



The Open Source Software Model and a Business Case & for Open Source Software Implementation: Creating a Symbiotic Open Source Solution

Alan I Rea, Jr.

Business Information Systems Department
Western Michigan University, Haworth College of Business
1903 West Michigan Avenue, Kalamazoo, MI 49008
Phone: 269.387.4247, Fax: 269.387.5710
e-mail: rea@wmich.edu

INTRODUCTION

Open Source Software (OSS) was once purely the realm of geeks and techies. However, businesses depend daily on various Open Source-based technologies such as HTTP (Web) and SMTP (e-mail) to conduct transactions and maintain client contacts. Apache and sendmail (not to mention BIND, Linux, Perl, etc.) are the lifeblood of the Internet economy, yet programmers created this software without a traditional business software development model. Instead, it was developed, debugged, maintained, and distributed by volunteers. It's only recently that companies, such as Red Hat, have formed to profit from OSS.

It might stand to reason that businesses would also rely on OSS for their other needs, such as desktop applications, servers, and databases. However, in this arena businesses have long relied on commercial off-the-shelf software (COTS). Whether it's Microsoft or mainframe systems, COTS still has a stronghold in the corporate environment.

This trend is changing as more business and corporations are turning to OSS, such as Linux, to run servers. IBM is a long-time supporter of Linux on its architecture and Sun has recently joined in Linux support as well. Others have implemented Linux on the desktop. Red Hat Linux 8.x is specifically targeted for the desktop market. Initial news reports show that it's being well received by desktop users.

BUSINESS AND OPEN SOURCE

Why is this happening? Is OSS ready for "prime time"? Or have businesses found methods to implement OSS effectively? In 2001, MITRE released a business case study for OSS adoption for U.S. military program managers. MITRE argues that managers need to take five steps when considering OSS implementation:

1. Assess the supporting OSS developer community (e.g., Linux, Apache).
2. Examine the market.
3. Conduct a specific analysis of benefits and risks.
4. Compare the long-term costs.
5. Choose your strategy. (Kenwood, 2001)

Using the MITRE model as a benchmark, this research looks to create a new business case for OSS assessment and deployment in variegated business environments. The research develops a heuristic tool that businesses—no matter what their size or function—can use to determine what OSS implementation (full, partial, none) works best for their mission.

THE BUSINESS OF OPEN SOURCE

In the MITRE model, the first two steps involve an assessment of OSS offerings available to business. In the past, most OSS software was acquired

for low to no cost and then configured by each business, as it deemed necessary. While the front-end costs are low, the development and maintenance costs can become quite high. Most businesses cannot afford to train and re-train programmers with each new OSS implementation, nor can they afford to lose programmers' time that must be devoted to mission-critical tasks.

In order to fill the gaps between OSS purchase, implementation, and maintenance, businesses have begun to move to a service-oriented model. This is how OSS vendors such as Red Hat Linux make a profit. In some cases, the OSS service contract—also termed a "paid support network"—is less expensive than an outright COTS purchase. It remains to be seen if businesses will opt for a long-term service model.

However, does the service model look to be a viable component of OSS software development companies and business who use the OSS? While software is available for free as a download are businesses willing to at least purchase CDs and manuals for extremely reduced costs than COTS? Can OSS developers provide quality software for low cost and make a profit with various service models?

BUSINESS AND OSS SYMBIOSIS

Although businesses are sometimes ready to adopt OSS in terms of technological advantages and cost savings, the OSS model does conflict with competitiveness and business knowledge practices. Most businesses guard their intellectual property and software that embodies knowledge in order to remain competitive and/or show a profit by selling it. The OSS model requires it to be shared.

Some corporations, such as HP, have adopted a model of "Progressive Open Source" (POS). In POS, the OSS is layered into a three-tiered structure (Dinkelacker, et. al., 2002):

Inner Source: Open Source concepts, such as collaboration, sharing of code, and open modification are only available to employees. All collaboration takes place via the HP intranet protected by a firewall and other security measures.

Controlled Source: The source code is placed outside of the firewall, but this code is only shared with authorized partners on a necessary basis.

Open Source: The software is released to the Open Source community under a limited license. Only select pieces of software are selected for this level.

The HP model may be one that corporations adopt in order to use OSS yet retain a competitive advantage. This model can be effective if two items take place:

1. The POS model allows for enough sharing of OSS so that OSS developers can use the source code to improve existing applications and develop new offerings.
2. The POS model insures that all OSS develop of the source code remains open source and does not divert back into the inner or controlled source levels.

PRODUCT VERSUS SERVICE MODEL

The POS model has yet to be proven on a larger scale adoption, but it does attempt to bridge what is a large gap in synthesizing OSS and business—disparate culture. Most OSS is not developed for profit in the more traditional sense such as COTS. Whereas COTS has a high front-end cost, OSS can be downloaded for free or purchased on CDs for a marginal cost (compare the cost of Red Hat CDs versus Microsoft XP CDs).

This paradigm shift from product to service model suggests a change in software development. Instead of considering software a product, OSS is moving toward software development as a service. Paid support networks are emerging in OSS and are usually more cost-effective than a combination of COTS and service contracts.

This move to a service model in software development and implementation benefits both developer and business. As Feller and Fitzgerald (2000) note:

More important than sticker-price, OSS allows companies developing and implementing systems to share both the risks and long-term costs associated with a system. By shifting the locus of value from protecting “bits” of code to maximizing the gain from software use and platform development, OSS redefines software as an industry.

CONCLUSION

This research explores the changing OSS model of software development as seen in large OSS companies such as Red Hat Linux, traditionally COTS companies like IBM, and smaller operations offering OSS alternatives to popular applications, such as Ximian’s Evolution: a Microsoft Exchange program for Linux. In the research, the author looks forward to see what market niches OSS development companies can fill for businesses and if OSS can significantly displace COTS in business (or if it should).

Ultimately, the question we must answer is if a viable OSS symbiosis between OSS developers and vendors with businesses can be fostered and maintained. This research sheds light on the debate. While no definite conclusions can be determined without further study, some type of OSS symbiosis looks to be a part of the business climate in the foreseeable future.

SELECT WORKS

Dinkelacker, J., Garg, P., Miller, R., & Nelson, D. (2002, May 19-25). *Progressive Open Source*. Paper presented at the JCSE, Orlando, FL, USA.

Feller, J., & Fitzgerald, B. (2000). *A Framework Analysis of the Open Source Software Development Paradigm*. Paper presented at the 21st International Conference in Information Systems (ICIS 2000).

Hecker, F. (2000). *Setting Up Shop: The Business of Open-Source Software*. Retrieved October 4, 2002 from <http://www.hecker.org/writings/setting-up-shop.html>.

Kenwood, C. (2001). *A Business Case Study of Open Source Software*. Bedford, MA: The MITRE Corporation.

Open Source Initiative. (2002). *Open Source Case for Business*. Retrieved October 4, 2002 from http://www.opensource.org/advocacy/case_for_business.php.

Raymond, E. S. (2001). *The Cathedral & the Bazaar* (2 ed.). Sebastapol, CA: O’Reilly.

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/proceeding-paper/open-source-software-model-business/32208

Related Content

Application of Desktop Computing Technology Based on Cloud Computing

Kai Zhang (2021). *International Journal of Information Technologies and Systems Approach* (pp. 1-19).

www.irma-international.org/article/application-of-desktop-computing-technology-based-on-cloud-computing/278707

Internet Banking System in South Asian Countries: Preliminary Findings

Farrukh Amin (2009). *Utilizing Information Technology Systems Across Disciplines: Advancements in the Application of Computer Science* (pp. 222-242).

www.irma-international.org/chapter/internet-banking-system-south-asian/30728

Gene Editing Technology and Ethical Issues

Barbara Jane Holland (2021). *Encyclopedia of Information Science and Technology, Fifth Edition* (pp. 1952-1966).

www.irma-international.org/chapter/gene-editing-technology-and-ethical-issues/260321

Team Characteristics Moderating Effect on Software Project Completion Time

Niharika Dayyala, Kent A. Walstrom and Kallol K. Bagchi (2021). *International Journal of Information Technologies and Systems Approach* (pp. 174-191).

www.irma-international.org/article/team-characteristics-moderating-effect-on-software-project-completion-time/272765

A Novel Computerized Paleographical Method for Determining the Evolution of Graphemes

Gábor Hosszú (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 2017-2031).

www.irma-international.org/chapter/a-novel-computerized-paleographical-method-for-determining-the-evolution-of-graphemes/112609