

# Open Source Conference

By Mark Duncan, November 18, 2004, mark@askmar.com

On November 16, 2004, SDForum held its 2<sup>nd</sup> annual Open Source conference at the Network Meeting Center in Santa Clara, CA.

Attendance was up from the December 11, 2003 conference. Even though only 11 months have past since the last conference, there was a substantial change in many of the attitudes expressed:

2003	2004
Talking about having to directly sell and market to the various levels in a company.	Viral marketing had taken over. Open source companies have very lean marketing and sales organizations.
Concerned about SCO and its potential impact on Linux	SCO had about as much credibility as cold fusion, which also came from Utah!
Open source was a viable alternative.	Why are you using proprietary software?
Talking about being at the tipping point	We are past it.
Unclear how to value and charge for open source software.	Dual license and support are the primary sources of revenues.
People understood some of the advantages of open source.	Advantages were clearly articulated, and limitations were being defined.

*Author's note: We regard open source as a new distribution channel for software products. A software company is typically structured as a function of its channel. This means that a decision to change its channels is typically a bet your company decision. In the past, selling software to the enterprise market has required a direct, out-bound sales force. But with open source, sales can be reduced to a much smaller, in-bound team; and marketing can also be substantially reduced. In many open source companies, the majority of revenues will be spent on development.*

## **Open Source in the Enterprise: Practical Considerations**

The moderator was Jean Bozman, Analyst, IDC. The panel members were Eric Friedman, Infrastructure Architecture Team Lead, Wells Fargo; Guido van Rossum, Creator of Python, Elemental Security; Bruce Momjian, Principal Consultant, SRA / Founder PostgreSQL; and Julie Farris, CEO, Scalix

In the late 1990s, Linux bubbled up from IT administrators. Over the last two years, there has been a tectonic shift, from having to explain why we are using open source, to having to explain why we are buying a vendor product. The use of open source in products is no longer being questioned.

Part of the reason has come from companies like HP and IBM selling blade servers. Just as one adds a simple, reusable blade to a server, so the idea of simple, reusable components has rapidly become the whole ball of wax.

Open source software is a much smarter way of working across wide geographic areas and time zones.

Nobody expected open source software to go as far as it has. The market took off in 2000.

Everyone is being challenged by open source software; almost everything is being touched by it.

### **What is the Support Model for Open Source Software?**

Linux support is provided by the traditional relationships offered by IBM, HP, and Red Hat. Linux in running on commodity hardware, results in applications that don't know what they are on, and enables regarding a Linux server as an utility.

Now support for Apache requires a specific relationship with the vendor.

From an enterprise standpoint, support also means providing documentation and indemnification.

Open source gives you a choice of where you go for support. You have options on your coverage, level of support, cost, and customization.

### **What Types of Cultural Conflict are Occurring?**

What happens when all of your IBM mainframe guys retire in 10 years? This is driving the acceptance of using Linux on mainframes.

There are people who love command lines, and others who want a GUI (graphical user interface).

MIS departments don't want to be forced to "rent" software.

While there is a desire to have local variants, to "fork" them, this results in cutting oneself off from the main flow, and increases the cost of doing updates.

Linux is more modular and has more rigor, and higher quality due to its peer review.

### **What suggestions do you have for next steps?**

There are a lot of companies that don't understand open source.

While there is a higher initial learning curve with open source, once overcome, things become much easier and faster to implement using it.

The inherent competition between different open source contenders results in better products that become available faster.

### **Is open source being used as a club?**

Yes, open source software is causing conventional software prices to decrease.

Sourcelabs and Spikesource as seen as viable support options by the enterprise.

An issue is that there are considerable variations between vendors in what is considered to be a release, and what is considered to be certified.

Universities and colleges like open source software. They often cannot afford commercial software, and they like the peer review philosophy.

## Open Source Business Models

The moderator was Brian Grega, Business Development Director, OSDL. The panel members were Cliff Schmidt, Director Open Source Strategy, BEA; David McAllister, VP Product Marketing, Cassatt; Kevin Efrusy, Partner, Accel Partners; and Bob Bickel, VP Business Development, JBoss.

Will anyone get rich on open source software? Many will have profitless prosperity, but some will get rich. As a 1<sup>st</sup> derivative, we are seeing Ironport selling "sendmail in a box." As a 2<sup>nd</sup> derivative, Spikeserver is selling services.

JBoss is selling professional open source middleware, making it safe to be used, enabling large commercial companies to be able to use it.

One way of making money in open source is to sell support.

The four elements that we consider in formulating our business model to make an open source product is technology, community, customers, and the partners / ecosystem.

BEA is going from obtaining its revenues from licensing software to an open software model.

There is no point in innovating today unless it becomes widespread.

There is a lot of innovation that is possible.

There are lots of bits and pieces that form Linux, but all together, it is just an implementation of Unix.

Commodity hardware is fundamentally what has put the pressure on software prices.

What is there no equivalent in open source for VMware and VirtualPC? Well, this may still happen. But fundamentally, open source may not be appropriate for these products.

Linux is ideal for situations where you need billions of different device drivers, versus Sun where internal resources must be devoted to each driver. But there are certain areas where diverse set of people will not be able to reach consensus.

On the Apache Beehive project, it was interesting to compare the growth of developers and users. The developer list shot up initially, than flattened out. The user mailing list initially tracked the developers list, since developers were also signing up for this list. But than the user list shot way ahead of the developer list, as it started becoming adopted.

There is a fundamental allergic relation, where the usage of open source software causes us to go from a push to a pull model.

You want to develop a community of developers in order to avoid the perception of being proprietary.

Now when you look at how Linux is developed, there are 20 key guys who write most of the code, with another 800 or 900 people who do a few of the pieces, and billions of users.

Open source is not about innovation rather it is about the commoditization of software. Something like IBM VM took a lot of innovation and hard work — it would be difficult to do this using open source.

When we talk about open source vendors, the first tier are companies like Red Hat. The second tier is JBoss and other middleware companies. Using Java makes it easy to have loose coupling.

Where is software licensing headed? When you sell enterprise software, increasingly you find that the cost of sale was only sufficient to recover your sales and marketing cost. You don't make any revenues to pay for your R&D until you subsequently get maintenance and support fees.

But in an open source model, you have minimal sales and marketing costs. And since you can immediately charge for support, you obtain money to pay for your R&D. And when you charge for support, the cost is based upon the number of sites that the customer has.

Open source has lower development costs because it becomes distributed globally. Viral marketing promotes the product. You no longer have an outside sales force with travel and entertainment expenses, since all sales are done internally.

Often product innovation starts at the university, and is then adopted by the open source community. Grid computing is an example.

It is important to recognize that open source code becomes separate from a company. A community self governs, so if a company disappears, it is not disenfranchised, but can instead reform and continue.

## Vendor Perspectives

The moderator was Fima Katz, CEO, Exadel. The panel members were Stephan Vandor, Architect Linux Technology Development, Computer Associates; Jeffrey Wade, WW Marketing Linux, HP; Matt Thompson, Dir Developer Outreach and Open Source Strategy, Sun Microsystems; and Deborah Magid, Dir Strategic Alliances, Venture Capital Group, IBM.

IBM began its software business in 1995 and opened up its APIs. It now obtains \$14B in revenues from software.

Hewlett-Packard has a strong relationship with Red Hat Linux, and its Linux offering is built around how it provides support.

Sun has been active in open source. It has contributed over 14 billion (million?) lines of code, only UC Berkeley has provided more code to the community. As of today, Sun just put up the Java source code!

Stephan Vandor observed that he doesn't have any friends at Computer Associates in his efforts to release code to the open source community. He noted that Ingres started as open source 30 years ago, was acquired by CA about 10 years ago, and was just released as open source.

It was noted that Hollywood provides a model of how producers, directors, actors, writers, editors, and stage hands, come together for a short time to create something of value. In a similar manner, you can view how programmers from all over the world, jointly contribute code to create a module of value.

### **How will open source affect us?**

At HP we are looking at how we can reduce costs. We are looking at open source alternatives, giving us a different view on how to build solutions via different layers.

At IBM, we see that it is now becoming reasonable to use components at different layers. This works best for things that can be a commodity, it stops working when you have a specific domain or market niche that you are addressing.

Sun notes that in China, the government sees open source as a way of making money. China is paying programmers to write open source code. Fundamentally, open source is very different in countries like China, Philippines, India, and Brazil, than it is in the United States.

But there are things that are difficult to do in open source, where it is difficult to get consensus from a large number of people. It would be difficult to implement the threading and containers provided in Solaris, by Linux.

Sun is having success in China. Its initial 20,000 unit Java desktop in China order has already been increased to 100,000 units.

Computer Associates noted an IDC study that indicated that Linux drives about \$250M of its business.

At IMB, we pay our employees to contribute to open source. But for a startup, participating in open source is not proven.

Open source is about choice. Consider the ability to choose between Netbeans and Eclipse.

BEA by making Beehive open source was able to dramatically expand its customer base. You have to be able to move with the market in order to be successful.

At HP we wonder what will be the mature business model. Nobody knows where open source will end up.

## Keynote by Kim Polese, Spikesource

The lunch time keynote speakers was Kim Polese, CEO, Spikesource. Her topic was “The Renaissance of IT – the flourishing of innovation”

In the old model, software was supplied from a few vendors to many customers. Each vendor controlled a fixed habitat — it was more of a distribution system than a market. The objective was to capture, dominate, and hold customers. So even a vendor like Apple, was basically providing a better, furnished jail cell. Customers were forced into a buying mode that provided vendors with 80% margins.

But with open source, by making software a commodity, you create a thriving market. With its demand side model, customers now had a way of supplying themselves.

The Internet is what has enabled open source to grow. Each new open component adds to the net infrastructure, that anyone can use, no one owns, and anyone can improve.

So why isn't there chaos? In fact, products continually improve both in quality and flexibility. Quality in large part comes from having more eyes looking at the product. And the flexibility comes from enabling IT to do it yourself (DIY).

The quality, flexibility, and the ability to reuse components has driven the market acceptance of open source. The Apache server has a 70% share, MySQL has a 30% share, and Linux is growing at a compound annual growth rate of 33%.

The essence of open source is that everyone participates, no one rules. The independent customer becomes the most important player. This in turn, requires a new business model.

The Wall Street Journal ran an article, “Companies Do More With Less.” But fundamentally, it really was a story about open source. In the past, the market = what the vendor sold. But with open source, the market = what customers do.

Independent customers are giving vendors indigestion. Yet resistance is futile.

Take a look at the construction industry as an example. It loves commodities. Open source is the standard. Yet there is continual room for new innovations, that are gracefully and continually embraced by the industry.

Here are three examples:

- Verizon decided to take its existing pay phones in New York City, and make them into WiFi hot spots. Using open source, it was able to do this in a few weeks with a couple of programmers.

- Ernie Ball is a small business owner who went open source. He got raided by the Business Software Alliance when a former employee reported that he was using unlicensed software. He said, never again, threw out all his proprietary software and replaced it with open source, and never looked back.
- Ticketmaster compared the cost of open source at \$1M versus \$1.96M for using conventional software.

But open source has a high implementation cost. There are:

- Support issues
- Countless configurations don't work
- Legal and licensing issues
- No one throat to choke — who takes responsibility?

The result is that you can waste a lot of money on integration. Fundamentally, there is a need to assemble commodity software.

SpikeSource has been incubating for 17 months. Founded by Murugan Pal and Ray Lane, it now has 35 employees that are providing an automated system for assembling software.

SpikeSource interviewed CIOs, and found that they all wanted validation, integration, testing, support and service — as a total bundle.

What SpikeSource provides is a formalized core stack, knowledge base, support and maintenance — that integrates formal testing of building materials and methods — allowing the operating system, web server, application servers, SDA, and database components to be integrated — to provide file servers, mail servers, J2EE, portals, catalogs and web sites.

Initially, SpikeSource is implementing the 80/20 rule — by focusing on the 20% of the components, it can provide 80% of the need.

In essence, by applying the Henry Ford mass production / standardization of components to provide predefined, configured, tested and certified systems, we will cause the renaissance of IT.

How do we compare to IBM Globalservices? They throw bodies at the problem, whereas we provide automated procedures for testing and certification that make our people much more productive. We are entering into partnerships with companies such as JBoss and MySQL to grow and maintain our knowledge base.

## Success Stories

The moderator was Mitchell Kertzman, Partner, Hummer Winblad Venture. The panelist were Rajesh Setty, CEO, Cignex; John Roberts, Co-Founder and CEO, SugarCRM; Mike Olson, CEO, Sleepycat; and Edwin DeSouza, Sr. Director Product Marketing, MySQL.

At MySQL we have focused on the core capabilities that most customers need, and provided those capabilities with high performance and reliability.

At Cignex, we didn't have money for PR, so we went to open source distribution. We find that we get responses in hours, sometimes in minutes. Basing our software on open source means we don't have to start from scratch. There are 80,000 components, we were fortunate in having picked the right stacks.

Sleepy Cat provides an embedded database, non-SQL based, used at a low level. in sendmail, Apache, and 300 other applications. Founded in 1996, it has a dual license, both free and commercial. It is used by Amazon, Google, and Cisco. We have 1,600 downloads a day.

SugarCRM has had 70,000 downloads since April. It was funded by a top tier VC. We went with Lamplight instead of J2EE, since this is what the market wanted. Open source provides us with a level playing field.

At SugarCRM, we went with the Mozilla, dual license model. In order to use SourceForge, we were forced to be pure open source. Our free software is designed for medium sized businesses, but is upwardly compatible with the enterprise software that requires a commercial license, and that cannot be redistributed.

At Sleepy Cat, we underpriced in order to get people's attention. You can buy our package outright for \$150,000 or do a royalty. Our service plan cost is 20%, and we tack on the maintenance fee from the very beginning.

MySQL is doubling in revenues, with a big chunk coming from OEMs such as BMC and SAS. Large corporate accounts such as the Bank of America are coming to us to establish a support relationship, after discovering that many critical applications have been implemented via the back door. We are having 35,000 downloads a day. Google and Yahoo use MySQL to powers their ad pages. We picked the GPL as the most aggressive license.

### Competitive Arena

At MySQL we are staying focused on being the Honda Accord — providing quality, reliability, and performance. We are deliberately avoiding being a Cadillac. We find that for 90% of our customers, we solve their problems. It is a bit easier for us to provide legal indemnification because we own all our own code.

At Cignex we are in the service business. We find that big system integrators still avoid open source. As a result, we are starting to get million dollar deals.

At Sleepy Cat, we have a very good product, whereas the commercial products of a lot of our customers suck in comparison.

SugarCRM is the youngest open source company. People trust us to listen to them on where to take the product. We compete most often with Salesforce, followed by Siebel. Our product is bought, not sold. Software gets downloaded, and it is totally transparent to us. About 85% of our revenue goes to R&D.

At Sleepy Cat, we bring huge benefits to the market. But sometimes you have to wave flags so they know where you are.

MySQL observed that your product has to rock, only the best of breed succeeds.

The market is wide open, there are so many holes.

You need to understand the legal issues — and use them to your advantage. Who are your customers? They tend to be sophisticated. Realize that you will leave some money on the table. Listen to Tim O'Reilly and his lectures.

To succeed, you need a passion about the business and as a CEO, the ability to articulate and communicate it.

## Keynote by Andrew Morton, OSDL

Andrew Morton is the lead Linux kernel maintainer who works at OSDL.

There is a legacy infrastructure in which various functions have been implemented many times over 30 years or more. This infrastructure of modules has low margins and requires minimal maintenance, and is used by other products.

Why do companies pay their employees to contribute open source? It enables them to influence the direction the product takes, and it facilitates having internal resources that can support the software.

A newcomer with no track record faces considerable barriers to being able to participate and contribute. In order to establish trust, you have to spend the time initially going through their code, line by line.

Open source provides the ability to pool resources and amortize your costs, as well as to avoid getting locked into a vendor.

Typically a new feature will require at most, one or two people. But when you have thousands of different features aggregated together, it becomes a very large piece of work.

Open source is not appropriate for a large greenfield project that requires a lot of upfront planning.

We have been doing Linux with a 2 to 3 to 4 year development cycle. We stabilize version 2.4.3, then fork it to become version 2.5. We try to avoid changes to the kernel. We are at the point where massive, destabilizing changes no longer occur.

We have been improving our processes and tools to allow changes to occur faster. As a result, we are now on a 2 month development cycle, resulting in steady changes with each release, as opposed to a major leap.

All work is performed by email. We avoid instant messaging, and we never use telephone conferencing. Doing things by email results in documentation of changes, and encourages contributors who are not comfortable speaking English.

Reporting bugs by email allows many to communicate with many, and is pretty effective. But we also use Bugzilla for 1 to 1 correspondence on bugs. However, these bugs are reviewed, and will be bounced upward to the email lists if required.

Linux has formal testing performed at OSI and IBM, but these tests typically do not test fixes. It is our experience that random people on desktops doing random, unexpected stuff are the most effective way of finding bugs.

IBM, HP, SGI, Intel, Red Hat, and Souza are major contributors to Linux.

Linux now has 10 years of discussion, and thousands of people that are familiar with it.

We examined the claims made by SCO. None of these claims had any basis in truth. But it has promoted more formal documentation of who has touched the code. It is very difficult to integrate code from other Unix systems. Typically the drivers and file system have insufficient quality and the wrong style. You really need to write the code from scratch.

## Open Source Legal Issues

The moderator was Mark Radcliffe, General Partner, Gray Cary. The panel members were Steve Mutkoski, Corporate Attorney, Microsoft; Larry Rosen, Rosen Law; and Dave Marr, Sr. Counsel, Sun Microsystems.

When a lawyer says “interesting,” it means “ka-ching!”

The first thing that a lawyer will ask regarding open source is what is your business model, and what is your product? Why is it in the best interests of the company to open source?

There are over 50 different open source licenses. As one critic said about Mozart, “too many notes,” there are way too many different licenses. The problem with different licenses, is that when you are integrating many different portions of code, each with their own license, it becomes very difficult to sort out the nuances necessary to have an unified, common license.

When you modify and improve software, you can get involved in copyright, patent, and trade secret issues.

When we talk about the GPL license, be aware that section 7 that deals with patents, is a bit unclear, and has not been tested in the courts. As a result, no one is quite certain what it really means. But GPL doesn't get litigated because there is nothing to gain from it, and such efforts would be counterproductive. The exception of course is SCO.

Be aware, that with open source software, even if you think that you have reserved patent rights, you have still granted certain rights.

Open source has a tremendous weapon to fight patent litigation in its ability to search for prior art. There are companies such as Texas Instruments that obtain the bulk of their profits from its patent portfolio.

When you make something open source, even things like comments, may need to be reviewed for defamatory statements. There can be import / export considerations. And you may lose control over your technology.

Trademarks can get a bit tricky. Take Open SSH. It turns out there was a SSH Communications in Iceland, who was claiming the trademark on SSH.

But for all the talk about GPL, there are far more problems from proprietary licenses than from open source.

There is a lot of overreaching on copyright. Not everything you write can be copyrighted.

SDForum, 2<sup>nd</sup> Annual Open Source Conference  
November 16, 2004

Page 15

There is an interesting interaction between industry standards and open source. You should only accept contributions to the standard that meet open source criteria.

While an API can be protected by patent, it cannot be protected by copyright.

## Investing in Open Source

The moderator was Jason Wacha, General Counsel, Monta Vista. The panelists were Josh Stein, Associate, Draper Fischer; Larry Augustin, General Partner, Azure Capital; Pradeep Tagare, Strategic Investment Manager, Intel Capital; and Mohanjit Jolly, General Partner, Garage Technology Ventures.

We rarely see a deal where there isn't some open source involved, even if it is as little as running the Apache server.

Open source is a disruptive way of dealing with customers, it self-selects who your customers are.

In enterprise software, the initial revenue just pays for your sales and marketing costs, you only make money subsequently by selling support.

We believe that any major software company will have major open source competition, if not now, than in 12 months.

Traditionally, it has been very hard to go after the midmarket. But open source enables you to reach this market very efficiently.

SugarCRM has obtained 70,000 downloads over the past six months.

In the enterprise market, you are replacing \$1B in revenues with \$300 to \$400 in revenues. But having a lower price opens up the market, and by being able to reach the midmarket, the total revenues may be even greater than \$1B.

To get established, you tend to start with small customers and work your way up. Salesforce got traction from smaller customers, even though they are going after the enterprise now.

We think that \$4.5 million is sufficient for an open source company to break even, with maybe a \$500K seed investment to get it going.

Providing an integrated environment, SpikeSource is the first example, others will follow. We think that many companies will get rolled up as this occurs.

We expect that 80% of our exits will be M&A.

In the consumer space for open source, we are investing in a company doing 2D and 3D video editing that will have a subscription and means to trade content. Firefox with its browser is another example.

We see applications as the next open source wave.