

JBOSS: THE EVOLUTION OF PROFESSIONAL OPEN SOURCE SOFTWARE¹

Richard T. Watson
University of Georgia

Donald Wynn
University of Georgia

Marie-Claude Boudreau
University of Georgia

Executive Summary

Over the past five years, JBoss has evolved from a failed dot.com-era startup to the market leader in the J2EE application server market. In the process, it has developed a business model based on the notion of Professional Open Source (POS) and built an ecosystem to support company growth.² This article uses two established frameworks to describe the growth of JBoss and its associated ecosystem. It also explores the four strategic risks that face every firm: demand risk, innovation risk, inefficiency risk, and scale risk.

We believe the emergence of POS has eight lessons for IS leaders: (1) the cost of running an IS unit will decline, (2) software innovation and quality should improve, (3) benign POS monopolies might emerge, (4) IS units might become less dependent on a single POS support provider, (5) POS will escalate, (6) IS leaders will experience pressures from outside the IS unit to adopt POS, (7) IS units will develop an open source strategy, and (8) traditional software companies will be forced to adapt.

WHAT IS PROFESSIONAL OPEN SOURCE?

Open source software would at first seem to be an unlikely foundation for a successful business. The notion of software developers working for free and giving away the results, while high in public spirit, appears rather low on all measures of private enterprise success. JBoss Inc., with its JBoss Enterprise Middleware System (JEMS), has created a profitable portfolio of open source products by applying the principles of professional open source (POS).

POS combines the benefits of open source (OS) with the development methodologies, support, and accountability expected from enterprise software vendors. The benefits derive from open code and open licenses, high-quality code, and innovation. Moreover, POS wraps enterprise-level service and assurances

around open source software to meet corporate expectations in such areas as quality assurance, education, and support services.

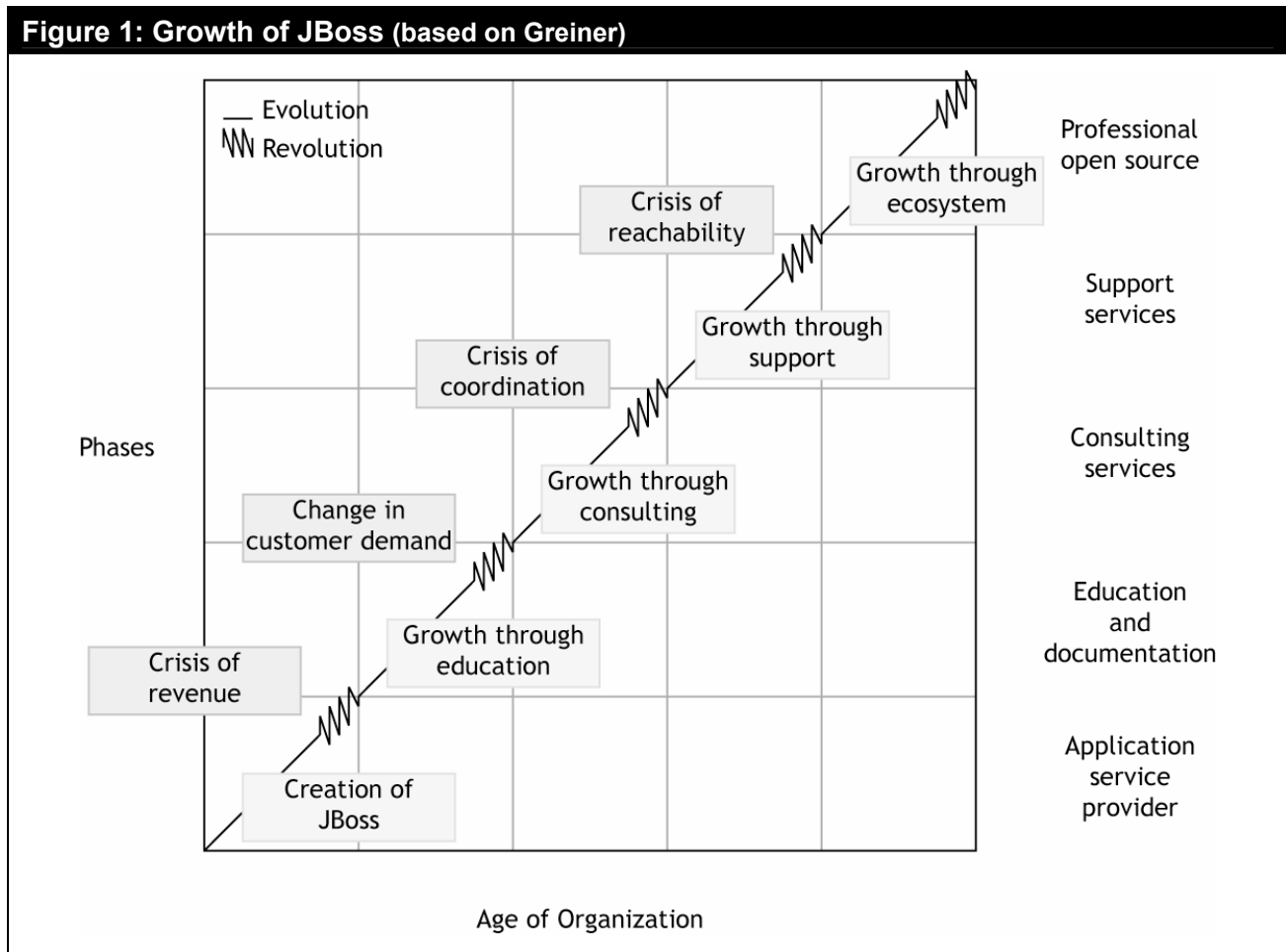
POS has evolved through several phases and, like all new business models, will continue to develop. JBoss is the thought leader and evangelist for POS. While building JBoss, CEO Marc Fleury and his executive team also created an ecosystem surrounding the company. In the following sections, we first use Greiner's³ framework to analyze the organizational growth of JBoss, and then use Moore's⁴ framework to describe the ecosystem's development. We elaborate issues raised by the JBoss case to discuss the impact of POS, including a strategic risk analysis, the distinctive features of POS, and lessons for IS leaders.

¹ Jack Rockart was the accepting Senior Editor for this article.

² The authors gratefully acknowledge the support of Marc Fleury, CEO of JBoss, for his willingness to assist us in this research.

³ Greiner, L.E. "Evolution and Revolution as Organizations Grow," *Harvard Business Review* (50:4) 1972, pp 37-45; and Greiner, L.E. "Evolution and Revolution as Organizations Grow," *Harvard Business Review* (76:3) 1998, pp 55-60, 62-66, 68.

⁴ Moore, J.F. "Predators and Prey: A New Ecology of Competition," *Harvard Business Review* (71:3) 1993, pp 75-86.



THE EVOLUTION AND REVOLUTION OF JBOSS

In the development of JBoss, the following phases stand out (see Figure 1):

1. Creation
2. Education and documentation
3. Consulting services
4. Support services
5. Developing the POS concept and scaling the business by growing an ecosystem

Greiner's framework envisions organizational growth as a series of growth spurts (evolution) and crises (revolution). Each phase of growth eventually faces a crisis that forces an organization to rethink its business model and organizational focus, which, when successfully resolved, establishes the next growth cycle. Greiner's original work, over three decades old, focused on the growth of industrial and consumer goods companies. Thus, his original model does not fit most of today's Internet-based organizations. Nevertheless,

the underlying notions of phases of evolution and revolution still hold, and his model can be deployed to represent the growth of organizations such as JBoss.

Phase 1: Creation of JBoss

The predecessor to JBoss was founded in 1999 with an Application Service Provider (ASP) business model. The goal was to develop software to allow companies to build their Java applications to run on an external application server instead of managing the server in-house. Thus, firms could avoid server operational and scalability issues because these problems would be the concern of the ASP vendor. Customers were to pay a fee for building applications using the JBoss framework.

JBoss built an application server, based on the Java 2 Enterprise Engineering (J2EE) standard, which grew into the core product of the company today, the JBoss Application Server (JBoss AS). Unfortunately, as the product reached maturity, the tech bubble burst and the venture capital market evaporated. Although JBoss AS was generally well received, it did not generate revenues because it was free in accordance with open source principles. Thus, the original company folded

in November 2000 with a useful product but no revenue.

Phase 2: Growth through Education

Despite the initial business failure, enterprise developers started to use the server. In early 2001, some requested training and documentation. So, JBoss re-emerged as an education business. The first training session, held in Atlanta in March 2001, attracted 20 attendees, each paying \$3,000. Selling JBoss AS documentation, which was a controversial concept within the open source community at that time, generated extra revenue. The prevailing notion was that open source software and documentation were free, but JBoss was able to monetize this part of its product line by responding to its customers' requests.

These two revenue sources were close to the firm's core competency: knowledge of JBoss AS. By capitalizing on this expertise, JBoss generated sufficient revenues to build a positive cash-flow business and take on additional developers to meet the growing demand. Scott Stark, JBoss' co-founder and CTO, made a living working full-time for JBoss.

Education set JBoss up to evolve even further, because once customers had learned how to use JBoss AS, they wanted to build systems and thus needed assistance in doing so. Faced with this change in customer demand, JBoss revised its business model again.

Phase 3: Demand for Consulting

Customers started requesting advice on building Java applications deployed on JBoss AS. Consequently, JBoss began hiring programmers from the base of independent developers who had voluntarily contributed code to the latest versions of the application server. The new recruits evenly split their time between development and consulting. JBoss set up the contractual arrangement with the customer and assigned a developer, who received a share of the revenue generated from the engagement. The customer, in turn, benefited from the expertise of a developer with intimate technical knowledge of JBoss AS.

Unlike software licensing, documentation, or training, growing a consulting practice requires a linear growth in people. The marginal cost of another copy of documentation or an additional license is minimal. Relatively few instructors can provide large numbers of training classes. But to grow via consulting, JBoss had to recruit and manage personnel who could provide the quality of consulting services the customers demanded.

As the need for consulting services increased, managing the network of loosely affiliated developers became increasingly difficult. Fleury had no expertise in operating or scaling a consulting business. Also, consulting was travel-intensive, which heavily drained vital developer resources. In the end, several tier-two developers became unhappy and left to start work on Apache Geronimo,⁵ a competing open source J2EE application server. Although JBoss was profitable, and consulting revenues had grown to nearly 40 percent of total revenues, consulting was not easily scalable.

Fortunately, customers' needs changed. They had learned how to use JBoss AS to build applications, and now they needed to ensure continuous operation of these applications. They increasingly requested on-demand support for JBoss AS used in their production applications. This change in customer demand and the problems of managing a consulting business impelled JBoss to once again redesign its business.

Phase 4: Growth Through Support

Enterprises require stable applications and a robust infrastructure to operate mission-critical applications. When problems arise that are beyond the ability of in-house staff to solve, customers want a support firm that can quickly resolve issues and restore operations. Ideally, they want a single point of contact to diagnose the problem, resolve it, and restore full functionality. They do not want multiple vendors indulging in a blame-shifting charade. They want "one throat to choke."

To fulfill this new need, in late 2003, JBoss transformed itself into a predominantly support-based business. As such, its business model resembles many established proprietary software vendors. Customers can still use JBoss AS for free, but they can also purchase 24x7 access to support personnel and a guaranteed two-hour response time. The service contract also includes such items as indemnification, development support, and deployment assistance. JBoss hired a number of the core developers full-time to provide these services directly to customers.

JBoss' service-based business model has an important advantage over consulting: it is highly scalable. Whereas consulting scales linearly with people, support follows a logarithmic model. Because a support firm's costs are proportional to the log of the number of customers rather than the number of customers (as is the case with consulting), it can grow support faster than consulting.

⁵ <http://incubator.apache.org/projects/geronimo/>

In early 2004, JBoss raised \$10 million in venture capital funding from Matrix Partners, Accel Partners, Intel Capital, and in a later private round, Bain Capital. The money was not critical to funding growth, but it did strengthen the balance sheet. Furthermore, anointment by the venture capitalists gave JBoss the credibility it needed to attract talented management. In particular, JBoss sought executives with experience and knowledge with enterprise middleware customers. These experienced senior managers have played a key role in converting JBoss to a support business.

During this phase, JBoss coined the term “Professional Open Source,” which refers to its intention to produce and distribute software under an open source license while creating an environment where open source developers can earn a living providing professional support to customers. Other firms, such as MySQL and Sleepycat, use similar models to leverage their existing open source software products for commercial gain.

The support model provided stable recurring revenues for JBoss. In economic downturns, software firms usually find it more difficult to sell new product licenses due to shrinking IT budgets. Maintenance contracts, however, are essential to a customer’s operations. Thus, support revenues are less affected by economic conditions, software trends, and competitive thrusts.

In addition, as software firms mature, their maintenance revenues compound year over year. Thus, an increasing proportion of their revenues come from services rather than licenses. For instance, Siebel Systems’ share of total revenues from services and maintenance rose from about 5% in 1995 to 55% in 2002.⁶ Oracle’s services’ revenues went from 45% to 65% during that same time. JBoss was forced to adopt a service revenue model early because it did not have licensing revenue. As a result, it achieved financial maturity much earlier in its life cycle than proprietary firms.

As JBoss AS gained market acceptance, the market expanded beyond JBoss’ immediate growth capacity and reach, particularly in the areas of sales and marketing. JBoss needed to find a way to capture high-end enterprise customers that rely on systems integrators and remote markets, such as Japan, where infrastructure is difficult to establish quickly because of cultural and language differences. This need set up the company’s next evolution: developing and extending an ecosystem.

Phase 5: Growth Through an Ecosystem

In 2003, the JBoss Authorized Service Partners (JASP) program was launched. Through this program, system integrators and software vendors are certified by JBoss to provide JBoss AS adopters with consulting, integration, and support services. The certified service companies are the first contact for support; more difficult problems are escalated to JBoss to be resolved.

This arrangement yields several advantages. Customers can arrange support from JBoss directly or via a local JBoss certified vendor, with the assurance that the vendor can direct more difficult problems to JBoss. Service partners gain direct access to the knowledge and expertise of JBoss’ core development and support team. Partners can resell support as a value-added complement to their other offerings, such as a server preloaded with JBoss AS. In return, partners share a portion of their services’ revenues with JBoss.

Partnering allows JBoss to expand its reach. Partners provide and recommend JBoss products to their customers, extending JBoss to new markets and countries. Local JASP system integrators exist in about 20 countries, in addition to the worldwide services provided by Hewlett Packard, Novell, and Unisys. Thus, JBoss has expanded internationally through its partners at a faster rate than it could have done solely through internal growth.

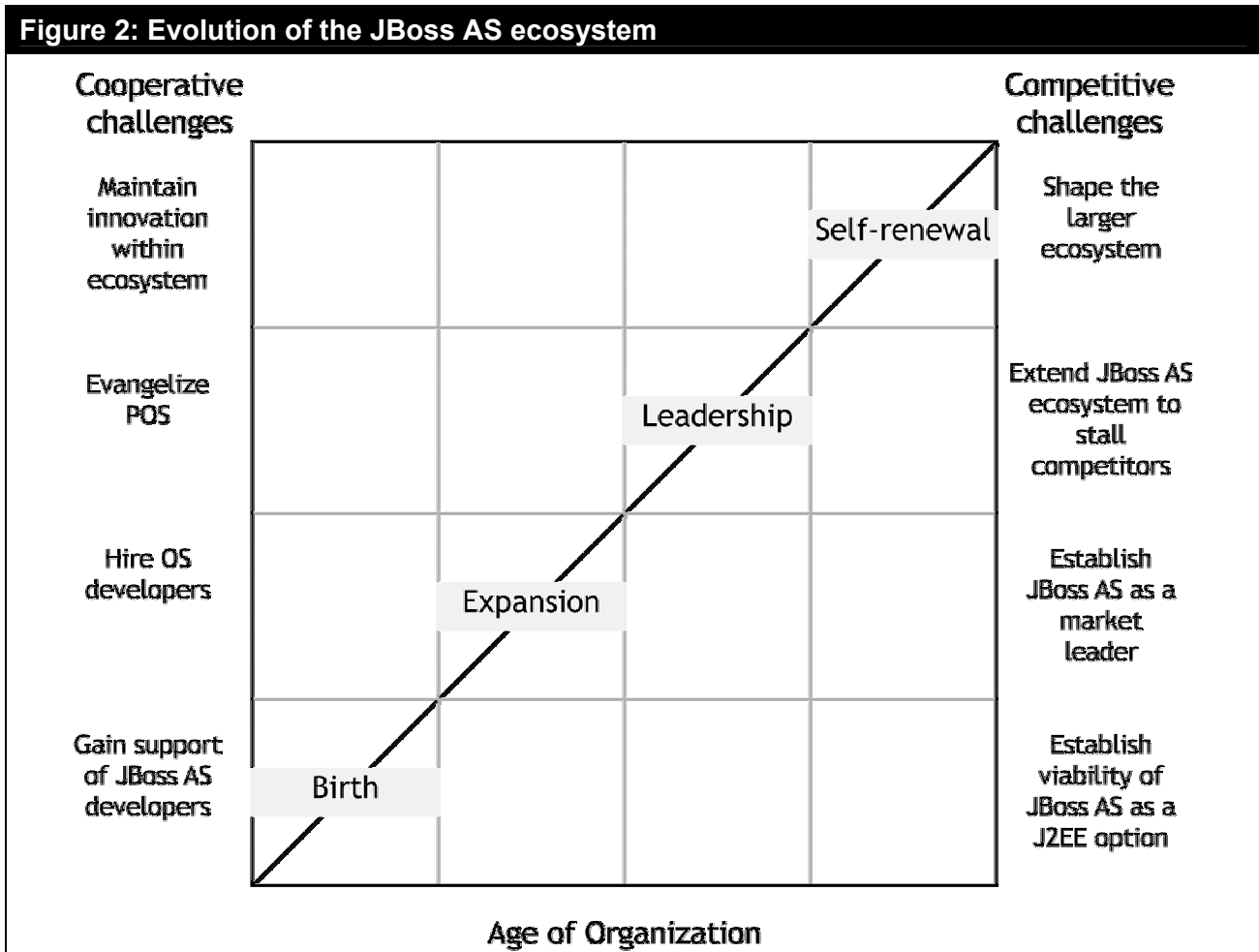
Partnerships allow JBoss to scale sales and marketing much faster than hiring such staff. Partners handle initial service requests and pass on only the more significant problems to JBoss. Thus, JBoss’ core developers need not handle the simpler, but time-consuming, issues. Finally, revenues are more predictable and stable under the service partnership model.

As an example of a partner relationship, HP announced in mid 2004 an open source middleware reference architecture based on Linux, JBoss AS, Apache, MySQL, and OpenLDAP.⁷ HP deploys this software stack on its server hardware for enterprise customers. After a sale, it provides implementation services and support to these customers, carrying JBoss products to a wider customer base than JBoss could achieve directly.

By mid 2005, JBoss, now five years old, had gained market leadership for three of its JEMS products. The POS model was gaining attention from industry observers, and Fleury appeared to have discovered a means of making OS a feasible business. This success

⁶ Cusumano, M., *The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times or Bad*, Free Press, NY, 2004, p. 37.

⁷ Fricke, P. “The JBoss Application Platform Strategy,” DH Brown Associates, Inc., July 13, 2004, www.jboss.org/pdf/dhbrown0704.pdf.



can be attributed not only to the creation of JBoss as a company but also to the simultaneous development of an ecosystem that gives JBoss a way to meet market demand growth rates.

EVOLUTION OF THE ECOSYSTEM

The JBoss partnering model exemplifies the need for firms to build an ecosystem or embed themselves in an existing ecosystem. Under the Greiner framework, JBoss is in the fifth growth phase, but growth does not occur without the existence of partners or stakeholders. JBoss needs customers, investors, advisors, employees, and so forth. Thus, part of Fleury’s entrepreneurial role is, ideally, to create a cooperative network centered on JBoss AS, or position JBoss AS in a successful cooperative network. Fleury initially chose the former strategy.

JBoss created a business ecosystem first around JBoss AS and then around JEMS. It gradually coalesced a collection of partners, open source projects, and customers into a structured community. The ecosystem

embraces four major areas: J2EE-based enterprise middleware, Java, the computer services industry, and the open source community. In its current growth phase, JBoss has to focus on building the ecosystem because this is its pathway to expansion. Unless the various entities in the ecosystem can each evolve successfully, then JBoss’ future is threatened.

An ecosystem has four stages of development,⁸ as illustrated in Figure 2, and these facilitate our understanding of the growth of the JBoss AS ecosystem.

Stage 1: Birth

The gestation stage required Fleury to provide leadership within the JBoss AS developer group. He was the emerging leader who showed OS developers that they could make a decent living as OS developers. By organizing seminars and selling documentation, he established a revenue stream. At the same time, he had to convince prospective customers that an OS product

⁸ Op. cit. Moore 1993.

was a viable alternative to commercial products for critical enterprise operations. Essentially, he had to sell a new business model to both those who would build the business (developers) and those who would sustain it (customers). Birth lasted for the first three phases of JBoss' evolution and revolution as an organization.

Stage 2: Expansion

The JBoss AS ecosystem took off when the senior executives recognized, during the fourth phase of JBoss' development, that service contracts were a sustainable source of profitability. The notion of POS emerged, and the OS developers and enterprise customers recognized the value of open source being professionally supported.

At this stage, competitors began to notice that JBoss had gained market share, threatening its proprietary software competitors. In addition, other OS developers recognized the potential of the POS model and the opportunity to compete with JBoss. Indeed, as mentioned earlier, some developers associated with JBoss left to create Geronimo. Ironically, surfacing the value proposition created greater competitive challenges for JBoss. The company has had to deal with these threats by clearly establishing that it is the leading OS application server and by providing support that meets enterprise standards of value and quality.

Stage 3: Leadership

Once the value proposition was established and the potential for a broad market recognized, JBoss has had to find a way to grow the ecosystem so that growth is not lost to competitors. Small software support firms can rarely scale rapidly if they depend on internal recruiting and assimilating skilled personnel to provide customer service. This is where leadership is critical in building an ecosystem that enables the support business to flourish. JBoss must find partners with the scale and expertise to handle service growth and bring them into the ecosystem.

A business model with few adherents is in danger of being treated as an oddity rather than as a sustainable mainstream offering. JBoss' value and reputation will be higher if it is one among many successful POS-centered ecosystems. Thus, within the OS community, Fleury has taken on the responsibility of promoting the POS model and convincing other OS developers to follow suit.⁹ To illustrate the robustness of the model, he has pulled other OS products into JBoss and announces JEMS (JBoss Enterprise Middleware Suite). Fleury has become the chief evangelist for POS and

⁹ See the JBoss blog for many examples, <http://www.jboss.org/jbossBlog/blog/>

has taken a key role in shaping the idea. At the first JBoss conference in Atlanta in 2005, he announced the formation of the JBoss Open Source Federation (JOSF), which "is a community of companies and sponsors created to bring together open source projects that share the common business model of Professional Open Source as well as integrate with the JBoss Enterprise Middleware System (JEMS)." JOSF is a clear example of active ecosystem leadership.

Stage 4: Self-renewal

JBoss is in the later stages of ecosystem leadership and early stages of self-renewal. The ecosystem is not completely established and stable, and there are still many international opportunities. Thus, JBoss must continue to provide ecosystem leadership. At the same time, JBoss is part of the larger Java ecosystem created by Sun Microsystems, and JBoss' future depends on the success of that ecosystem. Thus, self-renewal brings about a change in ecosystem focus. JBoss initially devoted attention to building an ecosystem around its own products. It now realizes, though, that its future is also linked to the success of the Java ecosystem.

JBoss will need to become a major player in the Java environment and has already taken actions to become more prominent. It has been elected to the Java Community Process executive committee, which means JBoss is one of the 15 organizations that determines the future of Java. It also participates in key specifications, such as EJB3¹⁰ and JBI.¹¹

CURRENT STATUS

Today, JBoss is somewhat similar to some proprietary software companies, albeit with a different pricing structure and development model. While firms such as Oracle depend on recurring license sales for a significant portion of their revenues, JBoss receives no licensing revenues.¹² Instead, it relies on recurring revenues from support provided directly to customers and through partners. The managerial core is similar to that of many corporations; it is internal. The development team, however, remains open to external developers.

POS, as implemented by JBoss, is a successful business innovation. JBoss AS has 34 percent of the J2EE application server market (compared to 33 percent for IBM and 27 percent for BEA Systems)¹³. Of the seven products in its JEMS suite (see Figure 3), three are

¹⁰ Enterprise Java Beans 3.0

¹¹ Java Business Integration.

¹² Other POS firms, such as MySQL, have a dual-licensing model and receive licensing as well as service revenues.

¹³ "But Wait, There's More," *The Four Hundred*, 14(3) 2005, www.itjungle.com/tfh/tfh011705-story05.html.

market leaders. JBoss is also a leader in the application server market, open source movement, and Java community.

JBoss' competitors are now reacting to the success of its POS strategy. In mid 2005, IBM announced the purchase of Gluecode, an early-stage open source rival in the application server market. IBM indicated that it intends to position Gluecode as a low-end complement to its WebSphere application server, charging for support in much the same way as JBoss. Interestingly, Gluecode uses the Apache Geronimo platform, which was founded largely by developers who defected from JBoss several years ago.

It is unclear how the market for application servers will change once other rival products are available and larger players begin to focus on competing head-on with JBoss. In preparation for more direct competition, JBoss may need to focus on strengthening its customer relationships, developing its brand, and expanding its middleware portfolio to secure its place in the enterprise software market.

THE STRATEGIC RISKS OF POS

POS enables a business to bridge the gap between a proprietary software firm and an open source community. When we analyze the strategic risks facing both, we believe POS has some important advantages for software development.

Every firm faces four strategic risks. Child¹⁴ identifies three: demand, innovation, and inefficiency risks. The fourth, scaling risk, we surfaced in our POS research. POS provides software firms with an effective approach to these four risks.

Demand risk

Demand risk is the risk of fluctuating demand or market collapse and may come about from changes in economic conditions, customer taste, or competitive thrusts. To meet this challenge, firms must determine the changes underway and revise their operations to meet new challenges.

Wal-Mart and Dell have altered the structure of the retailing and personal computer industries, respectively, through their low-cost strategies. Similarly, open source software typically drives the cost of software acquisition much lower than proprietary competitors. In most cases, the initial cost is driven to zero regardless of the number of servers deployed. Cost-driven IS departments are attracted by zero costs, as seen by the rapidly growing demand for JBoss AS.

The strategic plan for any software company must address basic issues such as the product/service mix, target audience, quality of revenues, and role within the community.¹⁵ One of the most critical strategic decisions a software company must make is determining how it will balance its revenues between sales of licenses and providing services to clients. Each distinctly influences operations and strategy. Companies that focus heavily on product licensing can generate high revenues, but must continue to produce new products or new versions of existing products to induce customers to buy more. Entry of a competitive POS firm threatens these traditional software companies that derive considerable revenue from product licensing.

Licensing revenues are also difficult to maintain during economic downturns or in the face of superior

Figure 3: JEMS Suite	
JBoss AS*	J2EE 1.4 certified application server platform
Apache Jakarta Tomcat*	JSP/Servlet Web application container
Hibernate*	Object/relational mapping (ORM) solution for Java environments.
JBoss Cache	Replicated and transactional cache to manage frequently accessed objects
JBoss jBPM	Workflow engine that enables coordination between disparate apps and services
JBoss Portal	Standards-based environment for hosting and serving a Portal's Web Interface
JBoss Eclipse IDE	Extends Eclipse with tools that enable programmers to develop and test new applications

* Market leader

¹⁴ Child, J. "Information Technology, Organizations, and the Response to Strategic Challenges," *California Management Review* (30:1) 1987, pp 33-50.

¹⁵ Op. cit. Cusumano, 2004.

substitutes. During these times, customers are prone to cut back (or eliminate) deployment of new or upgraded software. This drop in demand is especially of concern to software vendors whose existing products meet the needs of the majority of users. Additionally, the market may simply become saturated, irrespective of the existing economic climate. These potential fluctuations make pure product licensing strategies relatively risky. For new firms or for those who continually provide new, improved, or mandatory versions of their software, there is less demand risk.

By contrast, revenues from services (including support, maintenance, consulting, etc.) are more predictable and consistent. As firms develop a larger installed base, they can acquire more contractual service arrangements. In many ways, this revenue is similar to interest payments received by a bank.¹⁶ Even in less favorable economic conditions, these revenues are more stable and dependable than product licensing.

Product licenses have a higher margin than service contracts and are often a major portion of a software firm's revenue (See Figure 4). The incremental profits from licensing are typically quite high because the costs of distributing additional copies of software and documentation are low. Many young companies focus on licensing because the margins are so much higher. Mature software companies typically include more support in their business model because of the stability of service revenues and because renewing customers help service revenues to compound.

POS firms, such as JBoss, are not immune to demand risk, especially when the underlying features of their business model are imitated (e.g., IBM's purchase of Gluecode). Their pricing advantage is threatened and there is more competition. POS firms approach demand risk from different directions. JBoss relies strictly on services to generate revenues. MySQL, an open source database software vendor, relies on a hybrid dual-licensing scheme to derive both license and services revenues. In such a hybrid strategy, software vendors can take advantage of the stability of services revenues and the higher margins of product licenses.

Proprietary software firms with a high proportion of licensing revenues cannot easily convert to a POS model. They cannot turn off the tap of licensing revenues to switch to a services model without experiencing serious financial repercussions. The financial markets are likely to punish any firm that decides to abandon a significant proportion of its revenues, even though in the long run this might be the only viable strategy for competing with a POS firm in the same product space. By the time financial markets and shareholders accept the licensing business model as

invalid, the POS competitor might well have established itself as a market leader, which is precisely the situation that appears to face BEA as it competes with JBoss.

In summary, the majority of POS firms handle demand risk by pricing at zero and opting for the stability and compounding effects of service contracts. This strategy gives them an advantage, in terms of demand risk, over proprietary firms charging licensing fees.

Figure 4: Revenue percentages from licensing versus services for selected software firms¹⁷

Company	License Revenues (percent)	Services & Maintenance (percent)
BEA Systems	42.3	57.7
BMC Software	48.5	51.5
Business Objects	51.1	48.9
Cognos	44.0	56.0
Intuit	68.8	31.2
Novell	20.4	79.6
Oracle	34.9	65.1
Seibel Systems	36.4	63.6
Veritas	58.3	41.7
Wind River Systems	73.2	26.8
Average	47.8	52.2

Innovation Risk

Innovation risk is the peril of not innovating as well as your competitors. Software firms must meet the technological breakthroughs of their clients and others as well as keep pace with features their customers desire. Successfully dealing with innovation risk may be the only means to sustain success.¹⁸

The nature of open source is that the code is visible and available for all to observe, suggest modifications, and supply new code. When code is open, many developers can inspect it. Faults are detected more rapidly than when only a handful can review it. Furthermore, those who see the code can suggest improvements and submit code changes. Open source means computer science students around the world can be assigned to inspect and improve code.

¹⁶ Op. cit. Cusumano 2004.

¹⁷ Microsoft does not delineate between revenues from licenses and services.

¹⁸ Op. cit. Moore 1993.

JBoss and other POS firms have several distinct advantages over proprietary software because their source code is available on the Internet. First, skilled developers can detect and diagnose bugs and submit fixes. Second, the customer-focused nature of POS requires a wide-open dialog among stakeholders. By allowing new contributors, and not erecting barriers to entry or disregarding their efforts, a POS firm increases its accessible skills and intellectual capital.

JBoss is able to recruit developers worldwide from prior contributors to one of its projects. Generally, JBoss only hires people familiar with its software. In so doing, it can ensure that its core developers are capable of generating effective code immediately. It does not face the uncertainty that comes with hiring unproven coders.

Traditional OS firms have had a difficult time directing the efforts of developers because most are volunteers. However, POS firms employ key developers and have a managerial organization responsible for, among other things, communicating the current goals and managing the coordination needed to get desired outputs to the customers. JBoss' management team provides a roadmap and version plan not only to customers but also to developers. Thus, POS potentially can combine the innovative gains of open source with the coordination benefits of traditional management.

The openness of the source code, the willingness of customers to participate in code improvement, and the expertise of new hires are key factors mitigating innovation risk for POS firms.

Inefficiency Risk

Inefficiency risk is the inability to match competitors' unit costs. Strategic opportunities and market share can be lost if a firm cannot reduce its costs to match those of its competitors. For example, GM's health care costs for its workers, retirees, and their families are an estimated \$1,500 per car higher than those of Toyota, so GM has more difficulty competing in today's automobile market than Toyota.¹⁹

Inefficiency risk potentially offers POS firms the greatest advantage over their more traditional competitors. Indeed, traditional software companies usually bear higher costs for research and development (R&D), quality assurance, and sales and marketing. The lower costs of sales and marketing are perhaps the biggest differentiator between POS and traditional software firms.

R&D expenses for traditional software firms are largely composed of the salary, bonuses, and benefits of the employees who generate the source code for new products as well as enhancements for existing ones. Specific amounts vary by firm, but it is typically a significant percentage of revenues. For example, Microsoft, Oracle, and BEA Systems spent 16 percent, 14.2 percent, and 13.5 percent of revenues, respectively, on R&D in recent periods.²⁰

POS firms differ from both traditional software firms and open source communities in that most, but not all, of their R&D is conducted in-house. By hiring many of their developers, POS firms have R&D expenses comparable to traditional firms. However, they do have several advantages in their hiring practices, as discussed earlier. By recruiting developers from the existing community base, the training costs and quality risks for newly hired employees are potentially lower.

The sales and marketing expenses of traditional enterprise software firms arise from the existence of a sales staff that actively engages in selling directly to large IS departments. These expenses can run as high as 37 percent of a firm's revenues and 70 percent of new license revenues.²¹ Unfortunately for these firms, they have little way of eliminating these expenses without damaging their ability to convince customers to purchase and adopt new or upgraded products.

By comparison, POS firms typically have few or no active marketing campaigns. Instead, their marketing efforts are handled by a variety of actors, including aggregators (e.g. Red Hat, SuSe for Linux), partners (e.g. Novell, HP for JBoss or IBM for Apache), and even members of the community itself (e.g. the Mozilla community for Firefox).

POS firms depend on both the influence of the community and the low initial cost to "sell" their products. As more customers download and try a product, the potential revenue base is increased with little or no sales effort by the firm itself. The larger the user base, the higher the number of contributors and paying customers, albeit at a correspondingly lower rate. Marten Mickos, CEO of MySQL, stated in a presentation that for every 1,000 users that try its product, 10 users contribute in some fashion, and one becomes a paying customer.²² Considering that many of the popular POS applications generate several thousand

¹⁹ George Will, "What Ails GM," *Washington Post*, May 1, 2005.

²⁰ MSFT 10-Q, March 2005; ORCL 10-Q, March 2005; BEAS 10-K, January 2005.

²¹ BEAS, 10-K, Jan 2005.

²² Presentation, MySQL User Conference, Santa Clara, CA, April 20, 2005.

downloads per month, the numbers switching to paying customers can be quite reasonable.

The contributions made by new customers also differentiate POS firms from traditional software vendors. POS firms actively court and encourage their users (even their non-paying ones) to identify and submit bug reports; develop patches, extensions, and tools to supplement the firm's offerings; and evangelize to co-workers and employers. Such contributions are made voluntarily with little anticipation of financial compensation, which not only increases the information available to the firm but also keeps costs of QA, R&D, and marketing lower.

It is these low costs that are the most difficult for proprietary firms to match. Taken together, marketing costs can be as much as 51 percent of a firm's revenues (as in BEA Systems' case). Faced with rivals whose costs are significantly lower, incumbent software firms must either provide higher economic value or cut costs dramatically to continue to compete.

Once again, POS firms' open source code is a key factor in addressing a risk: inefficiency risk. Openness reduces software development costs and thereby reduces inefficiency risk.

Scaling Risk

Scaling risk occurs when a business cannot scale fast enough and efficiently enough to meet market growth. New businesses, and even some established ones, face this risk.²³ The result can be market share lost to competitors and imitators or disrupted economics of one's business model. Service businesses are difficult to scale because sales and service are often labor intensive. Software sales, particularly in the middleware sector, have traditionally relied on the high touch of a sales force. Solving customers' software problems often requires highly talented personnel with deep product knowledge.

The POS approach to sales and marketing is an effective scaling mechanism. POS firms still need to market, but they do it differently because they have separated the adoption decision from the purchase decision. Enterprises are encouraged to use and adopt the software, thereby providing a large pool of potential sales. Thus, market growth can be very rapid because product acquisition is simply a download, which helps explain why JBoss AS quickly gained a major market share.

Once adopted and incorporated into production services, many enterprises then need support to ensure high availability of their applications. At this point, POS firms can offer customers a number of services

or customized offerings to fit each customer's needs. From their experience actually using an application in their own environment, customers are often able to accurately specify the level of service they require.

As users expand their use of JEMS, they often realize they can gain greater value by having a relationship with the vendor that wrote the code. They decide they need to convert to being a JBoss customer. This customer-based "self-marketing" contributes to JBoss' low sales and marketing costs.

POS firms scale service by creating an ecosystem, as noted earlier. This is not surprising because they have their genesis in the OS ecosystem. POS executives are used to operating within a large ecosystem that works collectively to identify and solve problems. POS firms follow this model by partnering with professional service firms (e.g., HP, Unisys, and Novell) to multiply sales and service capability. By enabling the ecosystem to handle the less demanding service problems, the POS core firm can focus on the most intractable problems. In addition, by selecting partners with an established international presence and reputation, POS firms gain access to the world market.

A POS firm's marketing strategy centers on harvesting adopters. By tracking downloads, attendance at conferences and training sessions, bug reports, and so forth, the strategy gains considerable business intelligence on prospects for support contracts. Analysis of adopter activity identifies highly qualified leads for support sales. As a result, when done, marketing is highly targeted, more successful, and less costly per sale than the marketing traditional firms do to sell to enterprises.

Summarizing, POS firms have three approaches to address scaling risk: free downloading of applications, creating an ecosystem to support sales and support, and highly qualified sales leads.

Given the manner in which POS firms address the four strategic challenges, it is not surprising that their business model has attracted considerable interest. Analysis of these strategic risks helps surface key aspects of POS and identify takeaways for IS leaders.

THREE DISTINCTIVE FEATURES OF POS

POS has three distinguishing characteristics that are potentially applicable to other firms, particularly those in the information services business.

1. **Separation of product adoption and purchase.** Market growth is accelerated when customers can freely adopt without spending money or encountering restrictions.

²³ e.g., Apple's iPod.

2. **Seed and harvest marketing strategy.** POS recognizes that every user does not have to be a customer, but the more users, the more potential customers for the POS firm and the fewer customers for its competitors. Seed and harvest is a low-cost sales and marketing strategy.
3. **Dual growth.** POS entrepreneurs must simultaneously build their firm and their ecosystem, especially to mitigate scaling risk. In addition, as their firm grows, POS entrepreneurs might have to become active in enhancing the ecosystem in which the firm is immersed.

The combination of these three traits helps explain why JBoss and POS are attracting attention. They also hint at the underbelly of POS: the conversion of users to customers. Seeding and harvesting remains a low-cost marketing strategy when there are enough converts. If the conversion rate is too low, though, then marketing costs, no matter how low, will not be covered.

EIGHT LESSONS FOR IS LEADERS

The short history of POS provides two important classes of lessons: insights about how to build a POS firm, and revelations about the future of the software business. We focus on the latter because we dealt with building a POS firm in the discussion of JBoss' growth.

Every IS leader must be concerned with the future of the software industry because the viability of enterprise-level software suppliers directly affects an IS unit's performance. Let's assume the POS model spreads. If so, here are eight consequences, which are lessons for IS leaders.

1. **The cost of running an IS unit will decline.** Financial considerations are the principal reason to adopt POS products. The value proposition for customers is clear when POS produces high-quality products with a zero acquisition cost. For example, Sabre Holdings anticipates saving tens of millions of dollars a year by moving to open source.²⁴
2. **Software innovation and quality should improve** because of source code availability. There are more eyes to see the bugs and more people who can correct the code. As they say in the OS community, "With enough eyes, all bugs are shallow." Quality assurance should be superior when there is a motivated critical mass detecting and reporting errors. Furthermore, POS provides a mechanism for getting bug fixes promptly. In addition, a vast untapped development reservoir has not yet been fully integrated into the POS ecosystem: graduate computer science students. Imagine the innovation that will flow when their assignments include improving open source software. JBoss has already seen some payoff in this area, but this segment of its ecosystem is very immature.
3. **Benign POS monopolies might emerge** if certain POS products become market standards. If there are no license fees and the code is open, then a product can become an open standard like HTML and TCP/IP. A POS product cannot extract monopoly rents because there is no acquisition cost and the openness of the code means other firms can enter the support business.
4. **IS units might become less dependent on a single POS support provider** because open source allows others to enter the POS support business. The result is likely to be a more competitive market for POS support than for proprietary support. POS vendors will likely counter this trend by building their brands and emphasizing their depth of product knowledge from creating the code.
5. **POS will escalate**, leading to more POS products. Initially, most POS products supported the infrastructure layer, including middleware, database, and development tools (e.g., JBoss, MySQL, Sleepycat, Trolltech). Recently, however, commercial open source ventures have emerged at the application layer. For example, Compiere offers an open source ERP alternative. SugarCRM's suite of products targets improving customer relationship management. Pentaho has a business intelligence platform incorporating nine open source products. Medsphere offers integrated health care management software.
6. **IS leaders will experience pressures from outside the IS unit**, including senior management, to move to POS in much the same way outsourcing is often driven by external forces. In fact, some consulting companies have decided to tap both trends by offering services combining software development outsourcing based on open source products (e.g., Atlanta base SoftPros²⁵).
7. **IS units will develop an OS strategy**, with components including procedures for:

²⁴ Babcock, C. "Open Source, Part 2," *InformationWeek*, 2004, <http://www.informationweek.com/story/showArticle.jhtml?articleID=18402795&pgno=1>.

²⁵ www.softprosinc.com

- Determining risks and benefits of OS
- Determining total costs, including switching and ongoing costs
- Analyzing internal and external support resources
- Assessing the health of an OS product's ecosystem
- Assessing the impact on the IS unit's financial and operational performance

8. **Traditional software companies will be forced to adapt.** POS is already perceived as a sufficiently serious threat to force some traditional software companies to react, because POS adoptions are displacing their products. Proprietary Microsoft, for instance, is developing a "shared-source" alternative, IBM has acquired Gluecode, and Sun Microsystems has opened up its Java Enterprise Server (JES) server.

The POS model has a potentially large impact. IS leaders need to remain alert to its impact on their decisions.

CONCLUSION

Over the past five years, JBoss has grown through five phases to become the reference point for POS. The design of the POS business model and its key characteristics evolved during this period as JBoss navigated crises and reinvented the fundamentals of the firm. From a biological perspective, evolution can be seen in the gradual emergence of functionalities and behaviors that increase a species' survival prospects. Similarly, POS emerges over the five years as a set of business skills, procedures, and relationships that enable JBoss to thrive in its chosen habitat.

The continuing success of JBoss depends on how well it perceives and manages its next crises. POS has evolved through successful analysis of each crisis and successful solutions that have moved the company forward. JBoss might be on the verge of the next perturbation in its evolution as it begins to compete with IBM in the open source J2EE market. The POS model appears to work well when competing with firms marketing proprietary software. How well will it fare in direct competition with a firm with vast experience in marketing software and strong support services? JBoss might be the first POS firm to face this challenge because of its high market share. If its market penetration succeeds, MySQL could run into a similar challenge from Oracle, IBM, or Microsoft.

POS is an innovative business model that appears to work for the software business. It seems particularly

well suited for middleware, and we see no barriers to extending the model to other sectors. For example, Pentaho, which started in mid 2005, plans to follow the POS model for its business intelligence product. Pentaho's CEO states that after many years in the proprietary software business, he has decided his new venture will follow the POS model because of its competitive advantages.²⁶ In theory, the POS model could work for other information products because the cost of distribution is low, and customers seek additional value from support or subscription services.

POS is having an impact on business through the pioneers of this method in software development and licensing (e.g., JBoss, Sleepycat, TrollTech, and MySQL). It is a disruptive business model because it directly threatens software firms that rely on licenses for a large chunk of their profits. The disruption will be worthwhile for software consumers because it will lower costs and improve quality. They potentially gain from a "best of both worlds" combination, because POS combines the best attributes of traditional software and open source development.

We must keep in mind, though, that POS is an infant business model, about two years old, and having annoyed a few giant players because of market gains, it is now likely to face more concerted reactions from well-resourced and experienced competitors. Our ongoing research project will track the evolution of POS in this emerging and challenging environment; see Appendix 1.

New products, new markets, and new forms of production and distribution fuel free enterprises' creative destruction²⁷ and dislocate existing patterns of business. New corporate forms evolve to fit the changed competitive environment. Emerging from the Internet cauldron, POS is one of these new business forms and is a catalyst for further creative destruction. POS uses the Internet to lower the costs of production and distribution of software to create a new process for software creation. At the same time, POS is destroying the proprietary licensing model in some markets. IS leaders need to be cognizant of this change and why it is happening so that they can revise their strategic thinking and planning to take account of POS's impact on their IS unit's performance and the software industry as a whole.

²⁶ Personal conversation June 21 2005.

²⁷ Schumpeter, J. *Capitalism, Socialism, and Democracy*, Fakenham and Reading, London, UK, 1943..

ABOUT THE AUTHORS

Richard Watson

Richard Watson (rwatson@terry.uga.edu) is the J. Rex Fuqua Distinguished Chair for Internet Strategy and Director of the Center for Information Systems Leadership in the Terry College of Business, the University of Georgia. He has published in leading journals in several fields as well as authored books on data management and electronic commerce. His current research focuses primarily on electronic commerce and IS leadership. He has given invited seminars in more than 20 countries for companies and universities. He is the past president of AIS, a visiting professor at Agder University College, Norway and at Fudan University, China, and a consulting editor to John Wiley & Sons. He has been a co-chair of ICIS and a senior editor for *MIS Quarterly*.

Donald Wynn

Donald Wynn, Jr. (dewynn@uga.edu) is a PhD student in the MIS department of the Terry College of Business, the University of Georgia. He also holds an MBA from Middle Tennessee State University and a BS in Electrical Engineering from the University of Tennessee, Knoxville. Prior to his doctoral studies, he spent 15 years in telecommunications engineering, network management, and management information systems with BellSouth and EDS. His current research interests include open source software communities, ecosystems, business models, technology innovation, and information systems security. He has published in the *Journal of International Management* as well as several IS conference proceedings.

Marie-Claude Boudreau

Marie-Claude Boudreau (mcboudre@terry.uga.edu) is an assistant professor of MIS at the University of Georgia. She received a PhD degree in Computer Information Systems from Georgia State University, a Diplôme d'Enseignement Supérieur Spécialisé from l'École Supérieure des Affaires de Grenoble, and an MBA from l'Université Laval in Québec. She has conducted research on the organizational change induced by information technologies such as open source software and integrated software packages. She has authored articles published in many journals, including *Information Systems Research*, *MIS Quarterly*, *Journal of Management Information Systems*, *The Academy of Management Executive*, and *Information Technology & People*. Her teaching interests include design and management of databases, integrated software packages, and globalization of IS.

APPENDIX 1: SUPPORTING RESEARCH

This case study of JBoss is a product of a continuing study of open source software started in August 2003 by a research team at the Center for Information Systems Leadership at the University of Georgia. It draws on several interviews with Marc Fleury, co-founder and CEO of JBoss, and interviews with senior executives, usually the CEO, in leading open source firms following the POS model or a variation on it. The firms studied include MySQL, SleepyCat, TrollTech, Pentaho, MedSphere, Compiere, and Sourcefire. Others interviewed included an attorney specializing in open source software licensing, the CEO of a firm providing support for open source products, and the CTO of an Internet firm running almost entirely on open source software. We also attended several open source conferences and spoke to people familiar with the efforts of Brazilian federal and state governments to switch to open source software.