Is Open Source Software the Future of Software Development?

Sohail Subhani, PhD
Professor of MIS
Business Administration Department
Winona State University
175 Mark Street
Winona, MN55987, USA

Abstract

Just a few years ago, it would have been difficult to envision that open source software applications could seriously threaten closed source or proprietary software industry. But as we look at the existing software development landscape, it becomes vividly clear that the open source software movement is gathering momentum and poses a serious challenge to the proprietary software industry. The infrastructure of the internet, such as mail servers, web servers, and FTP servers is largely based on open source applications. Apache servers, representing sixty percent of the web servers on the internet, is currently the number one web server; Send mail controls forty percent of the e-mail servers. My SQL's market share is growing faster than Microsoft's SQL server and Access databases. Even large technology companies such as IBM, Intel, and Sun have started to support open source development with the expectation of enhancing their hardware sales. In view of the evidence, it is hard to refute the fact that the open source philosophy will dominate the software development environment of the future. In order to assess the viability of the open source software development, this paper examines factors contributing to the growth of the open source movement and also looks at the adequacy of the open source business model.

Keywords: Open Source Software, Proprietary Software, Software Development

1 Introduction

In early 1970's, the source code for most computer software was open and readily available, which allowed programmers to refine and modify the program to fit their needs. But in 1980's as the source code became increasingly proprietary, programmers lost the flexibility of modifying the code to meet their requirements (Towle, McFarland and Keppler 2004).A programmer named Richard Stallman, frustrated with his inability to refine and change the proprietary operating system code, established Free Software Foundation in 1985 for the purpose of creating a non-proprietary UNIX-like operating system. This marked the beginning of the open source movement. The term open source software was officially coined in 1998 (Samoldas et al. 2004).In the literature, the term "free software", "open source software (OSS)", "free open source software (FOSS)" is synonymous and is used interchangeably.

Open source software refers to a type of license that provides access to the source code, allowing developers to modify the code to fit their needs or to fix errors (Kalina and Czyzycki 2005). Under the open license agreement, programmers or developers making major changes to an open source application are required to share those changes with the public (Cusumano 2004). Open source software could be used, studied, copied, distributed, and modified free of charge, but it is erroneous to think of it as having azero price tag. It can become expensive once you factor in the cost of customizing, installing, and training (McGrath 2004). To understand the concept of open source software as being free, one should think of 'free' as in ``free speech," not as in ``free beer" (Stallman 1992). 'Free' in the context of open source software refers to the freedom to copy, study, run, distribute, and modify the open source application.

The open source software development is driven by a single developer or group of developers who write the first version of the software and make it freely available over the internet for other developers to participate and contribute pieces of code. They are mainly volunteer programmers who are motivated by the creative challenge and self interest, willing to cooperate to keep the big companies from controlling the computing environment (Delong and Froomkin 2000). In contrast, proprietary software is developed by a team of designers, programmers, and project managers employed by software companies that are motivated by profit. Instead of sharing the code like the OSS community does, the objective of a software company is to protect the proprietary software so as to make a profitable distribution (Towle, McFarland and Keppler 2004).

In order for the OSS applications to survive the onslaught of the proprietary software developed by large commercial software companies, it (OSS) must have characteristics and a viable business model that will enable it to compete effectively with the proprietary software. The next section looks at the inherent characteristics of the OSS development environment that facilitates the open source movement, followed by a section that discusses the viability of the open source business model.

2 Characteristics of Open Source Software Development

Flexibility, the ability and the freedom to tailor the software to one's specific needs, is a valuable feature of the OSS, and a major force propelling the OSS movement. A single developer or a group of developers write the first version of the software and make the source code freely available to other developers and users, who are then free to copy, distribute, improve, and change the source code as they deem fit.The developers and users are encouraged to participate and contribute to the original source code, the only requirement being that any major changes to the source code must be made available to the public. This approach of free availability and modification of source code is in sharp contrast to the proprietary software approach which does not make its source code available and prohibits any changes to it.

Collaboration and sharing, the essence of open source software development, encourages participation from a global pool of users and developers. Such participation promotes innovations and advancements in the field. The free exchange of information between the users and developers is the key strength behind the open source software movement. In an open source environment, users of the system are encouraged to directly participate as part of the development community which creates a user's feedback loop; in a proprietary environment, users offer suggestions and report defects, but they are unable to participate directly in the development process since they do not have access to the source code (Boulanger 2005). Such collaboration and sharing of information by the open source community coupled with almost instant feedback from users results in the development of superior applications that are in harmony with users' needs.

Software quality and reliability is a major consideration in any software development. A study found that open source software (OSS) code appears to be at least equal and sometimes better than the quality of closed source software (Samoladas et al. 2004). Open source models promote software reliability and quality by supporting independent peer review which is a natural by product of open source projects (Westermeier 2005). The entire community has the opportunity to review and test the software for completeness and quality (Hardaway 2005). Open source development methods are also conducive to producing better quality code than closed source simply because open source communities can provide access to extensive expertise (Boulanger 2005) and can put more skilled time into a problem (Stewart and Gosain 2006). Furthermore, the open source approach favors the development of a relatively bug free product; a large number of people have the opportunity to review the source code as it is freely available, unlike the source code for the proprietary software which is not made public . Moreover, available data suggests that when a software error is discovered, the open source community responds more rapidly to fix the error than the proprietary software vendors, who have a tendency to cover up and deny flaws in the software (Boulanger 2005).

Other advantages of the open source approach is that the software development and improvement time can be reduced significantly as the operational version of the software could be posted anytime and revised frequently by a large base of highly motivated co-developers, based on the feedback received (Hardaway 2005).It also costs significantly less to produce open source applications because the open source community does not charge for its labor, and collaboration is easy (Monniaux 2004).

By and large, open source software competes very effectively with closed source software based on quality, reliability, cost, development time, and flexibility. However, these features by themselves will not ensure the long term viability of the open source software. What is also needed for the long term success of the open source software is a viable business model..

3 Open Source Business Model

The long term success of any product or service depends on the viability of its business model. A good business modeless one that indicates how a product or service will provide value to customers and revenue stream to organizations. In market driven economies, products or services that do not provide value to its customers will not be able to generate revenue and, therefore, will not survive. On the other hand, providing value to customers with insufficient revenue stream will also not ensure long term sustainability. Profit oriented organizations need financial incentives as they are in the business to make money; non-profit organizations need revenue to pay for normal operations and functioning of their organizations. In order to assess the feasibility of the open source business model, one needs to examine how it provides for the value and for the revenue stream; one without the other does not produce a feasible business model.

The open source software certainly provides value for its users, the major one being the unrestricted availability of the source code that can be freely modified to meet users' requirements. Other valuable features include reliability, fewer bugs, almost immediate feedback from the user community, short development time, and low cost. Even commercial software vendors have begun to realize that restrictions on the distribution and modification of the software hinder its use (Stallman 1992). As a result, they – commercial software vendors - are now more amenable to the idea of opening up the source code to the public. One of the features that differentiates open source from the closed source software is the fact that the open source software is available free of charge. Given that, how does open source software generate revenue – a component as important as the value component in the business model?

The open source software projects are initiated by a group of developers who are mainly driven by the challenge of creating a flexible and user centric computing environment. Their motive is purely altruistic, not financial. However, as the popularity of the open source packages increase, a gap develops between the needs of the user community and the features provided by the core developers (Ousterhout 1999). The user community demands additional services such as support and consulting, training materials, and development tools used with OSS packages (Samoladas 2004));Users are even willing to pay for these services. As business opportunities created by the gap between the users' needs and features provided by the open source software become lucrative enough, commercial ventures move in to fill this gap.

Sometimes a company may open up some of its software code to the outside public so that it can quickly develop the technology into a stable one so as to expand its user base (De Latt 2004). This supposedly will help the company to sell the branded versions of the software and proprietary applications which then could become the source of revenue. At other times, companies encourage outside developers to write open source applications to augment the sales of their basic platform (hardware, software or devices like cell phones or PDA's). For example, in January 2001 Motorola invited developers to write Java 2 Platform Micro Edition applications for its iDEN handsets so that it could improve the sales of theiriDE Nhandsets in competitive markets.

One of the main differences between the open source and closed source business model is that the closed source business model relies on revenue generation by selling the software. Given that the marginal cost of creating the software is virtually zero and the competition intense, there is a downward pressure on the price that a software company can charge. Therefore, in order to maintain the price and the market share, software companies often introduce special features at an additional cost into their software. These very same special features that are supposed to give competitive advantage make the software more inflexible, expensive, and less competitive.

5 Conclusions

Proprietary software companies are reluctant to open up their source code to the public since they view sale of their software as the main source of revenue. But given the characteristic of the software industry where the marginal cost of producing the software is virtually zero and collaboration easy, there seems to be a growing sense of realization among the software companies that the sale of software may not continue to be a viable source of revenue. If such is the case, there will be very little incentive for companies to make the source code proprietary. To generate revenue, software companies will fully exploit the open source paradigm for commercial developments. It is quite conceivable that closed source software (CSS) projects might evolve into open source software(OSS) projects and the software development might be dominated by open source philosophy.

References

Boulanger, A. (2005). Open-Source versus Proprietary Software: Is One More Reliable Than the Other? *IBM Systems Journal,44(2),* 239-248.

Cusumano, Michael A. (2004). Reflections on Free and Open Software. *Communications of the ACM, 47(10),* 25-27.

De Latt, Paul B. (2004). Evolutionof Open Source Networks in Industry. *The Information Society, 20(4),* 291-299.

Delong, J. Bradford and Froomkin, A. Michael. (2000). Beating Microsoft at its Own Game. *Harvard Business Review, 78(1),* 159-164.

Hardaway, Donald E. (2005). Borrowing a Page from Open Source Software. *Communications of the ACM, 48(8),* 125-128.

Kalina, I. and Czyzycki, A. (2005). The Ins and Outs of Open Source. *Consulting to Management, 16(3),* 41-46.

McGrath, N. (2004).What Exactly Are the Merits of Open-Source Software? *IEE Review,October 2004*, 46-50.

Monniaux, David. (2004). Open Source vs. Capitalism and Communism.*Communications of the ACM, 47(4),* 11-13

Osterhout, J. (1999). Free Software Needs Profit.*Communications of the ACM, 42(4),* 44-45.

Samoldas, I., Stamelos, I., Angelis, L. and Oikonomou, A. (2004). Open Source Software Development Strive for Even Greater Code Maintainability.*Communications of the ACM,47)10),* 84-87.

Stallman "Why Software Should be Free," April 24, 1992 (available online at http://www.gnu.org).
Stewart, Katherine J., Gosain, Sanjay. (2006). The Impact of Ideology on Effectiveness in Open Source Software Development Teams.*MIS Quarterly, 30(2),* 291-314.
Towle, H., McFarland, C. and Keppler E. (2004). Open Source Issues in Business, *Journal of Internet Law, 8(6),*3-11.
Westermeier, T.J. (2005). Managing Open Source Software Risks in M&A Corporate Transactions. *Journal of Internet Law, 9(5),* 20-24.