source communities

alike provide open and accessible ways to tap into the product and its data, and when appropriate, to extend it. If you don't see these options, ask yourself why. Will your options to extend functionality be hindered by the lack of a clearly defined API or modular architecture? Or worse, will your data be locked into a proprietary structure, or marooned? Ways to access data should be non-negotiable to organizations that don't want to get stuck down the road.

Considering the Strength of the Software Community

A strong community of users and service providers can help effectively support any software product and guide its evolution. This is true of vendor-provided software, but it's a particularly important consideration for open source software. Because many open source packages are supported and improved solely by a community, the health of that community is critical. What should you consider when looking at software communities? And how do those considerations differ for open source and vendor-provided products?

- ◆ **How large is the community?** Any product has a user base—the stronger the base, the more likely that the product will be around in the future and continue to improve. Are there companies that specialize in implementing the software? Are there training resources? How about user groups and events based around the product? With open source, it's also important to look at the number of developers and contributors to the code base. While an extensive community is useful for any product, it's critical for an open source product—open source products need a community in order to thrive and grow.

- ◆ **How frequently are new releases put out?** Success in software comes from momentum and innovation. Products that don't innovate are quickly bypassed by their competitors—or, they become vulnerable to the introduction of an open source alternative. Make sure vendors have a strategy and rationale for software releases. In open source, look for communities that can put out important and well planned releases often. This demonstrates the ability to organize, and may be a strong indicator of the ability to survive.
- ◆ **How is community input factored into product enhancements?** Whether the product is open source or vendor-provided, look for a model that includes community input into the decision making process for new features, patches and bug fixes. With vendor-provided software, paying customers can be a powerful market force that can truly affect the pace and direction. If a number of customers advocate for a particular feature, it can be difficult for vendors to ignore the pressure. On the other hand, open source projects often open up the development plan to a community of users and active developers. When done well, this can help foster interesting, innovative and practical ideas into the software-planning process, and provide a diverse set of eyes for quality assurance. However, vendors can be unresponsive to customer concerns, and one or two people dominate the decision making on some open source products. Make sure you assess community involvement for the particular product you're looking at. How is feedback gathered? How are enhancements planned? If you don't see these things being decided in an organized, coordinated and professional way, run for the hills.

cont. on page 6

cont. from page 5

- ◆ **How advanced is this community?** Not all companies or communities draw the best and brightest with the most technically innovative ideas. Some projects are guided primarily by one or two well-intentioned individuals. Others are full of cutting-edge technologists challenging each other for mutual technical respect. You want to hitch your wagon to a community that has intelligent, motivated and innovative plans the same as you would in selecting a company to work for.
- ◆ **How well does a community support its users?** You cannot overestimate the importance of support. It is vital both to newcomers who need help getting started and to the experienced users who get stuck. Some vendor-provided software packages are supported solely by the company that makes them—in this case, the quality of that support is vital. Some open source products are supported primarily by a core community of developers and users through online messaging, making the helpfulness and responsiveness of that community a key concern. Vendor-provided software typically provides assurance of some kind of support, but this support might be far less robust than that of an active open source project. Further complicating this subject is that many of the more complex software packages, whether open source or vendor-provided, have communities of consultants and firms that can also provide support. But in practice, support can only be evaluated on a product-by-product basis. What are the support options? What do current users say about the quality of the support?

What are the Strengths of Vendor-Provided Software?

There are reasons many organizations pay for vendor-provided software, including:

- ◆ **Ease of Getting Started.** Organizations that don't already have experience with open source projects often find commercial software has a less-daunting road to getting up and running. Vendors often provide sales support and engineering staff trained to make this happen.
- ◆ **Documentation.** Finding up-to-date, well-written and easy to use documentation can be a problem with any software. However, vendor-provided software often comes with better documentation because it is someone's job to write it and keep it up to date for each release or version of the product.
- ◆ **Accountability.** As opposed to a piece of electronic equipment or a car, "bugs" and defects are expected in a software product, and discussed openly as a risk of using the product. While you're generally on your own for any issues with an open source package, commercial vendors typically warranty against software flaws that can impact your organization.
- ◆ **Mature Ease-of-Use.** While it varies by product, in general, paying customers and a team of customer-facing sales people often push vendor-provided software into better user interfaces and ease of use. Open source products are often driven by developers instead of business or marketing folks, which can result in user experience and usability being pushed down the list of priorities.

What are the Strengths of Open Source Software?

Why might you prefer open source software? Let's explore some of the most common benefits.

- ◆ **Free to Try, Buy and Update.** You can download the full version of an open source software package and start exploring it, typically without paying a dime. You can set up a prototype or rollout the full system without any fee for use. And, as long as you're careful with customizations, you can likely upgrade to future versions and get access to new features the community has built for you—without cost. Of course, don't forget that you'll typically need to invest time or money into installing, configuring and updating the package, but with open source you have more choice over what and when.
- ◆ **Ability to Customize and Control.** Developers often focus on providing a flexible data model, extensible and/or modular code base, and APIs (application programming interfaces) designed to give customers the ability to customize and modify for their needs. What's more, while vendor-provided options generally tie your organization to the vendor's vision for the product, or even occasionally commit you to terms that make it difficult to use other options, open source products allow you more control over your own destiny should you want to do something different.
- ◆ **Community-Driven Solutions.** Because open source software is developed by a community with shared needs, there tends to be a greater sense of what is really needed—and what is not. A Darwinian sense of value vetting occurs, and the best ideas get done, and the worst do not. Typically



communication and collaboration on the community's Web site, chat groups, events and conferences drive the most pressing issues directly into the hands of the developers solving problems—cutting out the middle men in marketing, management, sales, etc.

- ◆ **Alignment of Philosophy.** Most non-profits find the concept of open source attractive for cultural reasons. The idea that open source software represents value created without the intention of generating profit is a natural fit. If an organization needs to put time and money into software, they often find it preferable for those resources to support a package that is available to everyone without cost.

Critically Assessing Your Own Need

You can likely see benefits and drawbacks to each type of software. There is no single right or wrong answer for all organizations. Instead, you'll need to decide which strengths and weaknesses matter most for your unique situation.

cont. on page 8

Open Source vs. Vendor-Provided Software, cont.

cont. from page 7

For instance, if you'll need to highly customize a package, it's likely to be more expensive to customize traditional vendor-provided software—if it's even possible. However, if your organization generally needs lots of training and support, traditional vendor-provided solutions are likely to be a better fit.

Start by building a list of your needs, researching and weighing options on more factors than just cost. Conduct thorough due diligence, compare features and, whenever possible, try out the software. Do not allow the relative accessibility of an open source system to tempt you to start using it without having a solid set of requirements to compare against commercial alternatives. Likewise, you should never pay for software without researching its open source alternatives.

With a list of options in hand, plug all the potential benefits and potential pitfalls into a proven software-evaluation model. Your approach to evaluating open source software and commercial alternatives should be roughly the same. Decide which of the considerations in this article are important to you. Rate the packages based on these considerations, the features and functions you need, and on such attributes as usability and maintainability. Ensure that you've included all the costs of implementation and ownership as well as licensing costs.

When it comes down to the wire, open source software and vendor-provided software aren't so different. It's critical to look at the features, attributes and costs for each—and then weigh the unique benefits each model can bring. By choosing to use an open source system, you are tapping into a community of peers and taking more control of your software destiny, but you are also

choosing a course of action that will require a different mindset—and perhaps more work to support—down the road.

In the end, there's no easy answer. Open source software isn't likely to be a free-and-easy cure to all your woes, but it's certainly a viable option worth considering. Vendor-supported solutions don't have a monopoly on solid feature sets or good support, but they may be a faster and easier choice especially for simple needs.

But don't let the choice overwhelm you. The possibilities don't have to be dizzying if you break down your needs and considerations, and compare the options side by side.



Jeff Walpole is the CEO of Phase2 Technology, a web application development and consulting firm based in Washington, DC, with specialized expertise in the open source community. More insights on open source, web application development trends and web technology can be found on the company's staff-written blog at www.agileapproach.com.

Jeff Herron of Beaconfire, Peter Campbell of EarthJustice, and Michelle Murrain of MetaCentric Technology Advising also contributed to this article.

Thank You

River Network thanks Idealware for their continuing support of the river conservation community. This article can be viewed on their website at Idealware.org.

How to Evaluate Open Source Software

The basic steps for evaluating both open source and vendor provided software, are essentially the same. I suggest the following steps:

- ◆ *identify* candidates,
- ◆ *read* existing reviews,
- ◆ briefly *compare* the leading programs' attributes to your needs, and
- ◆ perform an in-depth *analysis* of the top candidates.

Significant technical mastery of software development isn't required for most of the process described here, but there are a few steps which are best performed by someone knowledgeable about software development. The amount of effort you should spend evaluating software is strongly dependent on how complex and important the software is to you. The whole evaluation process might take 5 minutes for a small program, or many months when considering a mammoth change to a major enterprise. The general process is the same; what is different is the amount of rigor (and thus effort) in each step.

Before beginning an evaluation process, it is critical that you have a basic idea of what you need. If you don't, you'll need to first determine what your basic needs are. Usually you will refine your understanding of what your needs are as you evaluate, since you're likely to learn of capabilities you hadn't considered before. Try to be flexible in comparing needs to products, though; a product that meets 80% of your needs may have other advantages that make it better than a product that meets 100% of your originally-posit-ed needs.

1. Identify Candidates

The first step is to find out what your options are. You should use a combination of techniques to make sure you don't miss something important. An obvious way is to ask friends and co-workers, particularly if they also need or have used such a program. If they have experience with it, ask for their critique; this will be useful input for the next step, obtaining reviews.

Look at lists of Open Source Software (OSS) programs, including any list of "generally recognized as mature" (GRAM) or "generally recognized as safe" (GRAS) programs. After all, some OSS products are so well-known that it would be a terrible mistake to not consider them. For example, anyone who needed a web server and failed to at least consider Apache would be making a terrible mistake; Apache is the market leader and is extremely capable. Refer to the resources at the end of this article for examples of GRAM, GRAS, and OSS program lists.

When running internet searches for lists of OSS, there are several different kinds of search systems you should try. Search using specialized sites which track OSS programs and sites which host or include many OSS projects. Also, use a good general-purpose Internet search engine, and search for the kind of product you're looking for. Be sure to avoid search engines with obvious conflicts of interest, e.g., a search engine owned by a maker of one product is unlikely to help you learn about their competitors. Try several variations of what you're searching for. Identify a few key words that would likely be in a description of what you're looking for. Others may not use the same naming conventions you do, so you'll need to try variations.

If all else fails, ask others. Find somewhat similar or related programs, and ask for what you're looking for on their mailing

by David A. Wheeler
www.dwheeler.com

cont. from page 9 lists. You can also hire someone to do a more detailed search.

2. Read Existing Reviews

After you've identified your options, read existing evaluations about the alternatives. It's far more efficient to first learn about a program's strengths and weaknesses from a few reviews than to try to discern that information just from project websites.

The simplest way to find these reviews is to use a search engine and search for an article containing the names of all the candidates you've identified. Also, search for websites that try to cover that market or functional area, and see if they've published reviews. In the process, you may even identify plausible candidates you missed earlier.



It's critical to remember that many evaluations are biased or not particularly relevant to your circumstance. Most magazines are supported by advertising, and they're a lot less likely to bite the hands that feed them. Systems that allow multiple people to comment can be easily biased by someone intent on biasing them. Still, it's worth hearing a few opinions from multiple sources. In particular, evaluations often identify important information about the programs that you might not have noticed otherwise.

An important though indirect "review" of a product is the product's popularity, also known as market share. Generally you should always try to include the most popular products in any evaluation. Products with large market share are likely to be sufficient for many needs and are often easier to support and interoperate. OSS projects are easier to sustain once they have many users; many developers are originally users, so if a small percentage of users become developers, having more users often translates into having more developers. Also, developers do not want their work wasted, so they will want to work with projects perceived to be successful. Conversely, a product rapidly losing market share has a greater risk, because presumably people are leaving it for a reason.

3. Compare the Leading Programs' Attributes to your Needs

Once you've read other reviews and identified the leading OSS contenders, you can begin to briefly examine them to see which best meet your needs. The goal is to winnow down the list of realistic alternatives to a few candidates. Note that you need to do this after reading a few reviews, because the reviews may have

identified some important attributes you might have forgotten or not realized were important. This doesn't need to be a lengthy process; you can often quickly eliminate all but a few candidates. The first step is to find the OSS project's web site. An OSS project's website doesn't just provide a copy of its OSS program; it also provides a wealth of information that you can use to evaluate the program. Project websites typically host a brief description of the project, a Frequently Asked Questions (FAQ) list, project documentation, web links to related/competing projects, mailing lists for developers and users to discuss the program or project, and so on.

Next, you can evaluate the project and its program on a number of important attributes through which the benefits, drawbacks, and risks of using a program can be determined.

Functionality

Does the program do what you want it to do? In particular, you should consider how well it integrates and is compatible with the existing components you have. You should also consider what hardware, operating systems, and related programs it requires and whether they will be acceptable to you (do you have them or are you willing to get them)? Few programs (proprietary or OSS) provide all functionality you would like. It's often possible to decide to do without, to supplement the missing function with some procedure, or use a separate program to supplement the missing functionality. One additional option essentially unique to OSS is that you can have the missing functionality added to the program itself, by changing its code. An organization can add functionality by developing the changes in-house, or by paying someone to add the functionality.



Cost

Strictly speaking, most OSS programs don't cost anything to get, however, the term "free" in the phrase "free software" is based on "freedom" and not on price. OSS programs still cost money to deploy in the real world, because initial licensing costs are a minority of the costs in most software deployments. Thus, when considering costs, you should consider all costs related to deploying a program. This is typically done by computing the total cost of ownership (all costs related to deploying the program over a period of time) or as a return on investment (by comparing the total costs to the total benefits), over a fixed period of time.

Support

Support includes training users on how to use the product, installing the product, and responding to users who have specific problems trying to use a working product. This includes product documentation (user documentation, reference guides, and any other source of information). It also includes any warranty or indemnification you need.

Look at the documentation available for the product, including user manuals, "quick start" material, administrator manuals, reference guides, and so on; do they make

How to Evaluate Open Source Software, cont.

cont. from page 11

sense? Can you understand them? Look for other documentation; many OSS programs have only basic documentation, and you may need to get books elsewhere on them. Take a look at this set of documentation - is it sufficient to use the program the way you want to use it?

Maintenance/Longevity

Few useful programs are completely static. Needs change, new uses are continuously created, and no program of any kind is perfect. It's important that a program is being maintained, and that it will be maintained far into the future. Of course, predicting the future is very difficult. However, if a program is being actively maintained, it's far more likely that the program you choose today will be useful tomorrow.

The OSS project site serves as a focal point collecting the improvements of various users. Thus, the project's Internet presence can give you an indication of how well the program is maintained. Examine the developer mailing list archives—is there evidence they're actively discussing improvements to the software? Are there multiple developers (so that if one is lost, the project will easily continue)? If their version management information is accessible to the public, take a look—are developers regularly making improvements and bug fixes? Look at how bug reports are handled (e.g., are they quickly examined and, if really a problem, fixed? How long does it take?). Ken Krugler suggests the following checklist for community activity: active mailing list, bug tracking system, and public wiki. “If a project has all three of these, then it usually has sufficient structure and commitment

from a community to be viable in at least the near term. A fourth is whether there's a book on the project/component.”

Reliability

A mature program is far more likely to be reliable. The project's web site itself is likely to try to describe the program's maturity; if the project declares that the program is not ready for end-users, they're usually right. The best way to measure reliability is to try it on a “real” work load, as discussed later.

Performance

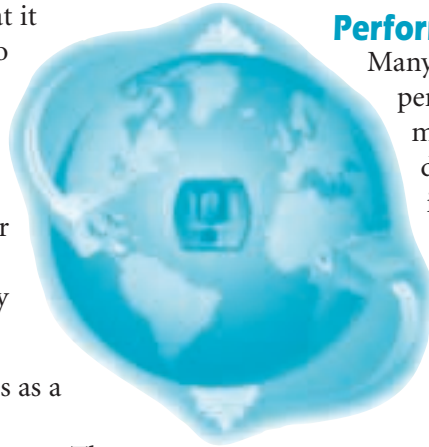
Many project websites include performance data. Project mailing lists may include more detailed performance information. The best way to measure performance is to try it on a “real” work load specific to your circumstance, as discussed later.

Scaleability

Scaleability, in this context, suggests the size of data or problem the program can handle. If you expect the program to be able to handle unusually large datasets, or be able to execute on massively parallel or distributed computers, there should be some evidence that the program has been used that way before.

Useability

Useability measures the quality of the human-machine interface for its intended user. A highly useable program is easier to learn and easier to use. In the end, evaluating useability requires hands-on testing.



Security

Evaluating a product's security is complicated, in part because different uses and different environments often impose different security requirements on the same type of product. One step toward solving this problem is to briefly identify your security requirements. Then, look to see evidence that the product works to meet those, and that it works to counter vulnerabilities in general. Independent evaluations of the software can give you some valuable information. One hint is simply looking at the user's guide - does it discuss how to make and keep the program secure? Does the project have a process for reporting security vulnerabilities? Another hint is to examine developer mailing lists to determine if they discuss security issues and work to keep the program secure.

Flexibility/Customizability

Flexibility and customizability are two highly interrelated attributes. Flexibility measures how well a program can be used to handle unusual circumstances that it wasn't originally designed for. Customizability measures how well you can customize the product to fit into your specific environment.

Interoperability

Nothing exists in a vacuum; you'll need to make sure that your product will work with the other products you use or plan to use.

Legal/ License Issues

Legal issues are another important attribute, and they are primarily defined by a program's license. Thus, you should examine the license requirements for each program you're considering, as well as their implications in your country.

Other Issues

Of course, there may be other issues that are important to you, such as local policies or unusual technical requirements. Clearly you should include those in your evaluation as well.

Perform an In-Depth Analysis of the Top Candidates

After this initial evaluation, you then pick the top contenders, and perform a more in-depth analysis of them. In particular, try them out on representative work loads. The important attributes to consider are the same as in the previous step; you simply spend more effort by actually trying things out instead of quickly reading the available literature. For performance and scalability, set up a representative situation dummy data (and a dummy amount of data) and see how well the program performs.

Obviously, you should try any new program in non-critical situations first, to see how well it works before truly deploying it. If the Open Source Software program had some but not all the functions you need, you should examine what it would take to add those functions. This can be done by paying others or by doing it in-house.



David A. Wheeler is an expert in computer security and has a long history of working with large and high-risk software systems. His books include Software Inspection: An Industry Best Practice (published by IEEE CS Press), Ada 95: The Lovelace Tutorial (published by Springer-Verlag), and Secure Programming for Linux and Unix HOWTO.

For the full version of David Wheeler's paper on OSS software evaluation, visit: www.dwheeler.com/oss_fs_eval.html.

Exploring the Variety of Open Source Tools

by Ryan Carter
River Network

www.rivernetwork.org



Open source software is not new; in fact, if you search the web, you'll find pages and pages of information covering the topic. But, the web can be a web: a puzzling, tricky—if not downright confusing—arrangement of information and facts. To help guide you through the maze of information, we've provided a quick, albeit non-comprehensive look at open source equivalents to popular vendor-provided software.

Purpose:	What it Does:	Vendor-Provided Program:	Open Source Equivalent:
Audio Editor	Software used to edit audio files on a computer. Also called a “sound editor,” audio sections can be deleted and mixed, and the pitch, speed and tempo of the material can be modified. The dynamic range can be compressed to make loud parts softer, and special effects can be added. Most audio editors provide support for a variety of audio formats.	Sony Sound Forge Goldwave	Audacity
Audio/Video Player	A video player is a kind of media player for playing back digital video data from media such as optical discs (for example, DVD or VCD) as well as from files of appropriate formats such as MPEG, AVI, RealVideo and QuickTime. In addition to VCR-like functions such as playing, pausing, stopping, rewinding and forwarding, some common functions include zooming or full screen, audio channel selection, subtitle selection and frame capturing. Many media players also support simple playback of digital audio, some are even specially designed for this purpose.	Windows Media Quick Time DivX	VLC Media Player
CD/DVD Burner	Burn is a colloquial term meaning to write content to a CD, DVD or other recording disc. DVD and CD drives with recording capabilities (sometimes called DVD or CD burners) etch data onto the disks with a laser. CD-Recordable (CD-R) and CD-Rewritable (CD-RW) are the two most common types of drives that can write CDs, either once (in the case of CD-R) or repeatedly (in the case of CD-RW). A DVD-R/DVD-RW drive allows writing to DVD, with a greatly increased storage capacity.	Nero Roxio	CDBurnerXP



Purpose:	What it Does:	Vendor-Provided Program:	Open Source Equivalent:
Customer Relationship Management (CRM)	A CRM entails all aspects of interaction a company or organization has with its customers or members, , whether it be sales or service related.	SAP Seibel SalesForce	CiviCRM SugarCRM
Desktop Publishing Page Layout	Use of special programs running on personal computers to design and print professional-quality documents. These programs assist the user with layout design; the insertion of photographs, sketches or graphics; many various typefaces; and the use of multiple input files so that any individual with a personal computer can produce customized documents to suit specific advertising or marketing needs.	Adobe InDesign Pagemaker	Scribus
Email Scheduling	An email scheduler enables you to define multiple email projects, save and schedule them separately, as well as send reminders.	Microsoft Outlook	Mozilla Thunderbird
File Compression Archiving	The process of transforming information from one representation to another, smaller representation from which the original, or a close approximation to it, can be recovered. The compression and decompression processes are often referred to as encoding and decoding. Data compression has important applications in the areas of data storage and data transmission.	WinZip WinRAR	7-Zip
FTP Client	Short for “File Transfer Protocol,” FTP client allows users to transfer and manipulate files on a remote computer. Commonly used to alter files on a web server or to download files from the web to install.	CuteFTP	WinSCP

cont. on page 16


cont. from page 15



Purpose:	What it Does:	Vendor-Provided Program:	Open Source Equivalent:
Geographic Information System (GIS)	A computer application used to store, view and analyze geographical information, especially maps.	ArcGIS	GRASSGIS Geoserver
Instant Messaging	Abbreviated IM, a communications service that enables you to create a kind of private chat room with another individual in order to communicate in real time over the Internet, analogous to a telephone conversation but using text-based, not voice-based communication.	AIM Windows Live Yahoo	Pidgin
Integrated Office Suite	A suite is a set of usually related programs sold together in a single package. Sometimes called “application suites,” “integrated office suites,” “software suites,” and “bundled software.”	Microsoft Office (Word, Excel, Powerpoint, Access)	OpenOffice (Writer, Calc, Impress, Base)
Operating System	The most important program that runs on a computer. Every general-purpose computer must have an operating system to run other programs. Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk and controlling peripheral devices such as disk drives and printers.	Microsoft Windows MacOS	Ubuntu (Linux)

Purpose:	What it Does:	Vendor-Provided Program:	Open Source Equivalent:
PDF Converter	A program that allows users to convert documents into the PDF format, for ease of viewing and printing. Generally appears as an additional printer in an application's print window, and is used like any other printer. When selected, the converter program converts and outputs ("prints") the current document into a PDF file on the hard disk.	Adobe Acrobat Adobe Distiller	PDFCreator
PDF Viewer	Short for "Portable Document Format," a file format developed by Adobe Systems. PDF captures formatting information from a variety of desktop publishing applications, making it possible to send formatted documents and have them appear on the recipient's monitor or printer as they were intended. To view a file in PDF format, you need a PDF viewer.	Adobe Reader	SumatraPDF
Photo/Image Editing	An image editing application that is specialized for digital cameras. It is used to crop and touch up photos from the camera and organize them into albums and slide shows. Photo editors do not have the myriad of editing filters and features of a full-blown image editor, such as Photoshop or Paint Shop Pro, but they include a basic set of tools for the consumer.	Adobe Photoshop	GIMP Cinepaint

cont. on page 18

cont. from page 17

Purpose:	What it Does:	Vendor-Provided Program:	Open Source Equivalent:
<p>Web-based Content Management System (CMS)</p>	<p>A <i>Content Management System</i> is a bundled or stand-alone application used to create, manage, store and deploy content on Web pages. Web content types can include text, graphics and photos, video or audio, and application code that renders other content or interacts with the visitor. A CMS may also catalog or index content, select or assemble content at runtime, or deliver content to specific visitors in a personalized way or in different languages.</p>	<p>WordPress Vignette</p>	<p>Drupal Joomla Plone Mambo</p>
			
<p>Web Browser</p>	<p>A software application used to locate and display web pages. Both of these are graphical browsers, which means that they can display graphics as well as text. In addition, most modern browsers can present multimedia information, including sound and video, though some formats require plug-ins.</p>	<p>Internet Explorer</p>	<p>Mozilla Firefox</p>

Purpose:	What it Does:	Vendor-Provided Program:	Open Source Equivalent:
WYSIWYG Website Design	<p>Pronounced “wiz-ee-wig,” it refers to the type of user interface that allows the user to see the results of editing as it occurs. In contrast to traditional text editors that require developers to enter descriptive codes, or markup but do not allow for an immediate way to view the results of the markup. A WYSIWYG (what you see is what you get) editor can display the results immediately.</p>	Adobe Dreamweaver	Kompozer
Vector Graphics	<p>Vector graphics refers to software and hardware that use geometrical formulas to represent images. The other method for representing graphical images is through bit maps, in which the image is composed of a pattern of dots. Vector-based images are more flexible than bit maps because they can be resized and stretched, look better on monitors and printers with higher resolution and often require less memory than bit-mapped images do.</p>	Adobe Illustrator	InkScape 
			

Getting Started:

Opening Doors to Open Source

by Stephen Twelker

River Network

www.rivernetwork.org



When deciding whether to use an open source application, the best method is the most direct. In his bestseller *Blink*, Malcolm Gladwell wrote, “We learn by example and by direct experience because there are real limits to the adequacy of verbal instruction.” Expand this to include written instruction, apply it to technology, and we realize that the best way to test open source software is to get your hands on some and work with it directly. But unless you want a big mess on your hands, it pays to do a little planning.



What do you want? I mean, what do you really want?

Do you need to build a website? Graph some data? Create an animated fireplace screensaver? The software is out there, but it's not going to tell you what you need. Without explicitly defining your problem, you're not going to find a solution. Turn your problem into a one-sentence mission statement. Then extract several search terms from your mission statement—you'll need them for your next step.

Research applications in the internet abyss

You have a love/hate relationship with online research. On the one hand, heck, you're online! As of last summer, there were one trillion web pages to explore. On the other hand, you're online, and your poor kid is asking the dog to read a book to him. Skip the Google, save yourself time and enter your key terms into the search engines at Download.com (www.download.com), Sourceforge (<http://sourceforge.net>) and Fresh Meat (<http://freshmeat.net>).

Use your (figurative) crystal ball to project three to five years into the future

Think back five years. What did you tell your friends/spouse/dates that you would be doing in five years? Were you even remotely accurate? When anticipating future technological needs, do you expect to be even in the right galaxy? Still, you have to do it. Don't spend time and money on applications for Windows if you expect to switch all your machines to Macs next year.

Determine your budget, then admit it's insufficient

Some of the best advice I've gotten from a tech consultant was also the hardest to swallow. After coming up with an estimate of your time and financial expense for a software project, double it, then double it again. There's your real estimate. Anyone who has sat down “for an hour, I promise” to reinstall an operating system, only to watch the day fade into twilight after a dozen confusing installation wizards, crashes and reboots understands this.

Choose your application, and get busy

When my wife asked me to build a website for her naturopathic practice, I dove right in. I bought some cheap server space (try out www.thinkhost.com for a green solution), which came preconfigured with the popular LAMP environment (the four components of LAMP—Linux, Apache, MySQL and PHP—are all open source and free, but take a little time and knowhow to install correctly, so I saved a little time and sweat finding a preconfigured server). Next I downloaded the Drupal web content management system (<http://drupal.org>) and spent a little time with the installation manuals. It's not difficult to build web pages and menus in Drupal, so I set up the basic information architecture, resulting in a functional, text-only site.

My wife joined me as we started to make the site look a little cooler. We wanted to incorporate some photos, which meant finding an appropriate module (<http://drupal.org/project/modules>), installing it and uploading our graphics. And since the default theme (the design and graphic layout) wasn't thrilling us, we looked through some freely available themes (<http://drupal.org/project/themes> and <http://themegarden.org/drupal6>), settled upon one we liked, and watched as our previously just-functional site became pretty neat. I made a few desired changes to the theme file (using an open source text editor,



of course), and the site was fully customized.

River Network's web team used Acquia, (<http://acquia.com>), a proprietary package of the Drupal code and selected modules, for our new site (www.rivernetwork.org). The scope of that project was such that we hired a professional developer to design the site (www.jeffreydalton.info). And it shows, we think—as talented as our web team is—we could not have assembled those modules and customized the code necessary to build the site. We are geniuses, yes, but we're not brilliant geniuses.



Open Source Mapping Solutions

by Thomas King

OpenSourcery

www.opensourcery.com

Interactive, online mapping tools have become popular over the past few years largely due to the success of Yahoo! Maps and Google Maps. These user-friendly applications have made integrating mapping data and mapping imagery with your website very easy. In the past, comparable solutions were incredibly expensive and complex because they required proprietary software with licensing fees and they wouldn't work on the basic shared hosting accounts that most nonprofits and river groups use. But now, since these major providers are doing most of the heavy lifting in terms of serving the data, mapping tools have become lightweight and flexible.

This is good news indeed for river conservation groups. Perhaps more than any other nonprofit sector, river conservationists can use maps to tell an interactive story of the value they provide members and donors (not to mention the population at large). Below you will learn some general open source mapping concepts and a few tricks on how to successfully employ mapping tools.

How Mapping Tools Work

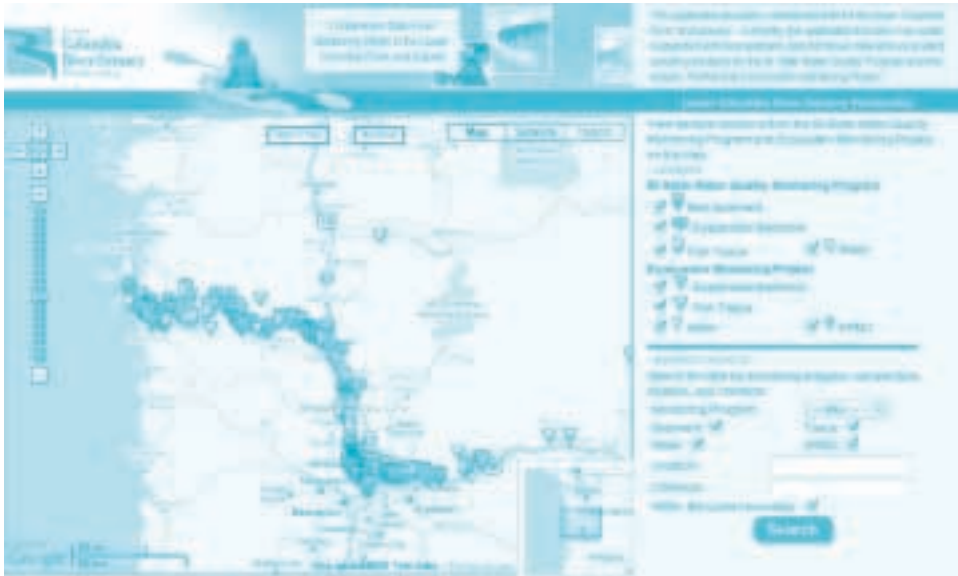
When an organization embeds a Google, Yahoo! or other third-party map on its website, it relies on an application programming interface (API) to serve the required data. In order to retrieve that data, your website must be able to make requests of the API and display the results in the form of an embedded map. Sounds a little like magic, but getting serious value from online maps doesn't have to be daunting.

At the most basic level, your river group can embed on its website a static map that shows your location and a defined surrounding area. With five minutes and an internet connection, you can see how a simple embedded map code snippet is created¹. And if such a map were all your organization required, a developer could copy the code snippet Google Maps provides and embed it on your site with little effort. Said snippet would then send requests to Google Maps' API and—voilà!—a simple embedded map appears on your site.



¹ Visit Google Maps and search for your house or office or favorite café. Above the map you'll find a "Link" button, which leads you to a page where you can customize the size and center point of your map. In the same window you'll find a snippet of code a tech-savvy volunteer or IT manager can paste onto your website.

Clackamas Stewardship Partners (OR) has implemented open source mapping tools. <http://clackamasstewardshippartners.org>



Lower Columbia River Estuary Partnership (OR) utilizes maps on their webpage.

<http://lcrep.org>

Unfortunately, the map described above is limited and, when you want to add new data points, requires manual revision. For more useful maps we recommend integrating with a content management system (CMS) such as Drupal, an open source framework, which can store your group’s water quality data, images, links, etc. on a database. Once configured, your CMS allows users to pull specific data sets from the database and dynamically send them to the embedded map.

Translation: the information your organization has worked so hard to gather can interact with robust maps provided freely by those benevolent behemoths. Check out some examples of what is possible:

- ◆ Display points on a river where your group is active.
- ◆ Upload images to these map points (photos of a cleanup).
- ◆ Attach videos of cleanup work in action.
- ◆ Display water quality data (easily uploaded by volunteers).

- ◆ Generate awareness and involvement in the community.
- ◆ Integrate educational uses with fundraising uses.
- ◆ And, nearly infinite creative relationships yet to be uncovered.

As you can see, a story begins to emerge: a story that can increase awareness of your organization’s good work and, in turn, the support it receives. Take a real-world example: The “Happy River Trust” pinpoints a river’s public access points and displays them on a map. The Trust then layers a map of pollution points over the access points (like a transparency, only it’s not your high school biology teacher fumbling the thing around until nobody knows what’s up or down). Powered by solid visuals, Happy River Trust begins a fundraising campaign based on that “story-in-data.” At the conclusion of the campaign, they close the loop by reporting to members how their funds have been used and what effect they’re having. It’s a wonderful way to tell a visual story, start to finish, of an organization’s value.

cont. on page 24

Open Source Mapping Solutions, cont.

cont. from page 23

In short, a good mapping tool is more than a flashy toy for your website: it's an incredible tool for telling your organization's story, reaching out to members, focusing resources on specific problem areas, and reporting back with data, images and videos.

Open Source Advantages

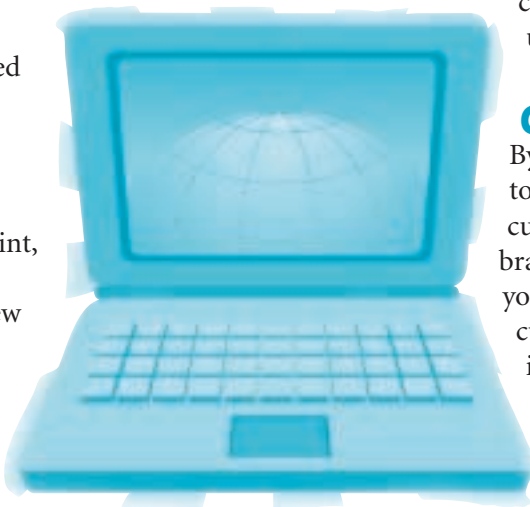
Because this article and this issue are dedicated to Open Source Software (OSS), I'll spend my words enumerating the advantages of open source rather than drawing out point-by-point comparisons to proprietary solutions. But I would be remiss to pretend proprietary solutions don't exist; on the contrary, dozens of software vendors have tackled the mapping problems and are worth exploring (ESRI's ArcGIS, Microsoft's MapPoint, and Clark Labs' IDRISI are just a few examples). River groups might also consider proprietary Application Service Providers (ASP), which are easy to set up but which offer less in the way of customization and control.

Control Over Your Data

Working with a good open source development shop means owning your code when the project ends, and being able to leverage that code for future improvements. This level of control makes it easy to continue development when new needs arise or if increased funding becomes available.

Among the under-valued advantages of open source software in general is the fact that users are never locked into long-term relationships. This means you can shop around for the best fit for every stage of development and avoid being stuck if the developer or team behind your initial software is no longer available. In short, the long-term success of your project isn't tied to a single company or individual.

Once your tools are built, they can be scaled to accommodate a growing and changing set of users. New staff, rotating volunteers, increased data requirements: all of these eventualities are easily solved without concern for extra licensing or user fees.



Customization

By definition, open source tools are well-suited for customization. In addition to branding the look and feel of your tools, it's possible to customize your fields, introduce new data types, alter your database to meet changing needs, and so on.

Another important benefit of open source mapping tools is the ability to keep visitors on your domain while they interact with your maps. With open source solutions, maps are built into the architecture of your site rather than requiring users to click through to the map provider's domain. While this may seem like a minor issue at first, we've found there's tremendous value in telling your group's complete story from within a single domain.

Community

When you start a new open source development project, there's no need to reinvent the wheel. In fact, the river conservation movement has been very smart about leveraging past work to improve future projects. In this way, open source software is philosophically akin to river conservation (community-driven) in addition to making pure business sense (reduced mid- and long-term cost, improved control over data, a more complete identification of data with individual organizations).

OpenSourcery, which is only one of many open source software companies currently working on mapping tools, has developed a series of websites for river conservation groups, each of which builds upon and leverages the improvements of its predecessor. Everyone wins in this situation because the organizations that pay for the initial development benefits down the road when they're able to improve their own software at a reduced cost. It's a different – and arguably better – way to think of intellectual property.

Framework

As I mentioned earlier, for an interactive map to reach its full potential it should be integrated with a CMS. There are many good options out there (which is the subject of a different article), including proprietary and open source solutions. We recommend Drupal because of the community of developers that has grown up around it, the powerful modules it supports, and the quality of the code. There are other open source CMSs and frameworks, all of which have their supporters. Ultimately it's up to each organization to find developers and a framework that will help it thrive.

Due to the number of development frameworks available (Drupal, Joomla, Mambo, etc.) and the different mapping

tools within each, it's beyond the scope of this article to delve into every mapping module available. In fact, a development shop worth its salt will enumerate and describe specific modules when the time is right. The important thing is to know what you need from the tools.

Sources of Data

In addition to the above-mentioned Application Programming Interfaces (APIs)—Google and Yahoo!—there are scores of APIs that serve valuable data. MSN Virtual Earth, AOL MapQuest, Pushpin (commercial), and Mapstraction (open source) are among the leaders. Even NASA provides images via their satellite image API. But perhaps the most exciting thing going right now is integrating your CMS with an open source tool called OpenLayers. OpenLayers was created to be an open source alternative to Google Maps that allows you to pick and choose data from a range of different servers (including your own local data).

OpenLayers is exciting because of the enormous amount of freely available, public(!) mapping data it can pull from places like the USGS, public universities, cities, states, and so on. Some of these organizations keep data on their servers and serve it to you (so you don't have to store the data on your own server), or you can request copies of the data for storage on and retrieval from your own server. OpenLayers remains lean and quick by only grabbing what it needs at any given moment.

Imagine combining a Google topographical map with a USGS watershed map, in turn enhanced by your own water quality data. Now imagine users turning on and off layers to get the exact view they're looking for. That's a powerful tool.

cont. from page 25

Next Steps

So you've decided to add an interactive map to your website. Now what?

First, ask yourself whether you're happy with a static HTML site or whether it's also time to introduce a Content Management System (CMS). The answer to that question begins with understanding how much you want the maps to integrate with your other online tools. If the answer is "not much," you can get away without a CMS. If you want your maps to show water quality data, images, or other info from your site, or if your data is constantly changing, it's likely time for a CMS. You might also think of upgrading to a CMS if you foresee multiple users administering your maps (or other website content). In other words, most river organizations will benefit from the dynamic tools inherent in a solid CMS.

Implementing the right CMS requires technical expertise. Help might arrive in the form of a volunteer or someone within your organization, but generally we recommend you work with a professional with whom you have a paid relationship. You may look at my byline and think twice about this advice, but anyone who has pushed a friendship into the realm of mission-critical work will understand this stance. Too often the relationship changes, the volunteer loses interest, and the project falls apart. You're talking about the central nervous system of your membership and fundraising efforts, so it behooves you to proceed carefully.



And while grant writing deserves its own article, it's worth noting that organizations seeking grant money can make a better case for themselves if they have a working relationship with an established development shop. Not least of all because a good shop can help you write a solid development plan and vouch for your long-term goals.

Let's conclude by summarizing the steps required for successful implementation of open source, online mapping tools. I strongly recommend beginning with a simple search of the services listed in this article. Spend ten minutes creating a custom map with Google. Visit the websites in the sidebar. Seeing how the most basic tools work will help you formulate a more informed vision for your group. The next step is to pour over the possibilities listed here and dream up the ideal mapping tool for your river group. You can prioritize later, but having a vision is important. Understanding your requirements, timeframe, and budget is the best way to prepare for the penultimate step, which is talking with developers. When you make contact, feel comfortable asking tough questions that will position your river group well for the long term. A good development shop will share specific examples of their work and help you determine whether they're a good fit. Whatever choice you ultimately make, take solace in the fact that open source development saves organizations from reinventing the wheel. Armed with a good plan, an honest development shop, and a strong community, your prospects for success are strong.



A Short List of Mapping Tools & Geospatial Data Sources

MAPPING TOOLS

Google Maps

The Google Maps Application Programming Interface (API) lets you embed Google Maps in your own web pages with JavaScript®. The API provides a number of utilities for manipulating maps and adding content to the map through a variety of services, allowing you to create robust maps applications on your website.

<http://code.google.com/apis/maps>

MapQuest

MapQuest Platform Services offers open map APIs and mapping developer tools that can help you easily integrate custom data into innovative mapping applications. With the MapQuest® Platform: Free Edition, developers can create an engaging online experience for their users powered by the MapQuest Platform's free APIs and map developer tools.

<http://platform.mapquest.com/products-free.html>

Map Server

MapServer is an Open Source platform for publishing spatial data and interactive mapping applications to the web. Originally developed in the mid-1990s at the University of Minnesota, MapServer is released under an MIT-style license, and runs on all major platforms.

<http://mapserver.gis.umn.edu>

Multimap

The Multimap Open API allows developers to use the same maps, aerial and Bird's Eye images from multimap.com in your own blogs and mash-ups. The spirit of the Open API is to encourage experimentation and innovation for the community of developers, whether you work for a company or are doing your own thing. www.multimap.com/openapi

Yahoo! Maps

The Yahoo! AJAX Maps API lets developers add maps to their web sites using DHTML and JavaScript®. Maps are fully embeddable and scriptable using the JavaScript programming language. Yahoo! Maps AS-Flash API's built-in geocoder means that you can specify a physical address or latitude/longitude coordinates for your map's location.

<http://developer.yahoo.com/maps/ajax>

GEOSPATIAL DATA SOURCES

Geography Network Explorer

The Geography Network is a global network of geographic information users and providers. It provides the infrastructure needed to support the sharing of geographic information among data providers, service providers, and users around the world. Through the Geography Network, you can access many types of geographic content including dynamic maps, downloadable data, and more advanced Web services.

www.geographynetwork.com

National Atlas

The North American Environmental Atlas is intended for use by both environmental scientists and the citizens of the United States, Canada, and Mexico. Its maps and services are designed to help the public visualize environmental topics at a continental scale.

www.nationalatlas.gov

NSDI Geospatial Data Clearinghouse

The National Geospatial Program provides leadership for USGS geospatial coordination, production and service activities. The Program engages partners to develop standards and produce consistent and accurate data through its Geospatial Liaison Network.

www.usgs.gov/ngpo

USDA Data Gateway

The Geospatial Data Gateway provides One Stop Shopping for natural resources or environmental data at anytime, from anywhere, to anyone. The Gateway allows you to choose your area of interest, browse and select data from a catalog, customize the format, and have it downloaded or shipped on CD or DVD.

<http://datagateway.nrcs.usda.gov/GatewayHome.html>

U.S. Geodata- U.S. Maps and Data

Geodata.gov can help you: find data or map services, make a map, browse community information, cooperate on data acquisitions and publish your data and map services. www.geodata.gov

Considering the Costs

by Waverly de Bruijn
River Network

www.rivernetwork.org

The number of open source projects increased from 16,000 in 2001 to 150,000 in 2007.

~ Red Hat, a professional Linux and open source provider.

The development and popularity of open source software (OSS) has grown considerably in recent years, and can be a great way for a river and watershed organization to save on upfront technology costs. Before jumping on the OSS bandwagon, however, it's important to know the long-term costs of the software. Will it be flexible enough to do what you need it to do, or will you have to re-invest time and money in just a few years? Does the software require tedious maintenance that might call for a staff member or volunteer with related experience—or even a contractor—to achieve, or are upgrades smooth and user-friendly? The following are a few considerations regarding the long-term costs and time investment of open source software. It's not exhaustive, but rather a general list of what might be coming your way and when choosing a particular software path.

components/modules). For more complex software, a drawback is that unless you're enrolled in a subscription support service (see below), you often have to install updates yourself by following the upgrade instructions on the software's website or a related site (which may or may not be written comprehensively or in a format that's easy to follow), or hire someone with expertise in the area to do it for you. Thus, although OSS and upgrades are free, be sure to weigh the level of complexity of installation, maintenance and upgrades against the potential investment of time and/or cost of hiring a tech expert.

Some OSS providers also pride themselves on the frequency in which they publish upgrades. Open Office, the open source counterpart to Microsoft Office, states on its site that "OpenOffice.org releases release software several times a year so you can take advantage of new features as quickly as possible." In the world of vendor-provided software, Microsoft Office comes out with an upgrade every few years, for which you end up paying \$150 and above. Thus, if you need to stay on top of the latest technological developments without paying an exorbitant price, OSS may be a better bet.



Installation and Upgrades

In the same way that one doesn't pay for installing open source software, the cost of upgrades and enhanced functionality is—at its base level—also free. Some software, such as the Mozilla Firefox internet browser, installs in a matter of minutes and alerts you to available upgrades. Installation and upgrades can get difficult the more developed your open source project is (it's easier to install and upgrade an image editor application than a website with many different



Compatibility and Customization

In the wide world of OSS, choices for different products abound. Some products, such as CiviCRM (database software), are compatible with the OSS Drupal content management system (CMS). Other

products don't necessarily communicate well with each other, so be sure to consider the "big picture" of functions you'd like your software to perform, including compatibility with other products. Otherwise, you may find yourself paying a pretty penny for a developer to customize your software to the point at which they work smoothly with one another.

Another consideration is whether you will need to mold and shape your software to meet your unique needs. Many nonprofits contend with software that was primarily designed for the business world—we don't get "payments" from "vendors," we get "donations" from "members." Beyond the annoyance of language discrepancies, sometimes a database or a website needs fairly significant customization. If this is the case, be sure to price out the costs of customizing vendor-provided software (which often must be performed by the company or a designated third party) or open source software (an investment of staff time or paying a developer to alter the code).

Support

Open source systems have gained various levels of notoriety, resulting in "user communities" of various sizes. If you plan on installing and maintaining the software yourself, taking a look at online forum activity will give an indication of the likelihood that you'll be able to post a question and receive a response.

Some hardware and software companies are beginning to offer support packages to open source users. Acquia, commercial support for Drupal web content management system (CMS) and Red Hat, which supports Linux operating system are two such companies. These companies and others offer, by

subscription service, to take responsibility for installing upgrades and new versions of the open source system, fixing bugs and operating glitches that may result, and maintaining site security. Packages range from basic to elite, with associated costs of between \$0 (introductory offers) to thousands of bucks. For lower-level subscriptions (often what non-profits subscribe to due to cost constraints), access to company-fostered online tutorials, chat rooms and lists also enables organizations to troubleshoot with other open source users on how to solve various problems. Again, the option of commercial support for open source software provides an additional layer of assistance (especially helpful to nonprofits that don't always have a dedicated tech person with the time and expertise to devote to software maintenance and development), though it can add significant cost. Weigh this against the level of support received from vendors of proprietary software.

Cost and choice are the major benefits of "going open source," though sometimes the options seem overwhelming. Spend some time thinking about your needs, your capability/level of technological expertise and then apply those factors to your search for the right software. Research the product, its development and upgrade history and the size of the community—and try to estimate required levels of staff time and costs of developer or commercial support. Combined with the advice in the accompanying articles, you should have an easier time choosing the software that is right for your organization.



For a valuable resource that supports the suggestions in this article, see [The True Costs of Open Source](#) by Peter Smith of Arapiki Solutions.

Glossary of Terms

Application Service Provider (ASP):

An ASP is an organization that hosts an application on its servers and makes it available to customers. Users license the application and access it through their web browser via a private connection. Also called Software as a Service (SaaS) and On-Demand software.

Application Programming Interface

(API): An API is a means of requesting functions from a third-party service and having the results displayed on a browser. Used by Google, YouTube, Facebook, iPhone, etc. to give users access to functions without opening up their secret sauce.

Abandonware: Software that is no longer being maintained, released or supported by the manufacturer.

Author: In the case of a software program not created for an employer, the creator(s) of the program. Where the person creates a program as a work for hire, their employer is considered the author.

Binary: An executable program or a data file used by a program.

Closed Source: Software for which the source code is not normally available to the end users of the program.

Compatible: A type of software that will work similarly to another software package (such as one word processor accepting the word processing files of a different word processor).

Content Management System (CMS):

A web-based application designed to make it easy for users to create, edit, upload and publish content to a website. Non-technical users of a CMS can manage a website without fear of breaking the look or functionality. A CMS is often used to store data in an organized fashion.

Copyleft: A type of copyright which uses the protections of copyright to ensure the freedom of users of a program

Copyright: An exclusive right of reproduction or certain other uses of a literary, pictorial, audio or visual work.

Distribute: To transfer copies of a work to others.

End User: The party who operates a computer program to accomplish the performance of a specific task or tasks

Fair Dealing/Fair use: The right to use a copyrighted work in part; even against the permission of the copyright holder. Referred to as “fair use” in the United States.

Free: Software which is either available without charge, or which is available without the usual restrictions of proprietary software.

Freely Redistributable: Software for which the copyright holder waives the right to exclude others from making copies.

Freeware: Software for which the copyright holder allows persons to make copies, usually without allowing access to the source code.

Incompatible: Software which does not normally work in a similar fashion to other competing applications.

License: A privilege to use or copy a program

MIT-style: A type of license which waives almost all of the restrictions imposed by copyright.

Non-Free: To be restricted in some fashion.

Open Source: A type of software which grants the end user the privilege of copying, distribution and/or modification.



Patent: A right granted by the government to allow the owner of the patent to prevent others from making, using or selling an invention.

Proprietary: Software which is property of a specific company for which the source code is usually not available for inspection by end users.

Public Domain: A work which is not subject to copyright.

Shared Source: A type of license where the source code of a proprietary program may be available to certain parties.

Shareware: Software for which a limited license to copy is granted for a short period to allow the end user to try it before purchasing.

Shrink Wrap Contract: A license agreement which can only be read and accepted by the user after opening the product.

Source Code: The exact specifications of a computer program.

Vendor: A seller of computer hardware and/or software.

Vendor Lock-in: Condition where a person or organization has become dependent upon specific hardware and/or software, and where it would be impossible or prohibitively expensive to change to something else.



Open Source Software is a tool that can be cost-effective and beneficial to many watershed conservation organizations. Below are a few testimonials from some River Network Partners currently utilizing some of the many programs available.



I use Open Office software instead of Microsoft Office on our donated computers. I am quite happy with them.

I wish there was an open source operating system other than Linux or the other techno-geek programs. We have five old computers with no disks, and more and more problems. We have no operating system disks, so as they become bug and virus ridden, we can't reinstall. One by one they are becoming inoperable. We don't live in an area where there are volunteers who are techno-geeks, so I am thinking we are going to be Microsoft captives for the foreseeable future.

Appalachia-Science in the Public Interest (KY)
www.a-spi.org

I believe that open source software and the watershed restoration movement have very similar philosophies: both are based on community work and being open and honest about the work that is being done. Citizens drive both efforts.

We use Fire Fox and Thunderbird for our web browser and email client at our office and have found them to be very useful and bug free. We are still using Windows as our operating systems but when we get new computers (with new operating systems), I'm sure that we will switch over to Linux since it is so easy to use, is virus free, and is free to use! The product that we are using the most now is Goldberg, a Ruby on Rails—based web content management system. The design for the site was primarily done in GIMP, a Photoshop alternative. Both tools are free to use and our web programmer is creating the site in his free time for no cost to him or our watershed group. What a great partnership.

Huff Run Watershed Restoration Partnership (OH)
www.huffrun.org



I use open source daily. I run a Linux machine (Ubuntu) and use it for almost all of our computing needs and have saved our watershed council thousands of dollars (our treasurer uses QuickBooks and I have yet to find a compatible Linux program, although there are a few comparable ones, so that's our only exception). In general, I am a huge open source fan and encourage everyone to try using the programs. Most of them I'd recommend even for people who aren't computer savvy. Start with the ones that are available in Windows. If you're comfortable enough with computers and have an extra room on your hard drive, partition your drive and run Linux and windows on your machine (back it up first or do this on a new machine).

Programs I use regularly that are also available for Windows: Firefox web browser (so much safer than Internet Explorer!) Open Office (fully compatible with Microsoft Office, includes word processing, spreadsheet, database, outlook equiv, etc.); Gimp (similar to Photoshop); GRASS GIS (very powerful GIS program. There's definitely a Mac version, I think one for Windows too. I use QGIS as a front end and am not sure if that's available for Windows.)

Programs I use regularly for work that may not have Windows versions: Scribus (alternative to Adobe InDesign layout/desktop publishing); Inkscape (vector graphics, alternative to Adobe Illustrator); Quanta Plus (web design, alternative to Dreamweaver); QGIS (see above, front end to GRASS GIS); gFTP (FTP client) and others.

Difficulties I've encountered using Open Source Software:

- ◆ Running Linux without a reasonably fast internet connection: not recommended. You get all of your programs and updates and information via the net and if you don't have access, you will be frustrated.
- ◆ Learning how Linux works: file structure and installation are the main ones, but Ubuntu is really user friendly now.
- ◆ Making Linux work with a laptop with too many proprietary drivers: this can be overcome in three ways, with the first suggestion being easiest.
 - 1) Buy a laptop that comes with Ubuntu and support (Dell offers them).
 - 2) Use a desktop or an older laptop: these are generally much easier to install Linux on cold.
 - 3) Don't install the new versions of your operating system the day they come out. Wait a few months while the bugs are worked out and other people with the same machine can recommend fixes.

Salmon and Drift Creek Watershed Council (OR)
www.salmondrift.org

Resources & References

Organizations & Networks

Aspiration connects nonprofits to software solutions that help them more effectively meet their missions of positive global change. Aspiration connects and strengthens physical communities of nonprofit users and software developers by convening and facilitating innovative technology events; connects and strengthens virtual communities of nonprofit users and developers through the Social Source Commons; and provides capacity building services, working with software developers to produce better tools for nonprofits, and with nonprofits to better leverage these tools. www.aspirationtech.org

The Community Technology Centers' Network (CTCNet) is a national membership network of community technology centers (CTCs) and other non-profits, united in their commitment to provide technology access and education to underserved communities. www.ctcnet.org

The Free Software Foundation (FSF) is a 501(c)3 donor supported charity founded in 1985 and based in Boston, MA. The FSF has a worldwide mission to promote computer user freedom and to defend the rights of all free software users.

Idealware, a 501(c)3 nonprofit, provides candid Consumer-Report-style reviews and articles about software of interest to nonprofits. Through product comparisons, recommendations, case studies, and software news, Idealware allows nonprofits to make the software decisions that will help them be more effective. www.idealware.org

Nonprofit Open Source Initiative (NOSI) is an organization that was started to facilitate and encourage the use of free open source software in the nonprofit sector, and provide concrete resources to help organizations choose and use free and open source software. <http://nosi.net>

Organizers Collaborative (OC) is a nonprofit organization that identifies and develops low-cost, replicable technology, software, and trainings that meets the organizing, outreach, and fundraising needs of small nonprofits. OC is comprised of community organizers and technology consultants dedicated to helping nonprofit organizations and community groups integrate technology into their work. <http://organizerscollaborative.org>

Social Source Commons is a place to share lists of software tools that you already use, gain knowledge and support, and discover new tools. It's a place to meet people with similar needs and interests and answer the question: what tools do they use? <http://socialsourcecommons.org>

TechSoup offers nonprofits a one-stop resource for technology needs by providing free information, resources, and support. In addition to online information and resources, they offer a product philanthropy service called TechSoup Stock. Here, nonprofits can access donated and discounted technology products, generously provided by corporate and nonprofit technology partners. www.techsoup.org

Phase2 Technology provides web solutions for nonprofit, commercial and media publishing clients using open source technology including Java and Drupal. www.phase2technology.com

OpenSourcery powers startups, nonprofits and social service agencies with cutting-edge technologies. They believe in working with companies that share a vision of sustainable business. www.opensourcery.com

Articles & Resources

Generally Recognized as Mature Open Source Software Programs. This paper identifies Open Source Software/Free Software (OSS/FS) programs that are generally recognized as mature (GRAM), also known as generally recognized as safe (GRAS). www.dwheeler.com/gram.html

Nonprofit Open Source Initiative Primer. A useful overview of what open source software is, including a chapter on comparing vendor-provided and open source options, provided by a nonprofit organization dedicating to facilitating the use of free and open source software in the nonprofit sector. <http://nosi.net/projects/primer>

At local Penguin Days, nonprofit organizations explore the range of issues and options involved using Free and Open Source Software. www.penguinday.org




SourceForge provides fast, secure and free downloads from the largest Open Source applications and software directory. <http://sourceforge.net>

Vendors as Allies: How to Evaluate Viability, Service and Commitment. A look at how to evaluate the intangible aspects of vendor-provided software, like long-term viability, support and their commitment to your organization. www.idealware.org/articles/vendors_as_allies.php



Let River Network Help You Keep Your Head Above Water!

Join the River Network Partnership and connect to the information and resources you need to stay afloat!

-  Access our River Source Information Center with the 1-800 hotline: Let us help you research a particular issue and put you in touch with the necessary contact and resources through one-on-one consultations.
-  Log onto our Partner-only website: Browse updated postings of funding sources, upcoming events and trainings, and templates and river clipart.
-  Receive numerous other Partner benefits: Benefits include subscriptions to *River Voices* and the *River Fundraising Alert*, access to the online *Directory of Funding Sources for River and Watershed Conservation Groups*, one free publication, discounts on all other River Network publications, and much more!

- New Partner
- Partner Renewal

Sign Me Up!
www.rivernetwork.org



Annual Partner Dues are only \$100.

- Organizational Partner Agency/Tribal Partner Individual Partner

Name _____ Phone (____) _____

Org/Agency _____

Address _____

City _____ State _____ Zip _____

Email _____ Website _____

- My check is enclosed
 Please charge my credit card: Visa MasterCard American Express
 Card # _____ Exp. Date _____
 Signature/Name on Card _____

- First time Partners will receive an initial set of Partner materials, including your choice of (check one):
- How to Save a River Starting Up: A Handbook for New River and Watershed Organizations
 - River Talk! Communicating a Watershed Message Listening to Watersheds - A Community-Based Approach to Watershed Protection
 - Testing the Waters - Chemical & Physical Vital Signs of a River

Please make your check out to River Network and return this form to:
 River Network, 520 SW 6th Avenue, Suite 1130, Portland, OR 97204.

River Network works to support you and your needs. We provide training and technical assistance to our Partner groups. River Network does not promote legislation or represent your organization in legal matters.



River Network

Connecting People, Saving Rivers

520 SW Sixth Avenue, Suite 1130
Portland, Oregon 97204-1511

ADDRESS SERVICE REQUESTED

Education, Inspiration, Celebration



River Network's

2009 National River Rally

May 29 - June 1 ■ Hyatt Regency Baltimore ■ Baltimore, Maryland

River Network's 10th Annual National River Rally will include:

- A series of intensive, 4-hour workshops on Friday afternoon;
- Over 70 additional workshops covering topics of interest to staff, volunteers and board members of river conservation organizations;
- Office hours, mentoring sessions and real life, in-person conversations with funders and exhibitors;
- The River Heroes Banquet: a memorable and inspirational night of celebrating our own leaders;
- Unsurpassed networking opportunities, including field trips, a night at Camden Yards, an exclusive evening at the National Aquarium, breaks, meals and informal after-hour gatherings;
- River Action Day: American Rivers is organizing a day on The Hill for the river conservation community. Spend Monday learning how to make the most out of your visit and spend Tuesday meeting with your congressional leaders.

*Together, we will
celebrate rivers, teach and
learn from each other
and explore the power of
citizen action.*

For more information and
to register, please visit
www.rivernetwork.org/rally