Dispelling the Myths About - Free Open Source Software (FOSS)

I prompted myself to write this article after reflecting on my experience of trying to spur on the land surveying community to collaboratively develop a set of integrated processing tools that would assist land surveyors in meeting the national integration of legal surveys initiative. The proposed tools were based on using a framework of open source programs. Many members of land surveying associations and provincial and federal government organizations that I talked to expressed great resistance to the notion of using open source software – it was a hard sell. This article provides a brief overview of the benefits of open source software, demonstrates the power of community software development and attempts to dispel some common myths about open source projects.

Background

So what does free mean? The modern definition basically states that anyone can use, modify and redistribute the program to the world so the whole community benefits. Free does not refer to price but the liberty to run, copy, distribute, study, change and improve the software. You do not have to ask or pay for permission from the author(s) and you can charge a fee for distribution. Free does not mean there are no costs. The code has to be developed, tested and maintained.

The Free Software Foundation (FSF) is considered to be one of the leading organizations that promotes the ideals of free software. The FSF provides an environment for free software development including the provision of enabling tools and licenses for distributing the free software through their GNU operating project. There are a number of other organizations undertaking similar philosophies and stewardship roles.

Open source promotes communities to develop common framework applications that can be used by many and therefore creates efficiencies and avoids duplication of effort. Once a common framework exists, it makes it attractive for developers to use it as the core for creating other derivative works for specific add-on applications. The open source community is still trying to determine and understand what kinds of licenses are best for encouraging open source development. Some common licenses are GNU GPL, GNU LGPL, Mozilla, MIT and BDS.

The most common type of user documentation for an open source project is a *wiki* which is a website that allows visitors to add, remove, edit and change content. Wiki's allow for linking among any number of pages and the user can do text searches. This makes it ideal for open source developments to dynamically document user guides, sample use cases and applications for a project – it is essentially a data base for creating, browsing and searching information online. Many tools have developed that make the edit process painless for even a novice user. The open source community also extensively uses developer/user mail lists and real-time chat services to effectively communicate.

The actual open source code development is administered by a *Concurrent Versions System* (CVS) which keeps track of all work and all changes in a set of files, typically the

implementation of a software project, and allows several (potentially widely-separated) developers to collaborate. CVS provides a web front-end for hosting and maintaining project homepages, bug tracking, CVS, FTP, and mailing lists.

It is interesting to note that many large corporations directly support open source projects and some are developing creative ways to involve students in the process. For example, Google started a program called *"Summer of Code"* in 2005. This is an annual program, in which Google awards cash prizes to students who successfully complete a free/open source coding project during the summer. The student selects an open source project and proposes about a three month duration specific coding enhancement. The mentor organization evaluates the proposal and then submits it to Google for review. 131 organizations and over 900 students were selected to for the 2007 program. The mentors work closely with the students and if the project is satisfactorily completed the student receives $4,500 and a *"Summer of Code"* T-shirt from Google. Many of these students end up staying with mentoring organization.

From my observations, it appears that the general make-up of an open source project consists of a core of a few dedicated developers, a larger ring of interested collaborators who provide problem reports, bug fixes, and occasional enhancements, and a realm of general users. If the open source project is on the right path, the community grows and key new developers and support emerges from the outer rings to create a healthy, mature and robust developer/user environment.

There is a great deal of information available on the history of open source software development. Searching on the internet, you will see that there are Open Source projects currently underway for just about every major infrastructure software technology you can think of. Some examples that you may have heard of are *LINUX, APACHE Web Server, Perl, MySQL, PostgreSQL, PostGIS, Wikipedia, MapGuide (OS), etc.* I encourage you to explore the opportunity of using open source software and consider becoming an active contributing member of the user/development community within your specific area of interest – you may find it very rewarding.

Common myths about *Open Source Software.*

1) **Open Source Software Isn't Reliable or Supported (Answer: False):** Every single internet address (web and email) depends on the Domain Name System which is dependent on the open source program "BIND" and Virtually any email message sent over the net relies on "sendmail" which are both reliable open source software. It is well known that the open source Apache web server hosts more than 60% of the world's web sites. These programs have demonstrated a robustness under extremely fast changing conditions. Mature open source software is peer-reviewed and as bullet proof as software ever gets.
2) **Big companies don't use open source software and is not for mission critical applications (Answer: False):** "Perl" is a leading open source development language and is probably used by most Fortune 500 companies. Banks tend to be conservative institutions, however, you find they are venturing into using open

source for supporting their online transaction systems which are certainly mission critical.

3) **Open Source is all about licenses (Answer: False):** Open source is about internet-enabled collaboration and a community software development methodology. Licenses play a role only to the extent that they set out rules designed to make sure that companies don't undermine the playing field.
4) **Open source will destroy the software industry (Answer: False):** The benefits of open source are exactly the same as the benefits of any other free market - competition between multiple suppliers results in lower prices, more innovation, and specialization to meet the needs of new niches. This is in fact the chief benefit of open source - you are no longer locked into a single-source supplier. If worst comes to worst, you can solve your own problems; more likely, there will be a variety of third-party vendors who will be able to support you better than a single-source vendor ever will.
5) **There is no support (Answer: False):** Most open-source projects have a core of developers, Internet mailing lists, archives, support database and a wiki - all available at no cost. If a specific open source software has developed a mature community, it is surprising how fast you can find an answer if it is not already documented and you can always contract the services of a consultant.
6) **Open source only runs on Linux (Answer: False):** No - Most open source products run on a variety of operating systems.
7) **Open source is just like commercial software (Answer: False):** Choosing open source means you must take a more active role, but you have much more control of the process. One is not better than the other; they're just different. You must decide which is the right method to achieve your software goal.
8) **There is no product direction (Answer: False):** Open Source project maintainers typically start an Open Source project to solve a specific need and grow the project to meet other or increasing needs over time. They tend to have very specific roadmaps and accept feature and functionality input from other interested developers/users.

Why would I want to use Open Source? – The Benefits

1) **Reliability:** Reliability basically means the absence of defects which cause incorrect operation, data loss or sudden failures. Open source code tends to be significantly more reliable (bug-free) than commercial software. This is likely due to the code being totally open and peer reviewed and developed by experts that are passionate about the project. David Wheeler's website cites numerous empirical studies regarding the reliability and stability of open source products. For example, IBM determined that *"the Linux kernel and other core OS components -- including libraries, device drivers, file systems, networking, IPC, and memory management -- operated consistently and completed all the expected durations of runs with zero critical system failures."* Another example is the Apache Web-server which was demonstrated to have half the downtime of the leading commercial web server.
2) **Stability & Flexibility:** The code for commercial software is proprietary. This means that you are tied to the maintenance and upgrade schedule of the supplier.

And furthermore, sometimes commercial suppliers establish a virtual monopoly and then force upgrades onto its users. Open Source puts the choice of upgrading and maintenance in the hands of the users rather than the suppliers. Having access to the source code can also allow a business to choose to support itself on an old version where necessary or completely carry on with maintenance of the code with in-house resources or by consultant services. Having access to the open source code also provides an option to develop specific extension applications that the user requires according to their own schedule and control - this provides long term stability and great flexibility. Open Source software provides freedom by eliminating dependency and lock-in to a vendor and the ability to tailor the software to the way you do business.

3) **Audability:** Closed-source software forces its users to trust the vendor when claims are made for qualities such as security, freedom from backdoors, adherence to standards, maintenance and upgrades. Access to open source enables an informal inspection or, if required, a more rigorous third party auditing or certification.
4) **Cost:** Many Open Source projects are available free of royalties and fees. However, the solution with lowest *Total Cost of Ownership (TCO)* is usually the most desirable. There are numerous arguments that favour open source software as the lowest TCO solution that include: often zero purchase price; no mandatory need to account for copies in use thus reducing administrative overhead; reduced need and no fee for upgrades which also reduces management costs; longer uptimes which reduces the need for expensive systems administrators; near-zero vulnerability to viruses; lower vulnerability to security breaches and hack attacks reducing systems administration load; ability to prolong life of older hardware while retaining performance.
5) **Accountability:** A common question about the use of open source software is *Who do I sue if the software does not work? …who is liable?* These questions seem to be plausible arguments at first glance, however, if you look at the typical commercial software user license you will see that it absolves the vendor of liability for almost any problem you may incur.
6) **Support:** A good open source community has excellent documentation (wiki) and a very responsive user/developer mail list (and archives) that will be able to answer any almost any question. You can also always obtain support through numerous consultants, if necessary, however, when software works reliably, support ceases to be a primary concern.

What to look for in an open source project

A paper entitled "*The State of Open Source GIS"* by Paul Ramsey of Refractions Research states that successful *Open Source* projects are "*created through the growth of communities of shared interest.*" Paul lists the following key questions to consider when selecting open source software:

Is the project well documented?
Is the development team transparent?
Is the software modular?
How wide is the development community?
How wide is the user community?

Paul's paper goes on to provide an excellent overview of open source projects within the GIS field including the User Friendly Desktop Internet GIS (uDig) which Paul's company performed the initial development as a project under the GeoConnections program. Refractions Research continues to provide the driving force for development and community support for this project. uDig is the framework proposed to build the set of integrated tools for land surveyors.

I joined the uDig developers list (just to observe the process) and over time it was fascinating to see such dedication and open communication emerge within the community members. New list members are warmly welcomed and I was truly amazed at how fast they respond to questions and spend such great effort to solve each others questions. Building this friendly open community of common interest seems to be critical in fostering the successful growth of an open source project.

Refractions Research also developed PostGIS which spatially enables PostgreSQL (an open source object-relational database) to support geographic objects. It is used as a backend spatial database for geographic information systems (GIS), much like ESRI's SDE or Oracle's Spatial extension. PostGIS continues to be developed by an active community supporting functions such as data validation, topology support, coordinate transformation, programming APIs and much more. There are many successful case study implementations depicted on their website.

Case study: Gnu Gama – Least Square Adjustment program

Gama is an open source program for performing a least square adjustment of 1D, 2D and 3D survey observations. The project lead is Ales Cepek at the Department of Mapping and Cartography, Faculty of Civil Engineering, Czech Technical University in Prague. Gama adjusts observed coordinates, distances, angles/directions, height differences and 3D vectors in a local coordinate system. Gama is the survey observation processing engine proposed for the set of integrated tools for land surveyors.

Gama is the first open source program that I have used. I will never forget when I posted my first message to the Gama mail list asking if there was a way to integrate lower accuracy GPS derived coordinate observations and conventional distance and direction observations using Gama. Ales answered my question immediately. First of all he mentioned that the program was designed to adjust geodetic networks and did not have a provision to use coordinate observations with a large standard deviation. Ales saw the application need and decided to quickly modify the code to enable a ten metre threshold for the standard deviation of coordinate observations. It worked great.

As Gama was designed for use in Europe, the input for directions was in grads. I posed my second question and asked Ales if the program could be modified to accommodate sexagesimal degree input. Again Ales modified an input switch to allow degree or grad input for angles or directions. Needless to say, I was extremely impressed with Ales's open communication and quick response. I have never had such support service with a commercial product. This is a good example of how users and developers can interact

across international borders to quickly develop new requirements with an open source project.

Closing Remarks

Open source projects have steadily gained momentum over the last ten years. Like any new process, it takes time for the public sector and industry to become aware, understand and finally embrace this new concept. Tim O'Reilly states in his 1999 article *"… if you value competition and innovation, it's in your best interest to support the open source software development community. Not only should you be experimenting with open source products, you should be learning from its processes. My dream is that we can have the best of both worlds: a vibrant commercial industry based on openness and cooperation where it makes sense, and competition and proprietary advantage where it makes sense."*

We are fortunate to have the annual Free Open Source Software for Geospatial (FOSS4G) 2007 conference being held in Victoria in September which brings together the people who create, use, and support open spatial software. There will be over 120 presentation sessions with over 500 expected attendees. I encourage our members to explore the FOSS4G and consider attending this conference.

References:

http://www.oreillynet.com/pub/a/oreilly/opensource/news/myths_1199.html
http://en.wikipedia.org/wiki/Summer_of_code
http://www.foss4g2007.org/
http://campustechnology.com/articles/38779/
http://www.refractions.net/white_papers/index.php?file=2006-06-01_oss_briefing.data
http://www.fig.net/pub/fig_2002/TS2-1/TS2_1_cepek_pytel.pdf
http://www.fsf.org/
http://www.theaceorb.com/product/benefit.html
http://open-source.gbdirect.co.uk/migration/benefit.html
http://www.dwheeler.com/oss_fs_why.html
http://postgis.refractions.net/documentation/casestudies/