

## **Open Source: How Will It Reshape the Enterprise Applications World?**

The last few years have seen open source software emerge into the technology mainstream. Enterprises are now deploying mission-critical applications using the Linux operating system, MySQL database, Apache web server and Jboss application server, just to name a few open source successes. Vendors such as Red Hat have proven that the open source business model works—they have created profitable businesses based on revenues from support, services and training. Now, open source is beginning to work its way “up the stack” into the applications space. Search the Internet or go to [www.sourceforge.net](http://www.sourceforge.net) and you’ll find numerous projects for ERP, CRM, project management, time tracking, professional services automation, learning management systems and more!

What’s the value of open source? Well, the first thing that many people think of with open source software is “free”—or low cost. That certainly is an important piece of the value proposition; but as open source moves into the applications realm, software licensing becomes a smaller proportion of total cost of ownership. For example, a recent Forrester study (July 2005) noted that companies will spend a total of US\$13 billion on CRM initiatives in 2005, but only 25 percent of that will go toward software licensing, with the balance going toward integration, administration and maintenance. And what about business process redesign and training?

In the enterprise applications space, two other components of the open source value proposition will play a more important role: quality and flexibility.

With applications that have large end user populations, quality issues can create very high costs for end-user support and lost productivity. And, a vendor’s schedule for fixing a particular bug may not meet the needs of the organization experiencing it. Is open source quality really better? Well, think about this:

- In a typical proprietary software company, a small quality assurance team is largely responsible for product quality. They have limited time before a product ships to find bugs. Once the product is out the door, the development team’s efforts to diagnose, reproduce and fix problems competes with new product development. Also, they may have difficulty reproducing problems because the customers can only provide limited information—they can’t look “behind the scenes” to report exactly where in the code the problem occurs.
- In the open source model, everyone has equal access to source code. Organizations using a particular product don’t just find bugs, but they have a better picture of the underlying problems. And, they have the power to fix the problems, any time. This open process also makes it faster to bring fixes into the main code line and issue higher-quality releases in shorter time periods.

Flexibility is also important, especially in applications spaces where enterprises have typically had to spend heavily to customize systems to fit with their business processes. Lack of visibility to the source code and limited touch points (APIs) have contributed to the time and expense of these projects. Projects often require special skills that are in short supply; I still see advertisements for SAP BAPI programmers. With open source, developers have many ways to do a particular customization. In addition, enterprises get more competitive prices for services since everyone is working from a level playing field.

So—what are the drawbacks? Will the open source model work as well in the enterprise applications space as it has for technology infrastructure?

1. *Will the conditions in applications markets provide open source developers with the same motivation to contribute to projects?*

Many open source projects started because a hacker, or group of hackers, experienced a problem that he or she wanted solved. This personal tie to the problem provided the motivation to develop, test and debug a piece of software and share it with the world. Will this same tie exist for a supply-chain management application, where the person experiencing the pain is in a business role? Maybe, maybe not. Perhaps the driver for developing open source applications will not be the hacker's own pain but the pain of seeing his or her organization paying high prices to deploy a product that has essentially become a commodity.

In *The Cathedral and the Bazaar*<sup>1</sup>, Eric Raymond, one of the chief open source evangelists, identifies the following factors that drive a software product category towards open source:

1. Reliability/ stability/ scalability are critical [*my translation: a mission-critical application*]
2. Correctness of design and implementation cannot readily be verified by means other than independent peer review [*this really applies to all enterprise software*]
3. The software is critical to the user's control of his/her business [*a business-critical application*]
4. The software establishes or enables a common computing and communications infrastructure
5. Key methods (or functional equivalents of them) are part of common engineering knowledge [*the features and functionality required for this application are well-established*]

To me, this points towards mature market categories where there are a lot of competitors with little differentiation.

---

<sup>1</sup> Eric Raymond, *The Cathedral and the Bazaar* (Sebastopol: O'Reilly & Associates, Inc., 1999, 2001)

2. *Will the open source community be able to design with usability for "the rest of us?"*

As enterprise software becomes more complex and reaches a larger population of users (usually infrequent ones), software usability has become a greater concern. After all, without usability, you don't have adoption; and without adoption, you don't achieve your return on investment. Will the open source community be able to innovate on usability in the same way that it can innovate on functionality? Maybe, maybe not. Eric Raymond admits this potential shortcoming among his peers:

*... While hackers can be very good at designing interfaces for other hackers, they tend to be poor at modeling the thought processes of the other 95% of the population well enough to write interfaces that J. Random End-User and his Aunt Tillie will pay to buy.<sup>2</sup>*

If the developer community isn't going to be able to design for usability, someone else would have to do it. It could be that:

- *End users could join the open source community and contribute their expertise on business processes.* Theoretically, they would be highly motivated to improve the application they're using and in turn increase their returns on investment. Consumer websites have been successful in finding volunteers to beta test their offerings. However, without any training in application design, end user feedback might only be useful for identifying problems, not creating solutions.
- *Usability experts could join the open source community and contribute their expertise on user-centric design.* However, the sociological dynamics that create a large community of passionate open-source developers don't necessarily apply to usability experts. For example, one of the reasons that open source works is that most of the world's software code is custom code for a single organization. The large pool of developers writing this code have many incentives to share innovations. On the other hand, a very small percentage of this custom code is user-facing, and there are many fewer usability experts. Also, a much larger proportion of these people work for proprietary software companies, where competitive pressures would make it hard to contribute to open source projects.
- *Commercial open source vendors could contribute usability expertise to their projects.* This seems like the most likely outcome to me, as long as it fits within the companies' profitability models. The vendors would be motivated to shepherd the project forward with usability in mind, because their support, services and training revenue streams depend on end users being able to use

---

<sup>2</sup> *ibid*

the applications. Linux already has some precedent for this model; part of the value-add that Red Hat provides its customers is in systems management and software update tools.

### *3. And, will it be profitable?*

Commercial open source has proven success in infrastructure. For example, Red Hat delivered a 25% return on sales in its most recent quarter, higher than SAP (19%) and Oracle (17%). Open source businesses in the application space will have to consider the larger end-user populations their offerings address and possibly the need to spearhead usability in their respective communities. If they build their business models right, they should be able to find a path to profitability.

#### *Implications for Traditional Enterprise Software Companies*

Open source does represent a threat to the profitability of enterprise software companies, particularly those relying on “cash cow” products in mature markets. One possible response to this threat is to be acquired—leaving customers with even less choice and more unmet needs. Here are some strategies that software executives should consider in the future:

1. *Differentiate on usability.* An application that end users love to use will have higher switching costs than one that isn't meeting customers' ROI goals because people are reluctant to use it.
2. *Differentiate on vertical expertise.* Select one or two industries and become an expert on them. Build this expertise into products, services offerings and training.
3. *Consider open source as a part of the product lifecycle.* For example, Niku contributed Workbench, its mature desktop project scheduling application, to the open source community. This move allowed it to focus on its newer IT management and governance solutions. In June 2005, Computer Associates announced its intent to acquire Niku for \$350M.

New threats always bring new opportunities. I can't wait to see what happens.

*Linda Sonne is President of Giant Stride Marketing Group, a strategic marketing consulting firm delivering business and product strategy development, sales tools development, competitive analysis and product launches to improve the sales and marketing effectiveness of high technology firms. She can be reached at [linda@lindasonne.com](mailto:linda@lindasonne.com).*