

TechRepublic : A ZDNet Tech Community

Making a case for open source in your enterprise

by Guest Contributor | Apr 06, 2006 6:13:00 PM

Tags: Databases, Guest Contributor, open source, database, MySQL..., server

Takeaway: Bringing any new system into an established organization, especially when it is a concept like open source, is a matter of selling the idea.

By David HM Spector

All you read these days in the technology press are stories of open source (especially Linux) and how it's practically taking over the world. From Web servers and clusters through open source PBXs and MySQL Databases, it's everywhere.

Being a good technologist you probably even have Linux machines in your house that you use to keep your skills up to date. You see lots of potential, you've waited patiently for management to wake up and smell the open source coffee, and you even can think of some great test deployment opportunities but nothing seems to be happening.

How do you get open source systems in the door? Like starting any new venture, it's all in the sales pitch.

Sales pitch? Yes, a sales pitch. Bringing any new system into an established organization is a matter of selling the idea. And, in order for any new sale to happen, you have to have a solution to a problem that 1) works as well as whatever the customer (in this case your boss) is using now, 2) offers some benefit that the existing system doesn't have, such as lower price, or cost of operation, and 3) is supportable in the long term with a provable potential for a Return on Investment ("ROI") that is greater than what's in place now.

How do you know if open source is a good idea for your business?

Bringing open source systems—lets use Linux and the MySQL database as our straw-men—into an organization is (and should be) a lot harder than saying "but everyone else is doing it." Any competent manager would respond pretty much like your mom did when you tried that as a kid ("if everyone else were jumping off a bridge..."). The best way to make a convincing argument is to do

your homework and make saying "no" harder than saying "yes." And in any business that means appealing to the bottom line.

Unlike technologists, business people generally don't think about functionality as a first order of business, but rather about cold hard numbers. The fact that a technology is "cutting edge" or—even worse—"cool" is not an argument you want to use; cool is a term better suited to the marketing department than the CFO's office. When you want to get a business person to act, ask yourself: How does something we do affect the two most important letters in the Business-person's alphabet: "P" and "L": *Profit* and *Loss*.

If we stick to our example, and you wanted to replace your company's current proprietary server operating systems and database software you need to determine how much they cost to run now. How much more profit could be added to the bottom line if this expense—the annual cost of the OS and DB licenses—would be reduced if you could convince the company to make the switch?

You'll want to find out the following:

How much does the current OS cost per server? Are there per-user costs too?

How much is the base cost for the DB software?

Is your company paying extra for the number of concurrent connections to these databases?

Is your company limited in the number of databases or tables you can have? (Believe it or not, at least one major database company makes you pay by the field.)

How many DBAs support your databases, and how much are they paid? Can your current DBAs manage MySQL servers?

How much time and money will it cost to convert the data?

Are there costs associated with converting any of your customers that might come up because of this proposed switch?

Next, you'll want to figure out what I call The Three C's: Capability, Cost, and Coverage.

Compatibility: Can the new (Linux/MySQL) system handle the amount of data and the number of transactions the old one does? Can it handle the project growth you need?

Cost: Linux and MySQL are technically free (i.e., you can download them) but you don't want to be on the hook for 24x7 support, What will it cost to get the same (or better level) of support from one of the major vendors?

Coverage: What are the on-going costs to support this new regime? DBAs who need retraining and/or recertification? What about your SysAdmins? Is the new system so much easier to use that there's a possibility of a headcount reduction (Yes, it's unpleasant, but let's face it, it's a fact of life and always makes a system easier to sell)? If so, that's a big win.

Finally, if you're still in the running at this point, it's time to look around and see who else is using the open source systems you're suggesting. This is the one time the "but everyone else is using it" argument can be used; *especially* if they are your competitors or other comparable businesses in your industry. Extra points if you can show how much money they've saved.

Writing the business case

No matter if you are trying to bring in an open source database or some other piece of software, when you're writing the business case itself it pays to be short and to the point. Nobody likes to read long documents; keep it to 2-4 pages and cover the basics:

- Who are you?
- What's the problem/pain-point you've identified?
- What's the proposed solution?
- What are the advantages over the current solution?
- What's the impact (cost to implement, cost savings over the old system, etc)?
- What comparable businesses to yours have also benefited from doing this?
- What are the next steps?

If you must use PowerPoint, try to follow Guy Kawasaki's 10/20/30 rule: **10 slides**, fewer if you can swing it; the whole presentation should take no longer than **20 minutes**; and use nothing smaller than a **30 point font**. (Guy is one of the best VCs in the world and has great advice not only on startups but how to champion internal projects as well)

Which ever mechanism you choose (short paper or PowerPoint), the idea is to keep the information at the tactical level; you don't want to drown your audience in implementation details or bore them with a long presentation, and most importantly make sure that **you** with your in-depth knowledge of the situation you are trying to improve, are the center of attention. The paper or slides should set the stage for what **you** talk about. In PowerPoint in particular, keeping the font large keeps you from packing too much information on the slides.

Even once you've made a convincing business case, as an agent of change, you need to be prepared for any number of anti-open source arguments. Some of the most common ones (along with reasonable and provable rebuttals) are listed in **Table A**:

Table A

<u>Common anti-open source argument</u>	<u>Reasonable Response</u>
No "serious business" is using open source	Our competitors, such as <list of names> have seen a YY percent savings. Other large companies using this solution are...

<p>Won't the Gnu Public License (GPL) "force us" to give away our source code or products?</p>	<p>That's a common misconception; such a scenario would only apply if we tried to sell software based on the GPL'd code written by others without including that original GPL'd software with our distribution.</p>
<p>Open source software is buggy, our current vendor provides fixes.</p>	<p>Many studies, including ones cited here on CNET Networks show that, in general, opens source software has a defect rate that is far lower than that of traditional commercial software. The hallmark of most open source development efforts is the speed with which bug-fixes are applied, tested and distributed.</p> <p>By contrast, certain well-known OS vendors have had major security bugs linger for upwards of five years without being fixed, even across major platform upgrades.</p>
<p>Who'll support it?</p>	<p>There are X companies who offer 24x7 support of the product; the net cost of which is lower than our current costs. <list of comparable></p>
<p>What we have works fine, it will cost us more to switch</p>	<p>Be prepared to discuss a 1-, 3- and 5-year cost analyses of your proposed solution showing the savings.</p>

Remember, business people are not looking for new ways to *spend* money; they're looking for ways to **save** money.

Make sure you are engaging in a dialog; the best thing that can happen when you are pitching an idea is that you get a lot of questions and feedback. If there are valid objections to your proposal, don't be discouraged, take notes, and give whomever you're pitching a deadline by which you'll have answers to their concerns. And always follow up. Sometimes it takes more than one try to seal the deal.

The bottom line on making the case for open source software is just like making the case for any business enterprise: make the best pitch, and give solid business reasons to make the switch, and you are sure to get your project approved.

People who read this, also read...

Don't abandon .NET for open source
What does the Oracle-Sun news mean for enterprise CTOs?
VP of MySQL AB talks about open-source market trends
Open source grows beyond its roots
How scalable is Open Source?

Print/View all Posts

Comments on this article

Making your case Mark W. Kaelin  | 04/06/06

Well one place I worked at Tony Hopkinson | 04/06/06

no luck digitaltheory | 05/06/06

My Updates
My Contacts

Would you like your own dynamic Workspace on TechRepublic?

Take two minutes and set up a TechRepublic member profile.

[Sign In Now](#)

Would you like your own dynamic Workspace on TechRepublic?

Take two minutes and set up a TechRepublic member profile.

[Sign In Now](#)

Popular on CBS sites: [College Signing Day](#) | [March Madness](#) | [Lost](#) | [iPhone](#) | [Cell Phones](#) | [Video Game Reviews](#) | [Free Music](#)

[About CBS Interactive](#) | [Jobs](#) | [Advertise](#) | [Mobile](#) | [Site Map](#)

© 2010 CBS Interactive Inc. All rights reserved. | [Privacy Policy \(updated\)](#) | [Terms of Use](#)