
OSGeo Journal

The Journal of the Open Source Geospatial Foundation

Volume 2 / September 2007

In This Volume

Topology Basics

1Spatial: *Data Quality Concepts*

Introducing MapWindow & GeoNetwork

LizardTech: *Why we use Open Source software*

Local Chapter Reports: Taiwan, U.K., Francophone, Spanish...

Case studies: UN FAO, Fishing Vessel Tracking...

Community Event Reports: India, France

GRASS & distributed computing

News & Software Updates...

Sponsor Perspectives

LizardTech: Why We Use and Support Free and Open Source Software

An Informal But Earnest Manifesto

by Matthew Fleagle and Michael P. Gerlek, LizardTech

One of the authors, Michael Gerlek, has been a Charter Member from the beginning of OSGeo and was highly influential in its success. LizardTech officially became an OSGeo Associate Sponsor in 2006 and has been an active user of open source geospatial tools for a long time. It is with this in mind that they were invited to describe how they use open source from both a business and community perspective. – Editor

As a small company with a niche-market customer base originally built on the strength and ubiquity of its proprietary technology – the widely used MrSID technology and image format – LizardTech has reason to cherish private ownership. The fact that we have something that no one else can use without paying us for it has, to say the least, advanced our prospects over the years. Many might be unaware, however, not only that we support open source development efforts in general but that in particular, we have been supporting OSGeo since its inception.

Why would a small, proprietary shop like LizardTech use and support free and open source de-

velopment? Don't the advance of open source development and the spread of an ethos of sharing in the software industry represent threats to LizardTech's business model? We believe the contrary, and this article aims to articulate why. First we'll discuss some of the free and open source software (FOSS) products that we use or have used in the past, and then we'll talk about why it makes sense strategically for small companies like ours that have proprietary code not only to use the products made by the free and open source development community but also to materially support its work.

How and Why LizardTech Uses FOSS

LizardTech has used free and open source software packages since its founding in 1992. An informal survey of LizardTech's development team revealed just how much we rely on open source tools and libraries: we quickly were able to list dozens of packages, ranging from the popular (Firefox, Apache, Linux) to the geo-specific (GDAL, PROJ4, OpenLayers). Many of these libraries are built into our shipping, commercial software products. In fact, by comparison to the following list we use very few closed source, third-

party libraries.

Geospatial libraries: GDAL/OGR, GeoTIFF, GEOS, OSSIM, PROJ.4, MapServer, PostGIS, OpenLayers

Non-geospatial libraries: Berkeley DB, CppUnit, Crypto++, libcurl, dlmalloc, efence, Expat, gSOAP, libgif, libjpeg, libtiff, OpenSSL, libpng, libungif, xalanj, xercesc, xml4c, zlib, zthread, boost, littlecms

Other tools and packages: Apache, Tomcat, gcc, Linux, Perl, Python, Putty, Firefox, Cygwin, Wireshark, Emacs, gnuplot, siege, autoit

Okay, you say, you're impressed by the number of free and open source software packages out there that can be used in making geospatial software, but you wonder, are there benefits besides the obvious savings in license fees? By way of answer, we'll not only enumerate the benefits, we'll also give a couple of examples. Here are the benefits:

We don't have to reinvent the wheel. LizardTech is a sleek outfit. We don't have the resources, in terms of personnel and their time, to create everything from scratch, especially when it would be duplicated effort. If the work has been successfully done already, it makes good business sense not to waste time, money and sweat beating a new path up the same mountain. This is good strategy even if you decide to pay for the existing product. When companies start up they deploy OpenOffice.Org or license MS Word; they don't routinely invent new word-processing programs just because they know they'll need to send interoffice memos.

We can concentrate on our core competencies. LizardTech lacks experience in some areas that the open source community has plenty of experience in. This point is similar to the first, but it bears mentioning that we save money when our own developers can concentrate on what they do best. Diverting our energies into areas where we are not experts and don't need to be experts, when the experts stand by proffering their wares at little or no cost, is foolhardy.

We can play in larger arenas. Having access to intellectual property beyond the sum of our developer parts means that we can equip our products to play in larger fields without taking on the costs of finding and engaging additional personnel. This point is again related to the first benefit, but it highlights the opportunity to widen our customer base and sell more products to more people.

We can leverage well-used, well-tested code. The open source development community is not three guys in a garage – not anymore anyway. The community actively hammers away at open source projects, putting code to the test, finding out where it breaks, discovering bugs and supplying patches and other solutions in a robust volunteer testing and feedback loop. The code evolves at the pace of real-world use, testing and fixing, not of marketing schedules and sales imperatives. As a result, open source projects represent some of the tightest products never sold.

We enjoy unplanned advantages. Not uncommonly, a side benefit will emerge through the use of an open source project, some benefit that we may not have had any knowledge of beforehand. Intrigued? Read on...

Real-World Examples

LizardTech wanted to introduce reprojection to its popular GeoExpress product. This would give us access to a wider customer base by better synchronizing the functionality of our product to the identifiable pains of potential users. But while we are experts in imaging technology, we are not experts in reprojection. Projection systems involve hard, scary math. Moreover, in many applications people's lives depend on their accuracy, so they present the twin horrors of being simultaneously mission critical and easy to get wrong. Even if we'd had the personnel to send off to work on reprojection, we had rathered not mess with it.

Our options, then, were two. We could buy a commercial, proprietary package or we could use an open source package like the PROJ.4 reprojection library.

The second option was the hands-down smarter choice, and not just because of the absence of licensing fees. What if the corporate product had a bug that affected our project? What if it only worked on 32-bit systems but we needed it to work on 64-bit systems? We'd have to put in a bug report or a feature request for 64-bit support and then wait six months for the next release of the product, paying for a maintenance plan the while. And 64-bit support may or may not be in the box when we finally got to peel off the wrapper.

By contrast, with PROJ.4 we could likely get a bug fix or 64-bit support in short order. It's true that there is no "company" standing behind

the PROJ.4 package, but we've found (especially in larger projects) that this is rarely a problem. Unafflicted by corporate boardroom pressures, the open source community's priority is finding and fixing bugs in its code and adding features and functionality to its products. (In fact, we very frequently satisfy a particular customer request by taking an open source package and porting it to a compiler such as SunPro on which we previously hadn't supported that package.) If there was something seriously wrong with open source code, then we could pay someone in the community to fix it and still come out ahead in terms of time and cost.

While PROJ.4 gave our product a very important and visible feature, another kind of open source product supports our product invisibly. We use GDAL, a library that supports dozens of image formats used in GIS alone, to make sure that our product supports those formats. Rather than supplying a feature or functionality, GDAL gives us access to the wider GIS ecosystem. More than once we have found out through our customers that we support a format that was important to them that we otherwise may not have bothered with – the unexpected advantage mentioned above.

How and Why LizardTech Supports Open Source Development

It's true that it just wouldn't be cricket to take advantage of the now abundantly evident benefits of FOSS while refusing to support free and open source development and continuing to charge money for our own products. But LizardTech's central claim in this article is not that our use of and support for open source is moral (another paper, perhaps, submitted to a different journal). We're claiming that it's strategic.

There are a number of ways an organization can support open source development in material ways, that is, beyond simply using FOSS. They include monetary donations and sponsorship; membership and volunteering; feedback, bug fixing and patches; contracting open source developers for work, such as we did for 12-bit JPEG support; and attendance at conferences. LizardTech is active in the open source community in all of these ways, often as part of our regular workaday world. Here's how supporting open source benefits us:

If open source projects fail, we fail. It must be remembered that as a company, LizardTech relies on

open source products in order to create and improve our own products. Seen from this single perspective, it makes sense to feed the chickens that lay the eggs, especially when what you pay in the grand scheme amounts to ... ahem ... chicken feed.

The return on investment is high. Contributing, say, \$5000 a year to sponsor an organization like OSGeo, where that money helps keep open source projects alive and vital, makes good sense when you compare the options that would exist in a world without FOSS. Hiring several additional developers would cost several times as much annually, and supporting open source, which provides us with well-tested code and quick fixes when necessary, is a much better use of funds than paying a proprietary company for software that might be more buggy and harder to get fixes for. For that matter, how does one know if a product or library is buggy? Would you rather rely on the biased marketing messages from the proprietary owners of code or access firsthand experience from users and developers? Sponsorship in this regard is like a different and cheaper kind of contract labor.

We can effect positive, rapid change in the products we need to use. It has never been necessary for us to lobby for or engage in radically changing an open source package, but if a substantial change was necessary it's a cinch that the open source development community could turn it around much quicker than a proprietary code-owner whose priorities and schedules are beholden to the boardroom.

Open source conference attendance pays dividends. Given finite budgets, why would we want to take several of our development crew to the annual OSGeo conference instead of to one of the well-attended "proprietary" user conferences? Because not only will they get exposed to all the cutting edge technology that they'll find at other shows, but they'll also acquire knowledge they can bring back and use for free. Further, the information and knowledge they'll get is divorced from any corporate agenda: no one will try to sell them a grand solution or ply them with company-biased information. Finally, the developers they'll meet at open source conferences are more available for follow-up with conversations spawned at the event. A related way to support open source development is to donate facilities, soda and snacks to a local user group. LizardTech hosts a small monthly OSGeo meeting in

our Seattle offices.

Open source plays well with open standards.

We have also invested heavily in supporting ISO and Open Geospatial Consortium (OGC) standards, which are "open" in a way that is different from the way open source is. We've found that the open standards and open source development processes are similar enough that many advantageous synergies emerge between them.

It's the right thing to do. We said above that this was not about morality, but if science hasn't yet proven that the world smiles along with you when you smile, it's only a matter of time. It cannot but advance the wholly considered interests of any 21st

century company to adopt toward customers, competitors and partners an attitude that reflects the values of cooperation, collaboration, community, trust and humility embodied by open source development.

Matthew Fleagle

LizardTech's technical writer

<http://www.lizardtech.com>

[mfleagle AT lizardtech.com](mailto:mfleagle@lizardtech.com)

Michael P. Gerlek

LizardTech's engineering team manager

OSGeo Charter Member

<http://www.lizardtech.com>

[mpg AT lizardtech.com](mailto:mpg@lizardtech.com)

Editor in Chief:Tyler Mitchell - [tmitchell AT osgeo.org](mailto:tmitchell@osgeo.org)**Editor, News:**

Jason Fournier

Editor, Case Studies:

Micha Silver

Editor, Project Spotlights:

Martin Wegmann

Editor, Integration Studies:

Martin Wegmann

Editor, Programming Tutorials:

Landon Blake

Editor, Event Reports:

Jeff McKenna

Editor, Topical Studies:

Dr. Markus Lupp

Peer Review Manager:

Daniel Ames

Acknowledgements

Various reviewers & the GRASS News Project

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this journal, the $\text{\LaTeX}2_{\epsilon}$ style source has been kindly provided by the GRASS and R News editorial board.



This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 License. To view a copy of this licence, visit:

<http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



All articles are copyrighted by the respective authors. Please use the OSGeo Journal url for submitting articles, more details concerning submission instructions can be found on the OSGeo homepage.

Journal online: <http://www.osgeo.org/journal>

OSGeo Homepage: <http://www.osgeo.org>

Mail contact through OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8



ISSN 1994-1897