

0312005, EconWPA,
<http://ideas.repec.org/p/wpa/wuwpio/0312005.html>

Hayek, Friedrich A. von 1945. "The Use of Knowledge in Society," *American Economic Review* 35(4): 519-530(September).

Langlois, Richard N., and Giampaolo Garzarelli 2008. "Of Hackers and Hairdressers: Modularity and the Organizational Economics of Open-source Collaboration," *Industry and Innovation* 15(2): 125-143(April).

Raymond, Eric S. 2001. *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary* (revised edition). Sebastopol, CA: O'Reilly & Associates, Inc.

Weber, Max 1964. *The Theory of Social and Economic Organization*. Translated by A. M. Henderson and Talcott Parsons. Edited with an "Introduction" by Talcott Parsons. New York: The Free Press. Originally published 1947.

From the Editors

FOSS: The Evolutionary Process

Rahul Kala

rahulkalaiiitm@yahoo.co.in

http://students.iiitm.ac.in/~ipg_200545/

*Student, Atal Bihari Vajpayee
 Indian Institute of Information Technology and
 Management Gwalior, Gwalior, MP, India*

The Free/Open Source Software has gifted the world some of the finest pieces of software that are in extensive use. These include PHP, MySQL, Apache and many others. In the modern world Software represents a lot more than just being lines of code. They have the potential to change the way society functions, believes or aspires. A brief look over various deployed software at the macro and micro level reveals that either the software adjusts to the requirements of the society or the society adjusts as per the requirements of the software. While the commonly used Personal Information Management (PIM) software belongs to the former category, the Core Banking System (CBS) in banks is a typical example of the second category. This class of software also leads to resistance of adoption among

people like the resistance among the railway employees in the adoption of automating ticketing system and the bank employees in the use of CBS.

FOSS forms a characteristic way of development and deployment with a multi level collaboration amongst the users, testers and developers. Literature is filled with surveys, studies and explanation related to motivation, coordination, management and development under FOSS. In this article we present a general overview of the evolution (or development) of FOSS products along with time.

Evolution is the most fundamental word that represents the natural manner of growth and development. Evolution over the years has led to creation of complex species including humans from the basic uni-cellular organisms. The rivers evolved their path over time. Evolution grants full freedom for the entity, system or the phenomenon to develop in their own way by adapting to the changing environments every instance of time. Numerous such entities may evolve or develop from a single parent. The good ones survive and the bad ones get destroyed. This is what Darwin refers to as Survival of the Fittest. Evolution is hence an iterative process with the solutions improving over time and generations as per the changing environment.

Evolution has always led to magnificent pieces of work in the natural eco-system. These have been a great source of motivation for people who aspire to imitate the same in the artificial machines or processes. The majesty lies in the fact how Mother Nature is able to create systems of greatest utility out of the infinite possible systems that may be possible. One of the best examples of this is the Human brain. The brain continuously adapts itself as per the surroundings. This enables it to learn new things and attain more complex understanding and reasoning ability as the human grows. Another example is the birds. The coordination and synchronization that they have developed in times of flocking, escaping from enemies or other similar times is a matter of great perfection.

Software or in large the systems are always evolutionary in nature. There is a huge market comprising of numerous users. There are a large number of service or product providers that go by their own assessments, fundamentals and developmental principles. There is a continuous improvement to generate or evolve newer systems that are better as compared to the previous systems. Every new generation system tries to learn from the old generations. Many of the characteristics may be

inherited from the old systems. At the same time many new characteristics may be sought to enhance the system. The whole process follows the Darwinian principles where the most competent systems survive and the others perish. Another beautiful fact is that the systems do not only try to improve themselves as per the understanding of the past trends and the competitors, rather they compete in a changing environment where nothing remains constant. Changes may be due to external factors, change in customer needs, changes in technology, etc. It is common to find new systems emerging and many dying out with time.

The proprietary software forms a class of software that is driven by a corporation that owns the software. The software is usually built upon a good user requirements analysis. This not only yields the development of good software, but software that satisfies the users. The users are normally a well targeted section of the society. Every corporation offers a limited options governed by its business model. The whole development moves strictly by the company objectives and strategies. Further, competition exists among various firms that does not kill innovation. New products keep hitting the market. If a product fails it is replaced by a newer one in relatively shorter release cycle. The whole process follows this strict iteration of analysis, development, release and acceptance.

The open source counterpart has a different mechanism. Here everything is driven in a very free or natural way. There are numerous software available that may cater to the diverse needs of the diverse users. Further, market drives the development where users themselves place their demands or make their suggestions. They are waiting for the products to be released and demand whatever changes they might feel is necessary for them. There are a dozens of companies as well as users ready to meet these demands. The market has much larger releases and they come much frequently. Also the competitions exist at all local and global levels that leads to continuous improvement.

In 'the Cathedral and the Bazaar', Raymond explains the characteristic manner of functioning of the two forms by drawing their analogy with the strict code of conduct style of the Cathedrals and the free market style of the Bazaars. He explains how community based development and early and frequent releases lead to an advantage to the Bazaars or the Open Source model as compared to its counterpart. I use many of the same ideologies to compare and contrast our notion of the evolutionary nature of the two

models. Now we analyze the various aspects of the two movements one by one.

First notable characteristic is the adaptation against the changing environment. We know the fact that the user requirements are on a constant change due to various factors. The demands change with time and technology. The open source model with users as the co-developers presents a promising manner in which major and minor changes can be rapidly incorporated and circulated in the community. The early releases also ensure a fast acceptance and adoption by the users. This ensures that software is at pace with the changing environment as compared to the proprietary model where the changes take time to be incorporated and released. The reusability of code has further advantages in quicker integration of technologies and features into software as against the proprietary model.

However another aspect of the adaptation is the forecasting the future and providing the user with components they would require in future. Another way to view this would be to create demand among the users. Here the open source model derives its advantages as a result of constant interaction between the developers, users and numerous other people who constantly discuss share knowledge. This knowledge sharing contributes a lot in exploiting every possibility where thousands of ideas are generated and there are an equal number of people ready to develop, integrate, debug and deploy these ideas to make great software. This is unlike the research and development (R&D) along with analysis cycle of proprietary software which follows strict release styles with limited people.

The second major point we highlight in this article is the fact that many times there are diverse demands among people belonging to different sections of the society. These demands may be a result of culture, work procedures, or individual behaviors. The open source model may be conceived as a highly modular model with excellent re-usability of code. As a result the basic prototype is adjustable across multiple platforms, environments and demands. The modifications may be carried by free-lancers, hired developers, corporate using the software or the individual himself. These are released back to the community and increase the options available for others. As a result with the least of efforts, the people are able to get the best software. The proprietary software would either have only minor custom adaptation of GUI as a facility provided in the interface or the adaptation would be possible on demand only by the corporation owing the software.

This has known problems of resource intensive development, centralized nature, limited development possibility and availability of limited flavors.

The third point of discussion here is that real market is the evaluator of the system. It is uncertain what works in the real market unless the same is implemented in action. Both the open source and the proprietary software can have some idea of the performance by their speculations of the market and by the analysis of similar products. The open source with its community driven nature has a closer relation between the various elements. On the other hand the proprietary software is driven by pilot phases that help it speculate the market. Release of any product faces known and unknown competition that may drive its performance. Market speculation, which nearly goes with the performance of the actual product, is largely dependent on how close you bind with the actual users and their demands. This is naturally better in a community based approach. The proprietary software are now trying to use the same power of community for greater penetration into the market.

The last point we discuss is that there are always numerous corporate providing the same solution with different release periods. Open source model with constant user testing, community development and releases provide continuous flow of software into the market as compared to its counterpart. The various flavors may be provided by different groups, organizations or corporate. As a result the solutions are always plentiful and more suited for the requirements. Another point is that the changes should be substantial enough for the users to make a radical change in their manner of functioning or a change in option of use which is difficult task. It would be very difficult for a Windows user to migrate to Linux and similarly with other software. This is being provided by both proprietary and open source models with rigorous planning, analysis and monitoring.

The mentioned factors may be studied at two levels, local and global. The former would try to maximize the impact to a section of society, not caring about the rest of the world. The latter would try to look at the world at large and try to reach the goals against the common demands of the people. Localization may have such a big impact on the particular section of the society that even the best projects may not be able to have. An example is the Sahana project initially initiated by the Government of Sri Lanka. This Disaster Management software started from many parts of Sri Lanka that are hit by disasters and

later proved to be a revolution in its own way. The hole in the wall and the 100 dollar laptop are other projects that are in many ways specific to regions and have the potential of a deep impact. Numerous e-Governance projects may also be quoted here.

But letting the market decide the fate of the software may sometimes be very risky. We know that natural processes lead to disasters in many cases and the same may be the case with software as well. In case of software the impact may be even greater. This necessitates the need of the use of right strategy for evolving the software. This is what the software models refer by the role of leadership. It ensures the evolution of software in the correct direction.

The strategist of an open source does not play the same role as the team leader of the proprietary software. The strategist is one amongst the community who understands the community and coordinates them to optimize the output in the correct direction. On the other hand the team lead of proprietary software is one who just has some goals and is supposed to meet them against all odds and evens. There is no community here. Imagine being at the top of a cliff where you see the river rushing out. What would be better, to work over the path first and steer the river accordingly by trying to control each and every drop of water, or to see the flow and then make amendments accordingly? This is the difference between the two approaches.

Now since we've seen the two movements and the evolutionary approach, let's see how it all moves together. A market being filled with open and closed solutions for the same product is much like having the cathedrals situated in the vicinity of the bazaars. The user chooses solutions as per his needs, choices, beliefs and myths. Many people would not be aware of the choice they make – the open or the closed. They purchase whatever looks well from outside. Many people using VLC Media Player may not be aware of the GPL license associated with the software. In such a case what matters at the end is the final product and not the process from a users perspective. The process must hence ensure a competitive product irrespective of the model of implementation. As per the discussion the open source may have an edge in making better products. The other kind of people is those who take active participation in community and contribute at all ends. For them the open source may be a very strong choice as compared to the counterpart.

Open source naturally has possibilities of better products as per the specific needs of the users.

Problems may come in case the proprietary software has already established its strong presence in the market over the product. In such a case one of the major tasks is to make a shift in the choice of the users which is a very difficult task. This may be the reason for many of the open source solutions struggling in the market. It might take time to break this resistance of users, but the community driven open source definitely has potential for better solutions in the future. We might also witness a migration of the proprietary corporate on the open source lines. If not already done, once the monopoly breaks, the world would be completely free. So using this magazine as a platform let us all contribute at all levels to ensure in every field the proprietary software is challenged and outperformed.

From the Editor

Reaping the benefits of Open Source*

Venkateshvar Hariharan
venky@redhat.com

*Corporate Affairs Director,
Red Hat, India*

** Article taken from
<http://www.osindia.blogspot.com/>. Reprinted with
author permission*

Open Source Software is moving from the edge of the enterprise into its very heart. CIOs can benefit from this development

Recently, in a Gartner report, 'The State of Open Source 2008', it was mentioned that, "By 2012, more than 90 percent of enterprises will use Open Source in direct or embedded forms." The report goes on to add that, "Open Source is a phenomenon with a broad impact. Chances are, if you do not think you use it, then you use it; and if you think you do use it, then you use lots more of it than you know."

Why is Open Source becoming so pervasive? The reason is that we are now entering an era of Collaborative Innovation. Open Source Software (OSS) is the leading example of this trend, but the Open Source development model based on collaboration, community and the shared ownership of knowledge is rapidly expanding to other areas like

content (Wikipedia), medicine (Open Source Drug Discovery), scientific publishing (Public Library of Science) and other areas of society. With 1.2 billion people on-line, the Internet, which is the largest collaborative platform that mankind has ever seen, has enabled OSS like Linux, Apache, Mozilla Firefox, Open Office and others to flourish. In the next couple of years another 600 million people will join the Internet. Thus the trend towards increasing collaboration is only set to grow and this is reflected in the explosive growth of Open Source projects across the world.

A few years back, OSS could be found on the edge of enterprise, running workloads like mail servers and web servers. However, the growing maturity of the OSS ecosystem means that it is now moving into the very heart of the enterprise, running mission critical servers, desktop computers and even application areas like CRM, ERP, Document Management, collaborative wikis, Content Management Systems and many others.

In these recessionary times, OSS also offers a major advantage in that users can often freely download and try out the software and pay only for value-added services like support. The growing interest in OSS has thrown up a number of software startups that specialize in supporting OSS. For CIOs, this development provides an alternative because the cost of buying support for OSS is usually far less than the cost of purchasing licenses for proprietary software.

For instance, download and check out OpenOffice.org, the full-fledged, Open Source office productivity suite. Many organizations have made OpenOffice.org, the default choice on their desktops and have generated significant cost savings when compared to proprietary office suites. A leading bank, known for its technological savviness, has almost 70 percent of its staff working on OpenOffice.org. A few years ago, they looked at the increasing cost and hardware requirements of proprietary software and decided to switch to OpenOffice.org. Initially, users had to adjust to the new software, but a four-member helpdesk enabled them quickly become comfortable with OpenOffice.org. The cost of the help desk was far lower than the licensing fees and the increased hardware costs that would have been incurred on proprietary software, which has been restricted to a small group of financial analysts within the company.

Similarly, a study done by IIM Ahmedabad found that the Government of Delhi has saved almost 80 percent by switching to OpenOffice.org. One