



"turning data into dollars"

Tom's Ten Data Tips – October 2009

Open Source

Open source is commonly associated with "free" software. This, however, is a misconception. Open source relates to "open standards", where source code and standards are laid in the public domain, free for all to see. The term "open source" was adopted in April 1998. Its fundamental philosophy is that if every developer contributes ideas, this "massive parallel process" will lead to the best, most robust applications. Indeed, some of the most vital applications (like running the NYSE) are based on open source.

Open source plays a role in many aspects around us, from business intelligence (BI) applications, to the foundation of Google's new smartphone G1. Java, Linux, Websphere/Apache, MySQL, are all based on open source, often commercially applied. And many, many others. JasperSoft, Pentaho, Talend, are some of the open source BI developments. A cooperative paradigm enables creativity and entrepreneurship in a massive parallel mode, analogous to how Apple has chosen to allow third parties to develop software for its iPhone.

1. Cheaper License Fees Should *Never* Drive Open Source Selection

A common motivation to consider open source solutions is the appeal of "free" or nominal license fees. However, open source "readiness" of the organization is a crucial prerequisite that needs to be determined first. To this end one might consider using an assessment instrument like Bernard Golden's Open Source Maturity Model (OSMM). He described this in his excellent book "Succeeding with Open Source."

Just like you don't embrace some useless tool because you got it for free, an organization will not adopt working with open source because it is "free." Isn't that also why most of the "goodies" you receive at tradeshow wind up in the garbage bin? One could argue that's where they belong, but leaving them at the tradeshow is actually far more efficient. Investigate needs first, then look for a tool. You don't go looking for needs that might be satisfied with a tool, not even if it comes "free."

2. Open Source Is Flexible

One of the highly desirable features of working with open source solutions is that by their very nature, they enforce standardization, yet applications are highly flexible. The “community” ensures that new developments (improvements or feature extensions) adhere to prevailing standards.

The business model is more flexible, because the structure of agreements typically involves no per-user or per-processor fees. This allows you to start very small (and inexpensive), yet at the same time retain the option of growing without bounds at little or no additional cost.

3. Open Source Pervades The Enterprise Market

Microsoft’s CodePlex foundation, and their cooperation with Novell (a “die-hard” Linux player) in November 2006 shows how even molochs like Microsoft need to acknowledge the open source tidal movement.

Oracle got a similar “grip” on MySQL which the market is vehemently opposing. After the Sun acquisition by Oracle, this made a lot of people nervous. So many companies’ back-ends rely on MySQL (formerly Sun), that this has drawn a lot of attention. This is also the main reason why the European Commission has held up the deal so far as they are afraid it will hamper competition in the open source market (US has approved the deal already).

4. Today’s Thought Leaders Are Tomorrow’s Laggards

There is unmistakable animosity against open source software. Some go so far as to say this is all a concocted conspiracy by the proprietary software establishment. Although giants like Microsoft or Oracle clearly have a vested interest in discounting the value of open source, there is more going on here.

Sun’s Java programming language is dominant, yet was *not* developed by any recognized force. MySQL is in wide use, yet is managed by a company of less than 200 employees. At the same time, corporate giants have launched new initiative with much aplomb that (much) later died a slow death. It is impossible to predict where the next innovation is coming from. There has never been as much investment in open source by venture capital firms than in the past year.

5. Common Sense Doesn't Always Make Sense

Many of the preconceptions against open source are based on "common sense." Let's have a look at some of these "obvious" arguments. One of them has to do with support. Because a vendor is backing a product, you are entitled to service and support. But what does that mean, in practice? An agent taking your call, is not what you pay for. An answer to your problem is. Large corporations are often slow to release patches and updates. And the open source community can prove remarkably responsive, *in particular to those who are also willing to "give back."*

Another argument against open source is that more expensive surely must mean better. Do you really need feature rich, enterprise-class solutions? This can be questioned for 80%+ of applications. Bigger isn't always better, and anyone who has tried to program an advanced VCR knows how all those "features" can get in the way of easy implementation. And they do.

6. Open Source ≠ Freeware

Open source is sometimes confused with freeware. Although most open source licensing agreements do not require any fees, or no more than a nominal amount, they are fundamentally different from freeware. Freeware is software that is distributed for free. Sometimes these are promotional test copies that stop working after a certain time, or that have limited functionality for demonstration purposes. And sometimes freeware behaves just like "normal" software, except that it's free. The difference is that with freeware, although the software *itself* is distributed, users do *not* get access to the source code. And therefore they cannot modify or improve it. They can only *use* it.

7. Staffing Makes All The Difference

One of the key determinants in assessing a fit for open source is whether you have staff in-house that is comfortable (and preferably experienced) working with your prospective (open source) product. And even *if* that expertise is available, you want to research how hard it would be to find replacements. The last thing you want is some half finished project with no qualified people to be found on the (your) market.

The good news is that open source development typically appeals to many bright programmers. And often the people who helped build the

product can be hired as well. They will have the strongest commitment to putting “their” product to the best possible use. They are also excellently placed to further develop or refine it.

8. Open Source Facilitates Piloting

With open source you can take a pragmatic approach because trying out solutions does *not* require major financial commitments. You don’t get locked in, and you can scale up at very limited cost. This all allows for very liberal piloting (rapid application development, RAD) which is known to be the most powerful method to surface requirements. Often only *after* a solution has been put to use do you learn what users *really* need.

9. Open Source Is Here To Stay

The market dynamics of software production clearly give open source an edge. After a software product has been created, the costs for reproducing is near zero. The lowest possible price point (in the long run) is determined by maintenance cost, which is lower for open source than for proprietary vendors: they cannot “push” costs for researching bugs or development to a community of essentially volunteers.

This dynamic pushes the prices down for *all* software that gets commoditized. That is one of the reasons why more “generic” products like Linux broke through faster than specialized “niche” products like open source CRM or ETL. The latter are newcomers on the market, and due to the special purpose nature, it will take longer for the market to be “forced” to mark down to their price level.

10. In The End It Is About Money

Money makes the world go ‘round, as they say. The number one reason for adopting open source still usually is “cost reduction.” Not much wrong with that. But there are other, compelling reasons to consider it. Corporate clients desperately try to avoid vendor lock-in. If you know a fair amount of customization will be required, again, open source is definitely an option (this tends to get prohibitively expensive with proprietary vendors).

In the market place, open source is frequently used *alongside* proprietary solutions (see also tip# 8). Most often to fill “gaps”, by building solutions that weren’t viable using vendor tools. These niche

applications help companies become “comfortable” and experienced running open source projects. And that, too, will help drive TCO further down. Which will propel further adoption...