

A background image of many white birds flying across a sky with a color gradient from red at the top to yellow at the bottom.

The Top-Five Mistakes to Avoid When Migrating to Open Source

By Andy Astor and Donald K. Rosenberg

MIGRATING AN ENTERPRISE

to a new technology is always challenging; it's a voyage into the unknown that can take months or even years from planning to implementation. Open source migrations are no different. They involve combining time-tested enterprise policies and procedures with new practices. Successful migrations require an understanding of the open source environment, and there's no sense compounding your difficulties by making avoidable mistakes based on misunderstandings.

The most common misunderstanding about Open Source Software (OSS) is that it's truly cost-free. You probably already know this is impossible. Of course, there are savings on licenses, but licenses are only a portion of the total cost of software ownership. Training, maintenance, customization, conversions, and ongoing support dwarf

initial licenses by comparison.

Of course, OSS does drive down overall costs by increasing competition. OSS offerings push down proprietary prices. Furthermore, competition from multiple open source sources likewise trims prices to customers. The "open" in open source enables customers to switch vendors more easily, allowing them to find the best levels of quality and price for their needs. Lower cost, if you experience it, is a byproduct of the freedom to choose.

But other than cost reduction, what myths do you believe in? The worst myths are the ones you create for yourself by jumping to conclusions. Take a deep breath, remember your enterprise best practices, and be assured your experience in administering enterprise software makes a good foundation to continue. You don't give up good practices; you simply add some new tricks as you move to OSS.

Failing to Minimize Transition Impacts

All migrations involve transition costs; effective enterprise managers plan for them and look for ways to minimize them. Significant costs may include:

- Modifying your applications to run in a new environment
- Training staff in new platform architectures
- Data migration (often the mission-critical budget killer).

Consider various strategies for controlling costs. For example, to ease migration, look for products that offer Application Program Interfaces (APIs) similar to what your staff already knows. Berkeley Software Distribution (BSD) is a Unix variant, and Linux is sufficiently like it to make transition from Unix environments easier. This is a large part of why migrating to Linux is such

an effective strategy for many companies.

Look for offerings with similar benefits in other parts of your enterprise software stack, such as databases and application servers.

Move upward through your software stack looking for likely substitutions. Although there are ready-made software stacks, such as those from SpikeSource (<http://spikesource.com>) and SourceLabs (<http://sourcelabs.com>), you might want to take advantage of the freedom to choose niche solutions that best suit your purposes.

There are several flavors of BSD. OpenBSD (www.openbsd.org), for example, is famed for security, and companies such as Novell and Red Hat specialize in supplying Linux to large businesses. You can engage consultants such as Optaros (www.optaros.com) if you want help making these decisions.

Expecting 100 Percent Migration

Your current operation is already heterogeneous. Legacy systems are intact, but there are a few Apple users, too. Linux and Windows are both present. What about your Website? Seen in this context, a migration is simply more of one thing and less of another.

Believing that a migration is simply a series of cookie-cutter implementations, such as “Let’s install LAMP (Linux, Apache, MySQL, PERL/PHP/Python)” curtails the flexibility OSS offers. LAMP isn’t truly an entity; it’s simply a handy way to talk about a functional stack, a way of assuring non-open source users there’s depth to open source product offerings by showing there’s something to put on top of that free operating system.

Depending on your needs, you could install substitutes—open source or not—for any of these stack components, from operating system (e.g., BSD or Windows) upward through application server (e.g., JBoss or WebSphere), database (e.g., PostgreSQL or EnterpriseDB), and application logic (e.g., C# or Java/JavaScript). Don’t forget, vendors such as SpikeSource and SourceLabs offer tested, tuned, and sometimes supplemented-function versions of the software you were thinking of just downloading. Don’t shy away from a best-of-breed approach. Remember, custom implementations let you run your business your way and may be more secure. A unique implementation is less susceptible to online attacks from those who would exploit software aimed at cookie-cutter stack implementations.

Windows is unlikely to relinquish its hold

on the enterprise desktop market anytime soon. Even if your company decides to migrate to Linux front-ends, that migration will take time and many users will resist. Identify and mollify. Not everything will be migrated to open source, at least not immediately. Keep an eye on products such as Win4Lin (www.netraverse.com), and remember the power of Linux to network with all sorts of machines and operating systems. Total migration can be your goal, but give it time.

Fearing Open Source Licenses

Your work in enterprise IT has taught you that licenses are to be taken seriously, studied, and followed, and that corporations use software under many different licenses. OSS licenses are no different: study and follow them. There are a large number of them, but they actually break down into two basic types:

- Those that require you to share your own code, such as the GNU General Public License (GPL)
- Those that impose virtually no restrictions, such as the BSD.

Of course, there are subtle differences among the 60 or so licenses the Open Source Initiative (www.opensource.org) supports, such as some that segment the code into a shared core and proprietary peripheries. But OSS licenses are no more complex or scary than any of the proprietary licenses you’re already familiar with. For more on licensing issues, see Lawrence Rosen’s book, *Open Source Licensing* (Prentice Hall PTR, 2004).

Even the GPL holds no terror for those who know to:

- Make source code accessible to those receiving the binary code
- Avoid compiling code under the GPL with code under other licenses without checking to see whether the combination is compatible.

These aren’t complicated concepts; just unfamiliar ones.

Be aware also, that some OSS licenses give rights merely to inspect the code, not change it, while others allow you to change but not redistribute it to others. Still others allow you to freely distribute your changes. Rather than grumble about the differing terms, find ways to harness them to serve your business goals. The various licenses were written to serve different purposes;

you should likewise be sure of your objective before you choose a license to match it.

A company planning involvement with OSS code should set up a committee to oversee the suitability of:

- Taking in OSS code and posting a list of approved intakes
- Contributing in-house code to OSS projects.

Monitoring intakes ensures code is legal and robust and avoids wasteful duplication by different parts of the company trying to download and implement the same OSS. Monitoring contributions keeps your proprietary code proprietary and makes sure you’re following licenses; this is a duty.

Ignoring the Community

One of the truly wonderful things about OSS is how much help and resources are given away with no thought of return. Contributors understand their contributions make everybody richer; that a rising tide will float all boats.

If you’re doing enterprise-level computing, you have a serious code shop. Not battalions of developers, possibly, but enough, and they can do more than simply write start-up scripts. Despite your support contracts, your workers will be out on the Internet looking for help. Let them give it, too. Contribute help, bug fixes, workarounds, and improvements to make the software environment better for everyone, including your own company. By all means, check the incoming and outgoing code, but don’t stifle the natural impulse of many of your employees to contribute. They’ll be receiving help, too, and are likely to receive even more assistance if your company is known as one that also helps others.

Finally, let’s set aside the idea that OSS is all kids (small and overgrown) bashing out code in their parents’ basements. Some of your own programmers may be open source community members. (Find them now, before migration!) Visiting a site such as <http://slashdot.org> will give you a feel for the wide variety of community membership and their expertise, humor, irreverence, touchiness, and hot buttons. The comments you’ll find on these sites are frequently more valuable than the news stories because experts contribute them. Moderators also sort comments into five quality levels so you can read the most worthwhile at Level

5 or sample the full rainbow at lower levels. As for sites full of open source information, there are more portals than you can count; www.linuxlinks.com lists more than 28,000 sorted links.

Not Planning an Adequate Paid Support Program

Your support program will include your own employees and the community. It should also include vendors. Just as a vendor is required in proprietary systems to be on the hook for supporting your organization, the same is true in an open source environment. You still need the “throat to choke.”

Your capable staff will keep you from expecting an outside vendor to do every little thing; strong in-house support is more efficient and cost-effective. You may find that you have hidden open source resources on staff. The community will also help. But don't expect them to spend all their time downloading raw material from the Internet and tuning it to work together. There are plenty of vendors whose expertise and economies of scale enable them to do it better and cheaper than you can. This is because open source material, as it exists in

project repositories, can be pretty raw.

This doesn't mean that open source is a free-for-all with no controls; it's far from it. But keeping all your servers, applications, libraries and their different versions humming together requires serious effort. Upgrade havoc isn't less in open source than in proprietary systems; it's frequently greater. The question is how to insulate your organization against it. Several of the vendors mentioned cater to enterprises and have more deliberate upgrade schedules than those catering to bleeding-edge developers and hobbyists.

Conclusion

There are no simple answers to the tough problem of enterprise migration. There never have been and never will be. The best advice is to be pragmatic. Planning is still the most important migration tool and you still need the skills and processes you've honed over the years. Don't believe everything you hear. If a “new wave” evangelist can't explain in crisp, clear business terms why his or her approach is better, then it probably isn't.

As you plan your migration, look for creative ways to minimize transition costs.

Don't expect to migrate everything all at once; find the low-hanging fruit and implement those projects first. Apply the same hard-nosed scrutiny to open source licenses that you apply to products with proprietary licenses. Finally, give your support to the open source community, but remember that enterprise-class software will always need professionally managed, commercial-quality support services and you should be willing to pay for those. ●

Andy Astor is co-founder and CEO of EnterpriseDB, the open source database software company. Previously, he was vice president of standards and strategy at webMethods, and served on the Board of Directors of the Web Services Interoperability organization. He is a frequent speaker at industry conferences.
e-Mail: andya@enterprisedb.com
Website: www.enterprisedb.com



Donald K. Rosenberg is president of Stromian Technologies, a marketing consulting firm specializing in OEM software licensing and open source licensing and marketing issues. He is the author of *Open Source: The Unauthorized White Papers*, (Wiley), a book taking a business approach to open source software.
e-Mail: donr@stromian.com
Website: www.stromian.com

