Convincing a large company to use free software? [closed]

Asked 15 years, 11 months ago Modified 9 years, 11 months ago

Viewed 894 times

5

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 9 years ago.

Improve this question

I'm currently a developer at my first job right out of college. I work for a large company, and the trend I notice with them is that they tend to go with more expensive, closed source software about 99% of the time, while there are perfectly good open source alternatives that are available, most of which are vastly superior to their closed-source counterparts. For example, we use this absolutely awful source control software that cost a ton of money, while there are quite a few open source and/or free options that in my experience, albiet limited, are much better and offer basically the exact same functionality.

I guess my question is: **How would an experienced developer approach management about using more free software?**

It appears there is another question very similar to this that did not show up when I made this one: How can I convince IT that F/OSS software isn't evil?

EDIT: Just come clarification. I'm not necessarily trying to change the company's procedure, I'm looking for advice on how to approach management about the subject.

version-control open-source

Share

Improve this question

Follow

edited May 23, 2017 at 12:01

Community Bot

1 ● 1

asked Jan 14, 2009 at 20:50

Ryan Thames

3,204 ● 6 ● 34 ● 33

Why would you want to? Free != good in business necessarily. – GEOCHET Jan 14, 2009 at 20:54

stackoverflow.com/questions/375678/… – Greg Hurlman Jan 14, 2009 at 20:56

1 So you're right out of college and you already believe you can determine what is "vastly superior" for a business? Hmmm.... – ctacke Jan 14, 2009 at 20:56

Let me guess... Harvest Source Control? ;) – Andrew Rollings Jan 14, 2009 at 20:56

1 Just like free != bad business. – EBGreen Jan 14, 2009 at 20:56

18 Answers

Sorted by: Highest score (default)

13

✓

1. Start using it in small utilities and things which are throwaway and don't need management buyin. This can prove the worth of an open source solution and put a crack in the door for using it in other projects.
2. Present articles from trade magazines showing that other people are using the open source solution.
3. Go with products which have commercial support options, such as MySQL, which enterprises seem to have an easier time swallowing.

Share Improve this answer Follow

answered Jan 14, 2009 at 20:55

Arcane

1,228 ● 1 ● 8 ● 15

+1: Only success will convince anyone to make a change. If you're successful, and people ask why, you can discuss how F/OSS helped. Until then, you've got no evidence. – S.Lott Jan 14, 2009 at 21:09

+1: This is a good place to start. As a relatively inexperienced professional developer, it's easy to not think of things like evidence and lose your patience. Good advice.
– Ryan Thames Jan 14, 2009 at 21:27

12

Pick your battles carefully. Wait until they are suffering. If they are happy with what they have, they will not switch, no matter how much cheaper or superior the alternative is. You need to catch them while they're trying to think of ways to save money, or while they're disgusted with the problems of the current system.

Share Improve this answer Follow

answered Jan 14, 2009 at 20:57
Andru Luvisi
25.3k ● 6 ● 55 ● 66

7

Be very careful with what you refer to as free. There is a very large corpus of products that would be perfectly valid for a student to use without paying that an enterprise would have to pay for. Also never forget Total Cost of Ownership (TCO). A lot of relatively expensive software is expensive because you get things like configuration and help support for them whereas that may not be the case for free software.

Share Improve this answer Follow

answered Jan 14, 2009 at 20:59
EBGreen
37.6k ● 12 ● 68 ● 86

If an enterprise would have to pay for them, they are neither free according to the Free Software Foundation, or open source according to the Open Source Initiative. F/OS software support can often be purchased commercially. – David Thornley Jan 14, 2009 at 21:17

You are correct, but it is the rare college student (or recent student) that can make these distinctions. – EBGreen Jan 14, 2009 at 21:19

5

I think you are not asking the right question. To me, the challenge is to have my Big Corp to buy the BEST softwares for me, be it free softwares or not.

Paying for Windows or paying for Linux is not important (what is 100 $ for a Big Corp ?).

But having things done better is really important.

I think that your request to your boss should not be : "Hey, it's free and it's as good as XYZ, why are we using XYZ ?"

Why you boss would risk something trying the product you told when XYZ seems to be ok ?

It would be much more better to ask : "Hey, here is what I cannot do with XYZ : (your list). With my product, I would be able to do that and much more so fast than I would have a lot of spare time to test our own software !".

Small money is usually not a show stopper. Being able to work faster *in order to do much more testing* (or any other things that could help your boss have a better image) is definitely an excellent argument !

Best wishes, Sylvain.

Share Improve this answer Follow

answered Jan 14, 2009 at 21:32

Sylvain Rodrigue

4,946 ● 5 ● 57 ● 67

5

I work in a big company that has recently moved into being more enthusiastic about open source solutions. There have been a few big hurdles:

- **Customer won't support it** - we're defense contractors. We do almost nothing without customer say-so. As the customer's opinions have changed we've been able to change our architectures and tool usage. That said, there are still scenarios were open source is unacceptable and we don't use it.
- **No tech support = scary** - in several cases, it's been possible to make the point that open source may not have a single point of company tech support, but it does have huge communities that will support questios for free, and that there are consultants available as needed for the really hard stuff. Plus many, many releases of new versions for bug fixes. And, several competing expensive products have not been able to service tech support

needs. Being able to point to specific internal examples with long. well documented, histories of support problems, has been key.

- **Fear of security issues** - we had to develop a process for scrutinizing and controlling every peice of open source introduced. We've managed to find criteria for what we deem risky, versus what we deem relatively benign based on info-sec policy.
- **Fear of lawsuit** - Being large, and profitable, we fear lawsuits, we're great targets. We now have a process for the legal team to scrutinize every open source license. This has proved to be a win - since the legal team now has briefings on every major version of the typical open source licenses, and they can quickly review most stuff.
- **Version control** - fear that if those wacky developers can just download anything they like the world will self destruct. OK, well, practically speaking, the concept of "how do we know what's in a given product" - being able to show a FOSS version control process that is managed internally has been important.

It was definitely a slow process - small projects proved profitable and customers started encouraging it in proposals. That made it useful for executive management. It's helped that those that support it have been williing to put in some extra time to making the business case in terms of efficiency/cost savings, and have been willing to negotiate repeatedly with various parts of the corporate infrastructure.

Making open source work has taken the effort of IT, the info security folks, the legal team, the procurement team, and technical management. Knowing that before you talk to your manager is probably a key to success.

There's also some political savy - for a first project, don't encroach on any sacred cows - ie, those projects that may not be successful, but are high profile and owned by someone with lots of political power. Instead, choose some wacky new thing that isn't available right now and prove the cost savings in a way that is unlikely to provoke a defensive reaction.

Share Improve this answer Follow

answered Jan 14, 2009 at 21:36

bethlakshmi
4,609 ● 25 ● 45

▲

4

▼

When you try to introduce open source software to a big company (or even a small one, in many cases), the biggest counter-argument you're going to hear is "There's no tech support." Companies tend to be wary of using software that's supported by the community, because there's no guarantee (or in some cases, service agreement) that questions about the software will be answered within a reasonable time frame, or at all. In many cases, you can find a company that will provide support for the open-source package you want to use (for example, Red Hat does this for its Linux distribution, even though the contents of the distribution is mainly open source). Showing management a business entity that can support the software will often go a long way.

3

The other counter-argument to using open source software that I've heard the most often is "Open source software is buggy." This is a tough one; this opinion is pretty ingrained in some corporate cultures. Two possible responses are "The open-source community fixes bugs quickly" and "Since we have the source code, our engineers can fix bugs"--but that's often not what managers want to hear.

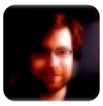
So, in essence, it depends on the company, their attitudes, and how much they trust you to make business-critical recommendations. I've used all of the arguments above with different levels of success in different companies.

Of course, in these economic times, the "free" part may go a long way. :-)

Share Improve this answer

Follow

answered Jan 14, 2009 at 20:58



MattK

10.3k ● 1 ● 33 ● 41

3

"Free software" doesn't necessarily mean your company is going to get software for free. Many successful open-source projects are also offered with licenses and services that cost real money and are geared to organizations that want or need to be assured of good support. MySQL is an example

Share Improve this answer

Follow

answered Jan 14, 2009 at 21:00

Scott Evernden

39.9k ● 15 ● 80 ● 84

2

The reason for a lot of big companies using closed software is that they can call support and the vendor will issue a hotfix, patch or cumulative update

Share Improve this answer

Follow

answered Jan 14, 2009 at 20:54

SQLMenace

135k ● 25 ● 211 ● 225

This doesn't answer the question, and commercial support like that is available for many major open-source software projects (RedHat, Ubuntu, MySQL, etc.) as well. – ceejayoz Jan 14, 2009 at 21:15

I can tell that you've had a lot better luck with software vendors than I have. My favorite was something we sent off to the vendor, and several months later got back, "Yes, it's a bug." – David Thornley Jan 14, 2009 at 21:18

or, "Yes, that's a known bug. We'll be fixing it in a future release." – Bruce Alderman Jan 14, 2009 at 21:32

Changing a large company's habits are often like turning an Oil tanker around... it takes a long time and uses a lot

2

of energy.

If the company were in the process of evaluating the purchase of new software for a specific task, Then I would make sure to write a concise opinion memo about why my choice is better.

If the software is something I would use personally and not a server product that multiple developers are forced to use, then I would just ask my manager to use it.

If the software is in place, does the job (even if I don't like the way it does it), i'd learn as much as I can about it to give it as much chance of work for me, or at least make my life easier. If it still sucks really bad, I probably wouldn't try to change it until it was time for the company to pay for an upgrade.

If the software works but is just annoying... I'd do as above, learning all there is to know about it just to make my life easier and then deal with it.

Share Improve this answer Follow

answered Jan 14, 2009 at 21:00

Jeff Martin

11k ● 7 ● 51 ● 76

Corporate or Enterprise Momentum is the term that I often use. – EBGreen Jan 14, 2009 at 21:03

You're probably right that the system you'd recommend is better than the one currently in place. But like some other

2

posters said, choose your battles, especially when this is your first job out in the real world. You may become expendable quickly.

It's not really so much a matter of what's better, even if your way IS better, it's a matter of the culture and the way things are done and the cost of switching. Even if, hypothetically, their system can be magically transported to your OSS system, with no loss of data, dates, records, or anything, you're still going to have people who say "I liked the old way better."

Remember: Experience is what you get when you don't get what you want. I know it may sound glamorous to be "the new guy who recommended a great new versioning system that everybody loved", but you also could just as easily become "that hotshot who insisted on a new versioning system that everybody hated." It's a much smarter career move to just play by the rules at least for a little while until you have some clout and can make some recommendations. In the meantime you may even learn why the old system is preferred, or learn to like it more the more you use it.

Share Improve this answer Follow

answered Jan 14, 2009 at 21:31

nerdabilly

1,248 ● 4 ● 15 ● 34

I know what you mean. It took us years to convince our managers that everything would be okay if we moved away from using Interbase (a commercial Relational DB)

2

to it's opensource counterpart Firebird. Mostly it was fear of no support that blocked the move. I think the factors which changed their mind were:

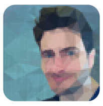
- tests showing better performance
- that there are companies that provide and charge for supporting the OS alternative
- constant pushing of the argument by passionate developers

I think cost savings would have played a part if our company were paying for the site licenses but in fact our customers were.

Share Improve this answer Follow

edited Jan 15, 2009 at 3:56

answered Jan 14, 2009 at 21:15



CodeAndCats

7,955 ● 9 ● 44 ● 60

2

I look at this question like this. I work with the .NET framework. I could ask my employer to migrate to PHP. This is a disadvantage to me, as well as my company, for many reasons. Let's start with the obvious.

1.) I know PHP, but can do much more, and a lot faster, with .NET. 2.) Paying for a service, usually ensures a better experience. The Visual Studio IDE is second to NONE when developing an application. 3.) I can develop an application much faster in VS than hard-coding PHP. 4.) This is the most important one. If I work with a big company, I want my programmers to develop my app faster, and I expect it to run faster. PHP (an example Open Source language) is fast, and reliable, but if I can spend the money, I'll deploy ASP.NET.

Basically, big business, or even small business, wants to spend their money, as long as it's for a good reason. Your best bet is to say, 'Hey, if you want to deploy ASP.NET (or whatever), send me for some training. Then I'll be able to develop OUR application to my best ability'.

Share Improve this answer Follow

answered Jan 15, 2009 at 4:18

BBetances

925 ● 1 ● 14 ● 23

▲

2

▼

not to sound totally cynical, but:

- an experienced developer probably would not approach management about something like this, unless he/she was already an expert with the open source package. Companies like to have a phone number to call and someone to blame when things don't work. Free open source packages do not provide this kind of 'accountability' (yes we know it's a joke, but management doesn't't)
- it is unlikely that management is going to listen to someone fresh out of college about any major purchasing or technology decision. You have to learn

the business and earn everyone's respect first. [sorry!]

Share Improve this answer Follow

answered Jan 15, 2009 at 4:36

Steven A. Lowe
61.1k ● 19 ● 135 ● 204

1

Same problem everywhere. Once an organization gets beyond a certain size (e.g., the Dunbar number) it starts to show a certain woodenheaded quality that will confound you. Lots of history, people, agendas that you aren't aware of. And getting everyone to agree on your solution is difficult.

Best to start locally. See if you can persuade your manager or PM to use SVN or CVS or GIT locally for a project and then get it to diffuse.

But that situation is true where I work as well. I use SVN locally for myself, but a commercial product for integrating with others.

Share Improve this answer Follow

answered Jan 14, 2009 at 20:55

duffymo
308k ● 46 ● 374 ● 565

1

Companies will use whatever will ultimately make them the most money. That means whatever software will make their employees more productive. If there is a particular piece of open source software you think they should use then when the time comes to purchase the software to do job X then as long as you can prove it will make the employees more productive and they are able to get reliable support just a phone call away as with commercial software then they will use it.

Share Improve this answer

Follow

answered Jan 14, 2009 at 20:59

Wayne Koorts

11.1k ● 13 ● 48 ● 72

What will make them the most money and what they THINK will make them the most money are often not the same thing. – ceejayoz Jan 14, 2009 at 21:16

1

Big companies need to hire support staff for stuff like that. When they purchase software from a company, they are guaranteed support with the contract. Open source projects can die off a lot easier, whereas a large software vendor can be held responsible for much greater periods of time.

Share Improve this answer

Follow

answered Jan 14, 2009 at 21:00

Cheese Daneish

1,030 ● 7 ● 15

1

Every company has a culture, and fighting the culture can be something of an uphill battle. But if you're willing to try:

1. you'll likely have more luck getting BSD and BSD-like projects approved (MIT license, Apache, Boost, etc.);

and it doesn't matter if most of the arguments against GPL and LGPL are mainly FUD

2. you should refer to the projects as "royalty-free"
3. you should make sure things are approved by somebody that can approve them (your direct manager) because putting the company in a bind -- especially when you're new -- (even if the "bind" is only in their head) is not conducive to long-term employment
4. you can probably go a long way by simply asking what the procedure is to choose a library or tool

Share Improve this answer Follow

answered Jan 14, 2009 at 21:10

Max Lybbert
20k ● 5 ● 49 ● 69

0

From a configuration management perspective, having developers add free software stuff willy nilly whenever they feel like is a serious PITA to manage.

I've worked at companies where you were allowed to do it whenever you wanted and others where you could never do it.

There's definitely a balance to be found but if you're in a larger company with multiple projects, you do have to keep in mind that each time you add a new 'tool' it complicates the build process.

Share Improve this answer

Follow

answered Jan 18, 2009 at 18:29

GeoffreyF67

11.1k ● 11 ● 50 ● 57