

The Open Source Service Provider Model

A Model for Commercial Providers of Open Source Application Software

Prepared by:

**Jack W. Gunther
Michael K. Underwood
Rose J. Thorsen**



Information Management Specialists, Inc. (*Informs*)

Introduction

Institutions of Higher Education are leaders and innovators in the movement toward an open source model for software creation, deployment, and use.

The excellent book, *The Success of Open Source* by Steven Weber (Harvard University Press, 2004), chronicles the history and leadership of the higher education community in the growth of Linux and Apache. Today, that leadership is again on display as the higher education community takes on the serious task of developing open source education software. Blessed with an abundance of developer talent and emboldened by the acceptance of core-technology based applications, like Linux and Apache; it was only a matter of time before the higher education community began to look at applying the open source model to application software. Sakai, an open source course management system, and Quali, an open source financial system, are the creations of a group of dedicated developers in higher education guided by an active community and a foundation that manages each product on behalf of the community.

This activity has not gone unnoticed by established, commercial providers of application software for higher education. Open source is emerging as a clear alternative to the traditional software license fee model. It is driven by a growing dissatisfaction among buyers and users with the significantly increasing acquisition and carrying costs, and the perceived inflexibility of vendor-licensed application software. Whether these perceptions are right or wrong is immaterial. Today, many buyers of application software are acting on these perceptions and actively considering available open source alternatives. These actions are in turn driving providers of application software to consider alternative business models to sustain their market share. Providers are being forced to address such basic questions as: What is the right mix of software development and support services? What are our key competencies? Where is the real value in our offerings to higher education? What alternatives are there to the traditional license fee model? Needless to say, these are not easy questions to ask much less answer. How commercial providers address these questions and respond to emerging market demands will go a long way toward shaping how application software is created, deployed, and used in higher education over the next ten years.

The purpose of this paper is to present some of the issues and considerations affecting both buyers and providers of application software for the higher education community. Further, a model of operation that can benefit both buyers and commercial providers of application software is presented.

Some Key Questions

How is application software different from technology management software?

Application software differs from technology management software (operating systems, data base management systems, web servers) in many ways. Application software is the visible information management tool that people engaged in specific work activities use to do their jobs. Application software must be developed with the work activities and business processes it supports in mind. It requires that the people using the software (typically not IT professionals) be expertly trained and fully supported in their use of the software. The value of application software is often measured in terms of the improved efficiency or effectiveness of the business processes. In the past, this software was developed by large, in-house teams of application programmers because of the unique business knowledge required to design and deploy useful information management tools. Over time, this style of application software development gave way to acquiring the software from companies that specialized in building software targeted to specific business processes (ERP, Student Services, and Course Management Systems).

Technology management software, on the other hand, is generally invisible to the people engaged in work activities. The software is developed with new information technologies and capabilities in mind. Developers of this software are IT specialists and experts. There is little need for users engaged in business processes to be trained in the software. However, it is essential that application software developers be knowledgeable and trained in the underlying technology so they can build their applications upon it. The value of technology management software is often measured in terms of improved price performance of technology components and on the introduction of new and advanced IT capabilities. This software has historically been developed and offered by IT vendors who could attract highly skilled IT professionals capable of this kind of development.

While there are open source initiatives underway for both application and technology management software, the fundamentally different nature of each suggests that different open source approaches may be required.

How do constituents benefit from open source application software?

Open source, or freely available, software is a type of business model used to manage the creation, deployment, and use of applications. Like all business models, it describes where the exchange of money for goods or services takes place. Traditionally, providers of application software charged a license and maintenance fee for the use of their software as a means to recover their significant development and on-going support costs. The open source business model turns this concept on its head. Open source software is protected by a number of open source licenses that dictate that the source code is free to acquire and modify. As such, in the world of open source, who benefits?

The most obvious beneficiary of open source application software is the buyer. Their acquisition costs go, essentially, to zero.

Users of application software may or may not benefit from an open source model. Their concern is not so much with the cost of acquisition as it is with the ability of the software tool to improve their work activities and business processes. The sense of fit between the software and the work is more about its functional richness, ease of customization, sustainability and simplicity of use. These attributes are assigned values that are independent of the business model used to market the product.

The providers of application software may or may not benefit from an open source business model. Most providers came into existence when organizations shifted from an application-build strategy to an application-buy strategy. Providers were able to concentrate the increasingly scarce application expertise and IT expertise necessary to create and offer superior application software. A software license/maintenance fees model offers the best chance of recovering these expensive costs. For an application software provider to give away their source code as open source, they would have to write-off their up-front development costs as a sunk cost. Unless there is a clear revenue-generating alternative, this is a very risky step to take.

Application developers may or may not benefit from an open source business model. In the days of in-house development, application experts were given little recognition and career development opportunities. Their professional prospects increased significantly by working for an application provider. The key motivators for application developers are recognition and reward for their business knowledge coupled with their IT expertise and the opportunity to do challenging work. Further, as the coding of software becomes more of a commodity, successful application developers will understand the importance of collaboration with end-users and will work toward developing these relationships. Much of the creative work done in the design phase of an application development effort comes as the result of these relationships.

What is the economic impact on buyers and providers?

While open source software can be acquired for free, its total cost of ownership (TCO) is not zero. There are the on-going operating and maintenance costs, upgrade and improvement costs, and training and support costs. TCO investments are especially high for application software because of the continuing need to improve business processes, generally through the introduction of new process support tools, and the on-going need to train the user and support personnel involved.

For the buyer, an acquisition cost of zero for application software certainly reduces the barrier to considering new business improvement initiatives. Open source shifts the consideration of value relative to cost to other elements of TCO. On-going operating and maintenance costs, as well as, training and support costs take on a greater consideration in the evaluation of any new initiative.

For the application provider, an open source business model only makes sense when an alternative source of revenue is identified that is sufficient to cover the loss of license and maintenance fees. For most commercial application software providers, this means evaluating whether a shift in revenue from software licenses to service and support offerings can be financially beneficial. Depending on how the software has been capitalized and the company's market strategy and position, this may or may not be possible to do.

What are the risks associated with open source application software?

Like any new thing, open source application software is surrounded by doubt and uncertainty.

For the buyer, it is very crucial that the software used to support its key business processes be acquired from a reliable, trusted, and sustainable source. As they say, "I need someone's throat to choke when something goes wrong." The risk to the institution's reputation from using software from an untrustworthy source is simply too great. The relationship between a buyer and a trusted provider is built on the confidence that the software will work as advertised, maintenance is readily available for inevitable bugs, on-going support is ample and available, and new capabilities can be requested and delivered in a timely manner. Open source application software must meet these conditions of trust and confidence before buyers will accept it.

For the provider, the loss of control of the software represents a substantial risk. Anyone can download the source code, use it, and modify it. The larger the market share the provider enjoys the greater the risk of loss of control. One of the key strategies large providers pursue is to keep all of their customers and clients on the "same page" through the use of controlled releases of the software and the selective maintenance support of such releases. A shift toward services and support as key TCO considerations represents a potential risk to the provider. If the trusted relationship is built on a strong competency in software development, then the provider may lack sufficient service-oriented competencies to maintain that trusted relationship in its new role as a service provider. Finally, if the alternative revenue sources do not develop after shifting to an open source business model, the provider incurs a serious financial risk.

What are the conditions required to develop a robust open source support community for application software?

First and foremost, widespread acceptance of open source application software in higher education will require that it be provided by a trusted source. The level of trust should be at least equivalent to that experienced in the marketplace, today.

The provider of open source application software must have proven expertise in the business processes its software supports. This suggests that the provider must value business knowledge at least as much as it values IT expertise. In addition, the provider must have demonstrated competencies as a service provider to the community of open source users.

Open Source Service Provider Model

A high level of collaboration among the buyers, users, and providers must exist. Because of the importance of business process knowledge, users must be recognized as full partners in the development, deployment, and use of the software. Further, these users must feel free to collaborate among themselves, and must possess a marketplace view of the business processes affected, far beyond the confines of their individual institution or campus.

The development and periodic improvement of the software must be sustainable. Buyers and users must believe there is a future for the software they commit to.

There must be compensation to the providers for their efforts expended. Providing support services, maintenance, and training are all critical efforts that must be adequately compensated.

A governance structure must be in place that will coordinate the allocation of resources and provide for ample collaboration and communication within the open source community. The ability to police contributions to assure a high-level of quality is crucial. This governance structure can take a number of forms: an independent, nonprofit foundation, a user group aligned to a specific software product, or some combination thereof.

Considerations for Developing an Open Source Provider Model

Today, much of the written discussion about desirable open source models in higher education centers around the notion of community source (see “Values of Community Source Development,” Lois Brooks, Campus Technology). This discussion explores many aspects and benefits of a hybrid model that combines the best of the “cathedral” and the “bazaar” as first described by Eric Raymond in his 1999 essay, “The Cathedral and the Bazaar.” While Brooks acknowledges a role for commercial providers, we believe that the discussion to date has understated the importance and value of their role and contribution, especially when considering application software. Further, we believe the community source model as currently discussed has limitations that will reduce its effectiveness in delivering the full promise of open source application software.

For open source application software to be viable in higher education, commercial providers will have to play a central role in the community of interest. We believe this is true for application software that supports any major business process in higher education but is particularly true for software that supports student services. The student services environment is constantly changing due to regulatory requirements demanded by the federal government and policy and procedural changes demanded by those managing the institution.

Consider the conditions that must exist for a robust open source support community as previously described.

Trust: Commercial providers with any sustained success in the marketplace have achieved the required level of trust that an emerging community of like-minded and dedicated individuals will have to build over time. That is not to say the relationship with the provider is always amicable. It isn't. But, you know they will be there tomorrow when you want to complain to them. The community source model relies on the collective support of multiple institutions agreeing to work together indefinitely (or at least as long as their presidents will support them) for the mutual benefit of each other. It requires a strong governance structure to provide an equivalent level of trust already achieved by commercial providers.

Business Process Knowledge and Expertise: The strength of established commercial providers is their accumulated knowledge and expertise in the business processes they support across the broad spectrum of institution types. In addition, they have acquired and continue to nurture the all-important IT expertise necessary to turn business requirements into well-functioning software. Further, these providers must support their software and have established well-functioning processes for on-going support and training. A similar level of business

Open Source Service Provider Model

knowledge and expertise as well as support and training will have to be developed by any community seeking to offer an open source application alternative. In the community source model, this expertise is borne by each participating member agreeing to commit and maintain some level of resources to the community. For the necessary IT and business analyst resource, the aggregated or accumulated knowledge may be the same or greater than that of the commercial provider but the cost to gather that knowledge into a useable force (collaboration and communication) is greater.

Collaboration: Commercial providers have established a variety of means to involve their buyers and users in the creation of their software. Most have established advisory panels or user groups and look to them for guidance in the design of future software enhancements. Communication and coordination within the user group is a cost underwritten by the commercial provider. Today, we see these providers fostering community involvement by taking advantage of the Internet to build message boards, blogs, and electronic newsletters. A group of like-minded professionals can also readily create a robust collaborative environment. It is argued that such a community has the advantage of being instilled with a sense of ownership as a result of the collaboration. Also, such a community can be less structured and, thus, better able to explore multiple options for process improvement concurrently. But if a commercial provider releases the code to its user community (e. g., contributing it to a foundation) the community has exactly the same degree of ownership as if the code was developed from scratch by the community. Finally, while each approach to collaboration can bring equivalent value, the cost of communication and collaboration is more expensive for member institutions to continuously carry without the participation of a commercial provider offering to support the administration of the communication and coordination process.

Sustainability: By virtue of being a business, established commercial providers offer a level of accountability to the buyer that a volunteer community cannot. Contracts for service are signed, service levels committed to, and liabilities created. This is no small requirement for those depending on the provider for the on-going development and administrative support of their application. Contractual relationships with clear lines of authority and responsibility provide stability and a platform for forward-thinking individuals to plan for business process improvements. A community source team will have to demonstrate its own long-term stability and accountability to the buyers and users without the use of signed contracts to enforce those liabilities.

Compensation: The individual contributors working for a commercial provider are paid for their work. As their skills and expertise improve their compensation increases. They are able to shape a career path around their work. On the other hand, volunteers in a community are not paid for their efforts except to the degree that their bosses decide to include such work in their overall responsibilities. There is no obvious career path associated with community work, although there is an opportunity to establish a reputation and gain visibility as an expert in certain business processes.

Governance: Governance is probably the single most important factor that will decide the success of any open source application software product. Commercial providers by definition govern their software. For licensed software, there is the concern that the mechanisms used to govern the software have significantly restricted the commercial provider's ability to respond to user needs in a timely and flexible manner. A major reason for creating a community source alternative is to improve the flexibility and timeliness of new software introductions. However, an equivalent governance structure must be created so that decisions about the future direction of software development can be made for the benefit of the community as a whole.

An Open Source Service Provider Model (OSSP)

We believe that the best chance for an open source application to succeed in higher education is when there is a commercial provider as an integral partner in the supporting community. We

Open Source Service Provider Model

believe that by releasing its software as open source, a commercial provider can enter into creative, new relationships with application buyers and users that will provide a greater level of satisfaction than can be achieved by a community source team acting on behalf of its users.

We call this new relationship the Open Source Service Provider (OSSP) model. The model builds on the thinking and experience gained from the community source model but extends it to include commercial providers in a central role.

In the OSSP model, the commercial provider donates its established software to the community (perhaps as a contribution to a foundation) as open source. This software must meet a high standard of flexibility and functional richness. It must strictly adhere to open standards and, ideally, be service-oriented either through a web-services design or by being Services Oriented Architecture (SOA) compliant. These conditions assure that the software is flexible so that many contributions can be accepted and implemented; and is interoperable so that the software is easy to install and integrate with other support systems. By donating their established software as open source, commercial providers are providing established, successful, trusted software as a building block for future development. The commercial provider must commit to continuing to collect requirements and provide fixes to the code in accordance with its community of users and under the direction of its governing body. This retains the trusted relationship the provider already enjoys with its clients and offers a sustainable environment for users to plan for the future.

For student services, we believe that it is particularly important that the software meet the conditions of open standards and interoperability. This assures that the highest degree of collaboration can take place among the many providers of student services application functionality, integrated through a single "common" infrastructure.

As open source service providers, the commercial software providers shift their business models toward service and support of the user community. Their value is in their ability to help users improve their business processes through extensive training and support, as well as by providing administrative support and the required infrastructure to coordinate the ongoing support, improvement and distribution of the software. They will charge for this support and commit to service levels as agreed to. This cost is represented in the shift from acquisition to support services in the buyers' TCO evaluation. Commercial providers will only prosper in this new model to the extent that they can demonstrate superior support and training services to the community.

Open source service providers will offer their services to the user community. The OSSP's services may include such activities as administrative support for the group, including event and meeting management and execution, publishing of the user group's communications, and coordinating the documentation and distribution of group members needed enhancements. Providers become the primary means for communicating ongoing requirements, releasing new functions, and enhancing the support and training offerings. The OSSP collaborates fully with the governing body responsible for the donated software. This governing body might also serve as the governing body for the user group.

A community in support of that foundation is critical. There may well be interest in developing the software along lines outside of the interest of the commercial provider. Additionally, there must be a means for the OSSP to coordinate software maintenance and enhancements with other development efforts. This balance is best achieved when a strong community governance body and the OSSP work together, in concert, for the benefit of all users.

In the OSSP model, the TCO of the software is optimized. Instead of a few members of the community bearing the brunt of the costs in terms of dollars and time to develop and maintain the application's sustainability and continued improvement, these costs are more evenly distributed across the entire community. Each institution has the option of deciding on its level of participation. The users, through their user group/foundation, each contribute a relatively small

Open Source Service Provider Model

amount of resources, in terms of dollars and time, and outsource the administration, configuration management, application code development, integration and distribution functions to the OSSP. This has the effect of reducing the TCO for each institution and the community as a whole. The OSSP will have already contributed significant resources by donating extensive application software which otherwise would have had to be developed by the community, as well as by contributing the physical facilities (hardware, software, internet access, and overhead) and administrative resources required to sustain the product on behalf of the community.

Summary

Higher Education finds itself in a leadership position in the growing movement toward open source application software models. Based on the substantial experience and insight gained from such initiatives as Sakai and Quali, the higher education community is advancing the thinking about what constitutes successful open source models. While much of the discussion today centers around the concept of community source, we believe that traditional application vendors must play a central role in any open source application effort and offer the Open Source Service Provider (OSSP) model as a guide to developing that role.

The community source model is necessary but not sufficient in two key areas. It lacks an appropriate mechanism to ensure the equitable distribution of costs to member institutions and lacks any real mechanism for assuring product sustainability. With an OSSP as part of the community, these challenges are fully addressed to the benefit of the community overall.

An application vendor operating as an open source service provider, aside from a very substantial contribution of code, assumes the burden of coordinating and communicating among the user community, providing training and education, providing on-going maintenance and enhancements, and incorporating community member developed enhancements, all as agreed to and directed by the community. They charge a service fee for assuming this responsibility and make commitments with liabilities for future efforts through a service level agreement. This relieves the community from having to deal with these costs themselves, reduces their overall TCO, and allows them to explore innovative uses of the code, confident that the means to implement these innovations continues to exist.

We believe that the OSSP model is especially important for a successful open source student services community. Student services by their nature are dynamic and constantly changing. An application vendor operating as an OSSP is ideally suited to assist the community in meeting the challenges of such a dynamic application environment.

The choice of which supplier of application code to select rests with each institution. Factors such as resources available, strategic direction and preferences, investments to date, and type of software implemented will guide this choice. Perhaps a traditional vendor relationship is best. Maybe software supplied as community source is best. Maybe it is best if the best of both an application provider and community source were available. We believe that with the OSSP model, it is possible to blend the best that an application software provider and a community source team have to offer and create a better result than can be achieved by either alone.